Université de Strasbourg

Ecole doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

Habilitation à Diriger les Recherches

Juin 2018

Representation Methods for Image Analysis and Interpretation

Discipline: Computer Vision

Naoufel WERGHI

Jury

Rapporteurs :

Boulbaba Ben Amor, Professeur des universités, IMT Lille Douai

Jean-Luc Dugelay, Professeur des universités, EURECOM

Fabrice Mériaudeau, Professeur des universités, Université de Bourgogne

Examinateurs :

Ernest Hirsch, Professeur des universités, Université de Strasbourg

Garant :

Christophe Doignon, Professeur des universités, Université de Strasbourg

iCUBE

Laboratoire ICube (UMR CNRS 7357), Pôle API, Bd S. Brant, 67412
Illkirch

Tel : (0)3 68 85 45

ACKNOWLEDGMENTS

First, I would like to thank all the members of the jury for bestowing upon me the honour to agree for evaluating this work.

I would like to express my sincere gratitude and appreciation towards Christophe Doignon, for his timely counsel, guidance, great support throughout the preparation of this work.

I would also like to thank Michael, Fatma, Claudio, Nessima, Nesrine, Bilal, Safa, Haykel, and Yousef; I had the pleasure to see them evolve during their Master and PhD thesis.

Many thanks also go to Alaa, Marwa, Aruna and Salam for their great assistance during the course of my research.

I would also like to take this opportunity and thank Stefano, Alberto, Munawar, Salman, Jorge, Harish, Ayman, Bob, Anthony, Paul, for the wonderful fruitful collaboration we had together.

My heartfelt gratitude to everyone I had the privilege to work with in this study.

Finally, I owe immense gratitude to my wife, Sonia, for her continued and unfailing love, support and understanding all these years.

Table of Contents

Preface

This dissertation traces my academic activities that were undertaken since my PhD thesis in defence back in December 1996.

From 1997 to 1999, during my research fellowship at the Division of Informatics in the University of Edinburgh, I was a member of the Perception, Action and Behaviour unit that was directed by Bob Fisher. This unit has been conducting world class research in 3D computer vision - more specifically on the analysis of 3D scanned object data. This fascinating topic punctuated my research activities for a while. During my lectureship in the department of computing science at the University of Glasgow from 2000 to 2002, I continued to work on this theme, intensifying my focus on 3D data acquired with human full body scan data. As an emerging technology at that time, it gave me a chance to propose some pioneering works related to this topic. It also sparked my interest in biometry. I carried on with this research after moving into the College of Information Technology in the University of Dubai in 2003. I was also able to extend the scope of the initial framework of human body shape analysis to encompass a wider class of objects and give direction to a new research, noticeably in the field of medical imaging.

In 2009, my entry into the Department of Electrical and Computer Engineering in Khalifa University (KU) marked a new phase in my journey of initiating new and cutting-edge research activities. As a case in point, the launch of the post-graduate program in KU coupled with the advent of research funding institutes in the UAE that year provided a great impetus to explore new research avenues and investigate new projects, noticeably in 2D and 3D image analysis as well as their applications in medical imaging and biometry. It also gave me an opportunity to establish substantial international collaborations.

Inevitably, it is not feasible to cover all the works undertaken during the aforementioned phases. I have voluntarily preferred to highlight research activities that I believe have had a stronger impact than others, and for which my contribution has been beyond significant, while trying my best to produce a coherent document for the reader.

This dissertation comprises three parts. The first part is summary of the dissertation in French. The second part provides a general overview of my personal and academic profile, which includes a history of appointments, scholarship activities, administrative services, teaching activities and a selected list of publications. The third part is a monograph that entails the compilation of my main research projects. This part is intended to be self-contained in that it provides a compact yet comprehensive overview about the problems addressed, the proposed methodology and the original contributions. The reader can also use it as a roadmap to consult the representative papers reported in the appendix. We conclude this section with an overview about potential future research directions.

Summary of the dissertation in French

Université de Strasbourg

Ecole doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

Habilitation à Diriger les Recherches

Juin 2018

Méthodes de Représentation pour l'Analyse et l'Interprétation des Images

Discipline : Vision par Ordinateur

Naoufel WERGHI

Jury

Rapporteurs :

Boulbaba Ben Amor, Professeur des universités, IMT Lille Douai

Jean-Luc Dugelay, Professeur des universités, EURECOM

Fabrice Mériaudeau, Professeur des universités, Université de Bourgogne

Examinateurs :

Ernest Hirsch, Professeur des universités, Université de Strasbourg

Garant :

Christophe Doignon, Professeur des universités, Université de Strasbourg

ICUBE

Laboratoire ICube (UMR CNRS 7357), Pôle API, Bd S. Brant, 67412 Illkirch

Tel : (0)3 68 85 45

Préface

Cette thèse retrace les activités académiques que j’ai entreprises depuis l’obtention de mon doctorat. De 1997 à 1999, mon affiliation comme chercheur post-doc à la division d'informatique dans l'Université d'Edinburgh m’a tout d’abord permis de mener des recherches sur l'analyse de données d'objets scannés en 3D. Ce sujet fascinant a rythmé mes activités pendant plusieurs années. J’ai ensuite continué à travailler sur ce thème au sein du département de sciences informatiques de l'Université de Glasgow de 2000 à 2002 en m’intéressant plus particulièrement aux images 3D obtenues avec des scanners dédiés aux corps humains. L’étude de cette technologie, émergente à l'époque, m'a permis d’une part de proposer quelques travaux pionniers sur ce sujet et a suscité d’autre part mon intérêt pour la biométrie. J'ai donc poursuivi cette recherche en intégrant le collège des technologies de l'information de l'Université de Dubaï en 2003. Cette période de mon parcours académique fut l’occasion d’étendre la portée du cadre d'analyse, initialement centré sur la forme du corps humain, en englobant une classe plus large d'objets et d’entamer une nouvelle recherche dans le domaine de l'imagerie médicale.

En 2009, mon affiliation au département de génie électrique et informatique de l'Université de Khalifa marque une nouvelle étape dans mon parcours académique. Le lancement du programme de troisième cycle à l’Université couplé à l'avènement des instituts de financement de recherche dans les Emirates Arabes Unies ont en effet généré une grande impulsion pour explorer de nouvelles voies de recherche et étudier de nouveaux projets, notamment dans l'analyse d'images 2D et 3D. Ces nouveaux travaux ont été l'occasion d'établir des collaborations internationales importantes.

Ce résumé comprend une version courte de mon curriculum vitae faisant état de mon profil personnel et académique. Une compilation de mes principaux projets de recherche fournit ensuite un aperçu compact mais complet des problèmes abordés, de la méthodologie proposée et des différentes contributions.

1. Curriculum Vitae

Nom:	Naoufel
Prénom:	Werghi
Adresse:	Khalifa University, POBOX 127788, Abu-Dhabi, UAE
Courriel:	Naoufel.Werghi@ku.ac.ae
Position actuelle:	Associate Professor

Education

1996: Doctorat en Vision Robotique, Université of Strasbourg, France.
1993: DEA en Instrumentation et Control, Université of Rouen, France.
1992: Eng. Diplômé en Génie Electrique, Ecole Nationale d'Ingénieurs Monastir, Tunisie.

Parcours Académique

2012-Present: Professeur Associé , Département de Génie Electrique et Informatique Université Khalifa, UAE.
2009-2011: Maître de Conférences, Département de Génie Electrique et Informatique Université Khalifa, UAE.
2003-2008 Maître de Conférences, Collège d'Informatique, Université de Dubai, UAE.
2000-2002: Maître de Conférences, Département d'Informatique, Université of Glasgow, UK.
1997-1999: Post-Doc, Département d'Informatique, Université d'Edinburgh, UK.
1996 : ATER, Ecole Nationale de Physique, Université of Strasbourg, France.

Domaine de Recherche

Analyse et interprétation des images 2D et 3D, avec appliquées à la biométrie, imagerie médicale et la modélisation géométrique.

Encadrement de thèses Doctorat

2016:Present : C.Tortoricci, Detecting and interpreting 3D facial patterns from multi-modal data, Khalifa University
2016-Present : B. Delail, Action and Gesture Recognition Under Dynamic Illumination Conditions, Khalifa University
2014-Present: N. Medimegh, Robust Watermarking of 3D mesh model objects, Université du Centre
2014:F. Taher: Early Detection of Lung Cancer Based on Sputum Color Images Analysis, Khalifa University
2004:Y. Xiao : Automatic segmentation and fitting of whole human body shape, University of Glasgow

Encadrement de thèses Master

2017-Present: R. AlKadi, A computer-aided diagnostic system for the early detection of prostate cancer using MRI images, Khalifa University
2017-Present: M. Almufti, Machine learning techniques for the classification of synthetic aperture radar images, Khalifa University
2017-Present: E. Alhadhrami, Feature extraction for moving target classification using radar doppler echoes, Khalifa University
2017: B. Taha, Automatic Polyp Detection in Endoscopy Videos, Khalifa University, UAE

2013: N. Medimegh, 3D triangular mesh watermarking using ordered ring facets, Université du Centre

2013: N. Bnouni, D face identification by fusion of spiral facets descriptors, Université du Centre,

2011: H. Boukadida, Facial landmark detection and facial surface alignment using 3D images, Université de Manouba

2011: Y. Megubli, Extraction of facial descriptors from 3D images, Université de Manouba

Encadrement de Chercheurs

2015 : A.Elkahteeb “Computer-Aided Diagnosis System for Early Detectio Cervix Cancer from Pap Smear images”

2015 M-Chendeb “Computer-Aided Diagnosis System for Early and Automated Detection of Infantile Dysmorphic Syndromes”

2012 A. Vivekanand “Design and Implementation of a Grading System for Assessing Posterior Capsule Opacifiaction using Medical Digital Images”

2012 S. Khalifa “3D face matching for incomplete 3D facial images”

Publications sélectionnées

M. Hayat, S. H. Khan, **N.Werghi** and R. Goecke, “Joint Registration and Representation Learning for Unconstrained Face Identification”, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2767-2776

I. Reda, A. Shalaby, M. Elmogy, A. Abou Elfotouh, F. Khalifa , M. Abou El-Ghar, E. Hosseini-Asl, G. Gimel'farb, **N. Werghi** and A. El-Baz, "A Comprehensive Non-invasive Framework for Diagnosing Prostate Cancer," Computers in Biology and Medicine, vol. 81, pp. 148-158, 2017.

N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, ‘’ Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh’’, IEEE Trans. on Information Forensics & Security, 11 (5), pp.964-979, 2016.

N. Werghi, S. Berretti, A. Del Bimbo, “The mesh-LBP: a Framework for Extracting Local Binary Patterns from Discrete Manifolds”, IEEE transactions on image processing, Vol. 24, No.1, pp. 220-235, 2015.

N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, “Local Binary Patterns on Triangular Meshes: Concept and Applications”, Computer Vision and Image Understanding, 139, pp.161-177, 2015.

S. Berretti, **N. Werghi**, A.D. Bimbo P. Pala, “Selecting stable keypoints and local descriptors for person identification using 3D face scans”, The Visual Computer, springer, pp. 31-41, 2014.

A. Vivekanand, **N. Werghi**, H. Al-Ahmad, “Multi-scale Roughness Approach for Assessing Posterior Capsule Opacification”, IEEE Journal of Biomedical and Health Informatics, Vol.18, No.6, pp. 1923 – 1931, 2014.

N. Werghi, R. Sammouda, F. Alkirbi, “An unsupervised learning approach based on a Hopfield-like network for assessing Posterior Capsule Opacification from digital images”, Pattern Analysis and Applications, Springer, 13(4), pp. 383-396 , 2010.

N. Werghi, Y. Xiao, P. Siebert , “A Functional-Based Segmentation of Human Body Scans in Arbitrary Postures,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 36, No.1, pp. 153-165, February 2006.

N. Werghi, R.B. Fisher, C.Robertson, A. Ashbrook, “Object Modeling by Incorporating Geometric Constraints”, Computer-Aided Design, Elsevier, Vol.31, No.6, pp.363-399, May 1999.

2. Activités de recherches

Mes activités de recherche en vision par ordinateur couvraient différents aspects de l'analyse et de l'interprétation d'images. Une grande partie de mes contributions a été consacrée à la représentation, un concept que je décris comme l'élaboration d'un cadre de calcul optimal pour encoder les données visuelles afin d'assurer une solution efficace pour un problème donné en vision par ordinateur.

Ma première contribution à cet égard s'inscrit dans le projet pionnier sur les modèles de Conception Assistée par Ordinateur (CAO) d'objets manufacturés utilisant des scans 3D en forme de nuage de points. Cette recherche devait permettre aux concepteurs CAO de construire de nouveaux modèles à partir d'images 3D en introduisant de manière interactive de nouvelles spécifications sur le modèle sous forme de contraintes géométriques. Les méthodes courantes au moment de l'étude employaient des algorithmes génétiques. De telles approches facilitent la mise en œuvre des contraintes géométriques mais demandent un temps de calcul prohibitif (échelle d'heures) que les concepteurs CAO n'ont pas. Nous avons ainsi proposé une nouvelle approche combinant représentation vectorielle du modèle objet et contraintes géométriques. Il en résulte un modèle d'optimisation efficace et un nouveau paradigme permettant d'aborder le problème avec une optimisation quadratique standard. Les résultats de cette recherche sont présentés dans l'article de [Werghi ,1999]

Un autre projet traitant de la segmentation d'images 3D représentant des formes humaines propose une nouvelle représentation robuste fondée sur le diagramme de Reeb-graphe. Cette nouvelle représentation incorpore un ensemble de contraintes topologiques locales pour résoudre les problèmes de variabilité et de données de la posture et de bruit [Werghi,2006]. Ce cadre a été étendu dans un travail ultérieur à une classe plus large comprenant les objets articulés et tubulaires [Werghi, 2006a].

Une grande partie de mes recherches a été consacrée à la conception de représentations appropriées pour l'analyse de maillages triangulaires. Dans ce contexte, j'ai contribué à développer le concept d'histogramme géométrique, la première structure d'histogramme proposée comme descripteur de forme locale pour les maillages triangulaires [Ashbrook, 1998]. Pour pallier au manque de structure ordonnée, j'ai ensuite proposé le concept des anneaux de facettes ordonnées (ORF) [Werghi, 2011 ; werghi, 2011b ; Werghi, 2012]. L'ORF permet de générer des structures ordonnées sur des maillages triangulaires qui peuvent être déployées localement et globalement. L'ORF a été adapté à plusieurs tâches de traitement des surfaces faciales, notamment le recadrage, la compression, l'alignement et la détection du nez. L'ORF a également formé la fondation du mesh-LBP [Werghi, 2015], un nouveau concept qui a permis d'étendre les modèles binaires locaux [Ojala, 2002] aux maillages triangulaires. Ce concept a été appliqué avec succès dans différentes applications comprenant la reconnaissance de relief [Werghi, 2015 ; Werghi, 2015a ; Tortorici, 2017] et la reconnaissance de visage 3D [Werghi, 2015b ; Werghi, 2016].

Dans une autre contribution reliée au domaine de l'imagerie médicale, toujours dans le cadre de la représentation, j'ai proposé un système de diagnostic assisté par ordinateur pour évaluer la gravité d'une pathologie oculaire appelée opacification postérieure de la capsule [Aruna, 2014]. L'approche est fondée sur un nouveau concept, baptisé « rugosité multi-échelle ». Cette approche permet d'éviter les problèmes de segmentation de l'image émanant de la texture irrégulière et bruitée caractérisant les images PCO.

L'avènement récent des paradigmes d'apprentissage en profondeur fut l'occasion d'étudier des approches basées sur la reconnaissance de visages et sur l'imagerie médicale. Dans un travail récent [Hayat, 2017], j'ai contribué à développer un système de réseau neuronal convolutif qui apprend à enregistrer et à représenter simultanément les visages. Le système proposé a pour but d'améliorer de façon significative les performances de reconnaissances faciales.

Dans un autre projet, j'ai proposé une nouvelle contribution concernant la détection des polypes dans les images de coloscopie. Les polypes sont des protubérances qui se développent au niveau du tractus intestinal. Leur détection et leur élimination précoces sont cruciales pour une meilleure prévention du cancer colorectal. Dans ce contexte, nous avons proposé une méthode de transfert d'apprentissage utilisant des architectures standards (AlexNet et VGGNet) comme extracteur de primitives. Les vecteurs de primitives obtenus sont alors déployés comme entrées des classifications classiques telles que le SVM et le Softmax. Le suivi des polypes est un autre aspect qui a été étudié dans cette recherche. Les mécanismes de suivi classiques n'utilisent l'information d'intensité qu'à des fins de suivi. Cependant, dans cette thèse, nous avons expliqué que l'ajout de la contribution de couleur avec l'intensité pourrait conduire à un meilleur système de suivi. L'algorithme utilise trois formats de couleurs indépendants et une transformation affine. Les résultats de cette recherche ont été diffusés dans [bilal, 2017].

Pour la prochaine étape de mes travaux, je prévois de poursuivre une recherche multidisciplinaire couvrant l'analyse et l'interprétation de données 2D et 3D. Il sera également question de cibler des modèles et des applications d'interface homme-machine innovants. Compte tenu des progrès récents dans l'acquisition de vidéos 3D, plusieurs nouveaux scénarios d'interaction machine-personnes peuvent être envisagés. Les domaines d'application potentiels comprennent entre autres la gestion de l'identité, la réadaptation médicale, le divertissement et les soins aux personnes âgées. Avec la prolifération rapide des téléphones intelligents, les applications mobiles multimodales intelligentes intégrant et personnalisant les modèles d'interface constituent une autre direction prometteuse.
La récupération d'informations multimédias basées sur le contenu devrait être un sujet de grand intérêt dans un avenir proche. En effet, avec la disponibilité généralisée des numériseurs 3D et le domaine en plein essor de la technologie multimédia, de vastes collections de modèles multimédias hybrides peuvent être facilement construites et connectées à Internet pour différentes applications dans différents secteurs. Le développement de structures et de mécanismes adaptés aux besoins du client pour interroger et extraire des informations de ces bases de données hétérogènes constitue un défi intéressant.

Références

[werghi,1999] N. Werghi, R. Fisher, C. Robertson and A. Ashbrook, “Object reconstruction by incorporating geometric constraints in reverse engineering”, Computer-Aided Design, vol.31, no.6, pp.363-399, 1999.

[Ashbrook, 1998] A. P. Ashbrook, R. B. Fisher, C. Robertson, N. Werghi, “Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms”, Proc. European Conference on Computer Vision, 1998, Friburg, Germany, June, pp. 674-686.

[werghi06] N. Werghi, Y. Xiao and P. Siebert , “A Functional-Based Segmentation of Human Body Scans in Arbitrary Postures,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 36, No.1, pp. 153-165, 2006.

[werghi,2006a] N. Werghi, “A Robust Approach for Constructing a Graph Representation of Articulated and Tubular-like Objects from 3D Scattered Data”, Pattern Recognition Letters, vol. 27, no.6, pp. 643-651, 2006.

[werghi02b] N. Werghi, “Recognition of Human Body Posture from a Cloud of 3D Data Points Using Wavelet Transform Coefficients”, Proc. International Conference on Automatic Face and Gesture Recognition Washington, 2002, USA, May.

[werghi05] N. Werghi, “A Discriminative 3D Wavelet-based Descriptors: Application to the Recognition of Human Body Postures”, Pattern recognition letters, vol.26, no.5, pp. 663-677, 2005.

[werghi, 2011] N.Werghi, “Assessing the Regularity of 3D Triangular Mesh Tessellation Using a Topological Structured Pattern”, Computer-Aided Design and Applications, vol. 8, no.5, pp. 633-648, 2011.

[werghi,2011b] N. Werghi, H.Bhaskar, Y.Meguebli and H.Boukadia, “The Spiral Facets: A Unified Framework for the Analysis and Description of 3D Facial Mesh Surfaces”, Proc. Int. Conference Computer Vision Theory and Applications (VISAPP), 2011, Algrave, Portugal, pp. 30-39. BEST PAPER AWARD.

[werghi, 2012] N. Werghi, M. Rahayem and J. Kjellander. “An ordered topological representation of 3D triangular mesh facial surface: Concept and applications”, EURASIP Journal on Advances in Signal Processing, pp.144, 2012.

[ojala, 2002] T. Ojala, M. Pietikäinen and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, 2002.

[werghi, 2015] N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, “Representing 3D Texture on Mesh Manifolds for Retrieval and Recognition Applications”, IEEE Conf. Computer Vision and Pattern Recognition, CVPR2015, Boston, USA, 2015, pp. 2521-2530.

[werghi, 2015a] N. Werghi, S. Berretti and A. Del Bimbo, “The mesh-LBP: a Framework for Extracting Local Binary Patterns from Discrete Manifolds”, IEEE transactions on image processing, vol. 24, no. 1, pp.220-235, 2015.

[tortorici, 2017] C. Tortorici, N. Werghi, S. Berretti, “Defining Mesh-LBP Variants for 3D Relief Patterns Classification”, 7th International Workshop on Representation, analysis and recognition of shape and motion from Image data, 2017, Savoi, France. BEST STUDENT PAPER AWARD.

[ahonen2006] T. Ahonen, A. Hadid and M. Pietik¨ainen, “Face description with local binary patterns: Application to face recognition”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.28, no.12, pp.2037–2041, 2006.

[werghi, 2015b] N. Werghi, C. Tortorici, S. Berretti and A. Del Bimbo, “Local Binary Patterns on Triangular Meshes: Concept and Applications”, Computer Vision and Image Understanding, 2015.

[werghi, 2016] N. Werghi, C. Tortorici, S. Berretti and A. Del Bimbo, ‘’ Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh’’, IEEE Transactions on Information Forensics & Security, pp. 964 – 979, 2016.

[aruna, 2014] A. Vivekanand, N. Werghi and H. Al-Ahmad, “Multi-scale Roughness Approach for Assessing Posterior Capsule Opacification”, IEEE Journal of Biomedical and Health Informatics, vol.18, no.6, pp. 1923 – 1931, 2014.

[hayat, 2017] M. Hayat, S. H. Khan, N.Werghi and R. Goecke, “Joint Registration and Representation Learning for Unconstrained Face Identification”, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2767-2776

[bilal,2017] B. Taha, C.Doignon, N.Werghi and J.Dias, “Fast polyps tracking with color image regions alignment”, Surgetica 2017, Strasbourg France

Part-1: Personal file

I. Curriculum Vitae

Name:	Naoufel
Surname:	Werghi
Address:	Khalifa University, POBOX 127788, Abu-Dhabi, UAE
Email:	Naoufel.Werghi@kustar.ac.ae
Tel:	971 50 427 1945
Current Position:	Associate Professor

Education

1996:	PhD in Image Processing and Robotics, University of Strasbourg, France.
1993:	Masters in Instrumentation and Control, University of Rouen, France.
1992:	Eng. Diploma Electrical Engineering, Ecole Nationale Ingenieurs Monastir, Tunisia**.**

Professional Appointments

2012-Present:	Associate Professor, Department of Electrical and Computer Engineering, Khalifa University, UAE.
2009-2011:	Assistant Professor, Department of Electrical and Computer Engineering, Khalifa University, UAE
2003-2008:	Assistant Professor, College of Information Technology, University of Dubai UAE.
2000-2002:	Assistant Professor, Department of Computing Sciences, University of Glasgow,
1997-1999:	Research Fellow, Division of Informatics, University of Edinburgh, UK.
1996:	Adjunct Lecturer (ATER), National College of physics, University of Strasbourg, France.

Research Interests

2D/3D image data analysis and interpretation with application on smart systems, medicine, and biometry

Visiting Professorships

June 2017:	University of Canberra, Australia.
June 2016:	Institute Mines-Telecom, University Lille1, France
June-2015:	Zayed Institute the Pediatric Surgical Innovation, Washington DC
May-2013:	Korean Advanced Institute of Sciences and Technology, South Korea.
June-2011:	Media Integration Communication Center, University of Florence, Italy.
March 2001:	Department of Computer and Electrical Engineering, University of Louisville, USA.

Teaching activities

Programming, Artificial Intelligence, Data structure & algorithms, Software engineering

Awards

2017 Award for the BEST STUDENT PAPER in the International Workshop on Representation, Analysis and Recognition of Shape and Motion from Image data, RFMI 2017 for the paper titled *Defining Mesh-LBP Variants for 3D Relief Patterns Classification.* Paper presented by my PhD student Claudio Tortoricci.

2011 Award for the BEST PAPER in the International Conference on Computer Vision Theory and Applications, Portugal, for the paper titled *THE SPIRAL FACETS - A Unified Framework for the Analysis and Description of 3D Facial Mesh Surfaces*. Paper presented by my colleague Harish Bhaskar.

Research Grants

Year	Title	Fund	Source	Role	Affiliation
2017	Towards a computer-aided diagnostic system for the early detection of prostate cancer using diffusion weighted-magnetic resonance imaging	156000 DHS	AJF	PI	Khalifa University
2016	Automatic polyp detection in endoscopy videos	152571 DHS	AJF	PI	Khalifa University
2015	Hamama: Context-aware cloud-based robot assistance for informal care	195000 DHS	KU	Co-PI	Khalifa University
2015	Computer-Aided Diagnosis System for the early and automated detection of Infantile Dysmorphic Syndromes in the UAE	1674000 DHS	KU	PI	Khalifa University
2014	Computer-Aided Diagnosis System for the early detection of Cervix Cancer from Pap Smear images.	150 000 DHS	TFF	PI	Khalifa University
2014	People identification from Partially Hidden 3D Facial Images	160 000 DHS	NRF*	PI	Khalifa University
2014	Detecting Down Syndrome in Infants Using Facial 2D and 3D Images	190 000 DHS	KU	PI	Khalifa

					University
2014	Defining the Ancestral Lineages of Arabian Tribes using Mitochondrial DNA towards defining Arab Haplotypes for Fine Mapping of Disease Gene	195 000 DHS	KU	Co-PI	Khalifa University
2013	3D Face Matching for Incomplete 3D facial images	195 000 DHS	KU	PI	Khalifa University
2013	Engineering Personalized Medicine in the United Arab Emirates	2400 000 DHS	KU	Co-PI	Khalifa University
2012	A Grading System for Assessing Posterior Capsule Opacifiaction using Medical Images	194 500 DHS	NRF	PI	Khalifa University
2012	Algorithms for Watermarking Images	196 000 DHS	KU	Co-PI	Khalifa University
2009	People Recognition and Identification Based on 3D Facial Images	181 000 DHS	EF	PI	Khalifa University
2001	Automatic Segmentation and Fitting of Human Body Shapes from 3D Images	55 000 GBP	ESPRC	PI	University of Glasgow
2000	Automatic Segmentation of 3D images	3 000 GPB	CTF	PI	University of Glasgow
1999	Reconstruction of Built Environments for Virtual Reality Applications using Architectural knowledge	75 000 GPB	ESPRC	Co-PI	University of Edinburgh

AJF: Al-Jalila foundation UAE: http://www.aljalilafoundation.ae/
KU: Khalifa University
TFF: Terry Fox Foundation: http://www.terryfox.org/
NRF: National Research Foundation, UAE
EF: Emirates Foundation
EPSRC: Engineering and Physical Research Council, UK
CTF: Carnegie Trust for the Universities of Scotland, UK https://www.carnegie-trust.org/

Scholarship Activities

PhD supervision

Year	Student	Title	Affiliation
2016-Present	C.Tortoricci	Detecting and Interpreting 3D Facial Patterns from Multi-modal Data	Khalifa University
2016-Present	B. Aldelail	Action and Gesture Recognition under Dynamic Illumination Conditions	Khalifa University
2014	F. Taher	Early Detection of Lung Cancer Based on Sputum Color Images Analysis	Khalifa University
2004	Y. Xiao	Automatic Segmentation and Fitting of Whole Human Body Shape	University of Glasgow

Master thesis supervision

Year	Student	Title	Affiliation
2017-Present	R. AlKadi	A Computer-Aided Diagnostic System for the Early Detection of Prostate Cancer using Diffusion Weighted- Magnetic Resonance Imaging	Khalifa University
2017-Present	M. Almufti	Machine Learning Techniques for the Classification of Synthetic Aperture Radar Images	Khalifa University
2017-Present	E. Alhadhrami	Feature Extraction for Moving Target Classification using Radar Doppler Echoes	Khalifa University
2017	S. Salahat	Detection of Calcification in Abdominal Aortic Aneurysms	Khalifa University
2017	B. Taha	Automatic Polyp Detection in Endoscopy Videos	Khalifa University
2013	N. Mdimegh	3D Triangular Mesh Watermarking using Ordered Ring Facets	Nat. School of Engineering Souuse
2013	N. Bnouni	3D Face Identification By Fusion of Spiral Facets descriptors	Nat. School of Engineering Sousse
2011	H. Boukadida	Facial Landmark Detection and Facial Surface Alignment using 3D Images	University of Tunis,
2011	Y. Megubli	Extraction of Facial Descriptors from 3D Images	University of Tunis, Tunisia
2007	F. Alkirbi	PCO Assessment using Digital Images	University of Sharjah

PhD Examination

Year	Student	Title	Affiliation
2017	Y. Abukheil	Mapping and Navigation Techniques for Non-Invasive Active Endoscopic Capsules	Khalifa University
2016	E. Basaeed	Remote-sensing Image Segmentation using Convolutional Neural Network and Its Applications to Pan-sharpening and Detection	Khalifa University
2010	M.Rehayem	Segmentation and Fitting for Geometric Reverse Engineering	Orbero University

Research Associate Supervision

Year	R.A	Project	Affiliation
2015	A.Elkatheeb	Computer-Aided Diagnosis System for Early Detection of Cervix Cancer from Pap Smear images.	Khalifa University
2015	M.Chendeb	Computer-Aided Diagnosis System for Early and Automated Detection of Infantile Dysmorphic Syndromes	Khalifa University
2014	C. Tortotici	People Identification from Partially Hidden 3D Facial Images	Khalifa University
2013	M.Chendeb	Detecting Down Syndrome in Infants using Facial 2D and 3D Images	Khalifa University
	A. Vivekanand	Design and Implementation of a Grading System for Assessing Posterior Capsule Opacifiaction using Medical Digital Images	Khalifa University
2012	S. Khalifa	3D Face Matching for Incomplete 3D Facial Images	Khalifa University

Faculty Promotion Evaluation

Dr. Dr. Young-Ji Byon, Khalifa University, UAE, 2016
Dr. Mahmud S. Alkoffash, Al-Balqa Applied University, Jordan, 2016
Dr. Basel AlMourad Zayed University, UAE, 2014

Journal Paper Review

IEEE Trans. Image Processing
IEEE Trans. Information Forensics and Security
IEEE Trans. Pattern Analysis and Machine Intelligence,
IEEE Trans. Neural Networks and Learning Systems
IEEE Trans. Multimedia

Conference committees

Co-publication chair of IEEE International conference image processing, ICIP 2020

Member of the Steering committee of the Representation, analysis and recognition of shape and motion from Image data (RFMI), 2016, 2017

Chair of the "Image Processing and Multimedia Systems" track in the 56th IEEE Midwest symposium on circuits and systems, Columbus, Ohio, 2013.

Session chair in:

IEEE Conference, Men, Systems and Cybernetics, Shunghu, China 2008

International Conference on Machine Vision Applications, Tokyo, Japan, 2007

International Conference on Computer Vision Theory and Applications, Portugal, 2006

Administrative Experience and University Services

Khalifa University: 2009-Present

2011-2014 Deputy Chair of Postgraduate studies
Chair of the Research committee in the Electrical and Computer Engineering Department
Library Liaison coordinator in the Electrical and Computer Engineering Department

2010-2011 Final year project coordinator the department of computer engineering
Member of the Educational Experience committee
Member of the final year project committee in the college of engineering

University of Dubai: 2003:2008

2004-2008 Member of the Research Committee,

2005-2006 Member of the Undergraduate committee

2004-2007 Member of the General Education committee

University of Glasgow: 2000-2002

Department of computing sciences:
Course coordinator of C++ course
Coordinator of the departmental seminars

II. Teaching activities

My teaching activities were initiated at the University of Strasbourg, France during my PhD. I worked as a Teaching Assistant for two years (94-95 at the Institute of Science and Technologies. During my stint, I ensured the implementation of lectures and practical works in C programming and assembly language, tutorials and practical works in electronics as well as signal processing. I developed the entire teaching material, including lectures, tutorials and practical works.

During 95-96 I was appointed as a Lecturer at the National Higher Education School of Physics in the University of Strasbourg, France. Apart from ensuring the smooth completion of practical works in automation and control, I was also involved in monitoring and assessing students.

During my Research fellowship (97-99) at the Division of Informatics in the University of Edinburgh (UK) I ensured tutorials and practical works in machine vision and computational vision. I also contributed in supervising undergraduate and MSc students.

In 2000, I was appointed as the Assistant Professor at the Department of Computing Sciences in the University of Glasgow. My teaching encompassed C++ programming and Computer-Graphics. In all these courses, I developed the lecture materials, tutorial as well as laboratory sessions. I also developed a special workshop for the computer graphic course intended to prepare students for their final projects. In 2002, I co-developed a new module on digital image processing.

In the College of Information Technology in the University of Dubai, I taught Artificial Intelligence, Multimedia Technology, Java and C++ programming, Data Structure and Algorithms. I was responsible for developing all the related material, including lectures, laboratory sessions and assessments. I also supervised more than ten final year projects.

In 2009, I moved to the Department of Electrical and Computer Engineering at the Khalifa University, where I have been teaching Artificial Intelligence Software Engineering for undergrad students, and Pattern Recognition for the PhD program. I have been also in charge of coordinating these courses across the two campuses of Khalifa University. In addition, I have been maintaining homogeneity between the two campuses in terms of course content, delivery, assessment methods and outcome coverage. Furthermore, I have been in charge of supervising both junior and final year projects.

III. Publications

Referred Journals

A. Reda, A. Shalaby, M. Elmogy, A.Abou Elfotouh, F. Khalifa, M. Abou El-Ghar, E. Hosseini-Asl, G. Farb, **N. Werghi**, A. El-Baz, A Comprehensive Non-invasive Framework for Diagnosing Prostate Cancer”. Computers in Biology and Medicine, pp-148-158, 2017

N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, ‘’ Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh’’, IEEE Trans. on Information Forensics & Security, 11 (5), pp.964-979, 2016.

N. Werghi, S. Berretti, A. Del Bimbo, “The mesh-LBP: a Framework for Extracting Local Binary Patterns from Discrete Manifolds”, IEEE transactions on image processing, Vol. 24, No.1, pp. 220-235, 2015.

N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, “Local Binary Patterns on Triangular Meshes: Concept and Applications”, Computer Vision and Image Understanding, 139, pp.161-177, 2015.

S. Berretti, **N. Werghi**, A.D. Bimbo P. Pala, “Selecting stable keypoints and local descriptors for person identification using 3D face scans”, The Visual Computer, springer, pp. 31-41, 2014.

A. Vivekanand, **N. Werghi**, H. Al-Ahmad, “Multi-scale Roughness Approach for Assessing Posterior Capsule Opacification”, IEEE Journal of Biomedical and Health Informatics, Vol.18, No.6, pp.1923 – 1931, 2014.

F. Taher, **N Werghi**, H. Al-Ahmad, C Donner, “Extraction and Segmentation of Sputum Cells for Lung Cancer Early Diagnosis”, Algorithms 6 (3), 512-531, 2013.

Taher, **N. Werghi**, Hussain Al-Ahmad, C.Donner , "Automatic Sputum Color Image Segmentation for Lung Cancer Diagnosis”, KSII Transactions on Internet and Information Systems, (TIIS) 7 (1), 68-80 2012

S. Berretti, **N. Werghi**, A.D. Bimbo P. Pala, Matching 3D Face Scans using Interest Points and Local Histogram Descriptors”, Computers & Graphics, Elsevier, 37 (5), p.509-525 2013.

N. Werghi, M. Rahayem, J. Kjellander. “An ordered topological representation of 3D triangular mesh facial surface: Concept and applications”, EURASIP Journal on Advances in Signal Processing, pp.1-20, 2012.

F. Taher, **N. Werghi**, H. Al-Ahmad, R. Sammouda, “Lung Cancer Detection by Using Artificial Neural Network and Fuzzy Clustering Methods”, American Journal of Biomedical Engineering, Vol. 2, No. 3, pp,136-142, 2012.

M Rahayem, **N Werghi**, J Kjellander, “Best ellipse and cylinder parameters estimation from laser profile scan sections”, Optics and Lasers in Engineering, 50 (9), 1242-1259, 2012.

N.Werghi, “Assessing the Regularity of 3D Triangular Mesh Tessellation Using a Topological Structured Pattern”, Computer-Aided Design and Applications, Vol. 8, No.5, pp.633-648, 2011.

N. Werghi, H. Boukadida and Y. Meguebli, “The spiral facets: A unified framework for the analysis and description of 3D facial mesh surface”, 3D Research, Springer, Vol.1, No.3, pp.1-11, 2010.

N. Werghi, R. Sammouda, F. Alkirbi, “An unsupervised learning approach based on a Hopfield-like network for assessing Posterior Capsule Opacification from digital images”, Pattern Analysis and Applications, Springer, 13 (4), pp. 383-396, 2010.

N. Werghi, Y. Xiao, P. Siebert, “A Functional-Based Segmentation of Human Body Scans in Arbitrary Postures,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 36, No.1, pp. 153-165, 2006.

N. Werghi, “A Robust Approach for Constructing a Graph Representation of Articulated and Tubular-like Objects from 3D Scattered Data”, Pattern Recognition Letters, Elsevier, Volume 27, No.6, pp. 643-651, April 2006.

N. Werghi, “A Discriminative 3D Wavelet-based Descriptors: Application to the Recognition of Human Body Postures”, Pattern recognition letters, Elsevier, Vol.26, No.5, pp.663-677, 2005.

N. Werghi, “Segmentation and Modelling of Full Human Body Shape From 3D Scan Data: A Survey”, IEEE Transactions on Systems, Man, and Cybernetics, Vol.37, No.6, November 2007.

N. Werghi, Y. Xiao, P. Siebert, “Labeling of Three Dimensional Human Body Scans: A Topological Approach”, International Journal of Image and Graphics, Vol. 7, No. 2, pp. 255-272, 2007.

N. Werghi, R.B. Fisher, C.Robertson, A. Ashbrook, “Shape Reconstruction Incorporating Multiple Non-Linear Geometric Constraints, Constraints Journal, Kluwer, Vol.7, No.2, pp. 117-149, 2002.

N. Werghi, R.B. Fisher, C.Robertson, A. Ashbrook. Faithful Recovering of Quadric Surfaces From 3D Range Data By Global Fittin, International Journal of Shape Modeling, Vol.6, No.1, pp.65-78, 2000.

N. Werghi, R.B. Fisher, C.Robertson, A. Ashbrook, “Object Modeling by Incorporating Geometric Constraints”, Computer-Aided Design, Elsevier, Vol.31, No.6, pp.363-399, May 1999.

Conferences

C. Tortorici, **N. Werghi,** S. Berretti, “Defining Mesh-LBP Variants for 3D Relief Patterns Classification”, 7th International Workshop on Representation, analysis and recognition of shape and motion from Image data, Savoi, France, 2017. **BEST STUDENT PAPER AWARD.**

M. Hayat, S. H. Khan, **N.Werghi**, R. Goecke, “Joint Registration and Representation Learning for Unconstrained Face Identification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2767-2776

M.C. El Rai, C. Tortorici, H. Al-Muhairi, M. Linguraru, **N. Werghi**, ‘’3D Constrained local model with independent component analysis and non-gaussian shape prior”, IEEE International Conf. Image Processing, 2016, pp.3204-3208, USA.

I. Reda, A. Shalaby, F. Khalifa, M. Elmogy, A. Aboulfotouh, M. Abou El-Ghar, E. Hosseini-Asl, **N. Werghi**, R. Keynton, A. El-Baz, "Computer-Aided Diagnostic Tool for Early Detection of Prostate Cancer", IEEE International Conf. Image Processing, 2016, pp.3213-3217, USA

N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, Representing 3D Texture on Mesh Manifolds for Retrieval and Recognition Applications, IEEE Conf. Computer Vision and Pattern Recognition, pp. 2521-2530, USA, 2015.

N. Werghi, C. Tortorici, S. Berretti, "Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh", IEEE International Conf. Image Processing, 2015, Quebec City, Canada, 2015.

M.C. El-Rai, **N Werghi**, C Tortorici, H Al-Muhairi, HA Safar, "Landmark Detection from 3D Mesh Facial Models for Image-based", IEEE Conf. Engineering in Medicine and Biology Society, Milan, Italy, 2015.

A.El Khatib, **N. Werghi**, H.Al-Ahmad, "Automatic polyp detection: A comparative study" IEEE Conf. Engineering in Medicine and Biology Society, Milan, pp.2669-2672, 2015.

Z. Qian**, N.Werghi**, et al , "Ensemble Learning for the Detection of Facial Dysmorphology", Proc. IEEE Engineering in Medicine and Biology, Chicago, USA, 2014.

N. Werghi, S. Berretti, A.D. Bimbo P. Pala, "Local Descriptors matching or 3D Face Recognition." IEEE International, Conference on Image Processing, ICIP 2013, Melbourne, Australia, October 2013.

N. Werghi, H. Bhaskar, M. K. Naqbi, Y. Meguebli, H. Boukadida, "The Spiral Facets: A Compact 3D Facial Mesh Surface Representation and Its Applications", Lecture Notes on Communications in Computer and Information Science, Vol.274, Communications, Springer-Verlag. 2013
BEST PAPER AWARD

N.Werghi, C.Donner, F.Taher, H.AlAhamad, "Detection and segmentation of sputum cell for early lung cancer detection" IEEE International Conference on Image Processing, Orlando, USA 2012.

N.Werghi, "An unsupervised learning approach based on Hopefiled-like network for assessing posterior capsule opacifiaction", Proc. International Conference on Machine Vision Applications, pp.416-419, Japan, 2007,

Y. Xiao, **N.Werghi**, P. Siebert, "Topological Segmentation of Discrete Human Body Shapes in Various Postures Based on Geodesic Distance", Proc. International Conference on Pattern Recognition, pp.131-135, Cambridge, UK, August, 2004.

Y. Xiao, **N.Werghi**, P. Siebert, "A Topological approach based on Discrete Reeb-Graph for the Segmentation of Human Body Scans". Proc. IEEE Int. Conference on 3D Imaging and modeling, pp. pp. 378- 385, Canada, 2000

N. Werghi, "Recognition of Human Body Posture from a Cloud of 3D Data Points Using Wavelet Transform Coefficients", Proc. International Conference on Automatic Face and Gesture Recognition,2002,USA,

N. Werghi, R.B. Fisher, A. Ashbrook, C. Robertson, “Faithful recovering of quadric surfaces from 3D range data”, International Conference on 3-D Digital Imaging and Modelling pp.280-289., Los Alamitos, CA, USA, 1999.

N. Werghi, R.B. Fisher, A. Ashbrook, C.Robertson, “Modelling Objects Having Quadric Surfaces Incorporating Geometric Constraints. Proc. European Conference on Computer Vision, pp.185-201, Germany, June 1998.

A. P. Ashbrook, R. B. Fisher, C. Robertson, **N. Werghi,** “Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms”, Proc European Conference on Computer Vision, pp.674-686, Friburg, Germany, June, 1998.

A. P. Ashbrook, R. B. Fisher, C. Robertson, **N. Werghi,** “Segmentation of Range Data into Rigid Subsets using Planar Surface Patches”, . Proc. Int. Conference on Computer Vision, pp. 201-206, Bombay, India, January, 1998.

N. Werghi, C.Doignon, G.Abba, “Ellipse Fitting and Three-Dimensional Localization of Objects Based on Elliptic Features”, Proc. IEEE International Conference on Image Processing. Vol.1, pp.57-60, Lausanne, Switzerland, September 1996.

C.Doignon, **N. Werghi**, G.Abba, E.Ostertag, “Localization of Objects by a Monocular Vision System for Robotic Task”, Proc. International Conference on Intelligent Autonomous Systems (IAS-4); IOS press, pp. 513-520, Karlsruhe, German, March 1995.

Part 2: Research Activities

1. Research projects

This section describes the selected research projects I have been undertaking since my PhD thesis until now. These projects encompass different aspects and applications of 2D/3D image analysis and interpretation. While it seems difficult to put these diverse projects under a single umbrella, I believe that, in many of them, the notion of **representation** has been a common factor. The concept of representation might have different definitions or interpretations, but in the computer vision field, I believe it can be described as *the process of elaborating of an optimal computational framework to encode visual data in order to ensure an effective and efficient solution for a given computer vision problem, within a given application context.*

The introduction will account about the first circumstance in which I have been acquainted to this concept, and more generally the computer vision discipline. We will also briefly elaborate on the concept of representation so as to understand why it has marked a signification portion of my research contributions. Afterwards, we will go through a selection of research projects I have been undertaking, emphasizing for most of them, the representation aspect. For some projects, the notion of representation might not seem capital, yet I think it is actually approached, though in an indirect manner. We will elaborate on this point whenever it is appropriate. We will expose also at the end some ongoing and future research. In the last section we conclude with some reflections.

1.1 Introduction: The Representation concept

My first exposure to the concept of representation was during my PhD thesis, which I have been conducting at the GRAVIR research group (Groupe de Recherche en Automatique et Vision Robotique), in the University of Strasbourg. The thesis was about detecting and locating objects in a robot manipulator scene. It came in the context of research project investigated by the research group about the control of robot manipulator by means of visual feedback. Basically, the system encompasses a manipulator with a camera mounted on its end-effector as illustrated in Fig-1. Video stream is processed by a machine vision module (CYCLOPE which controls the robot axes in closed-loop fashion).

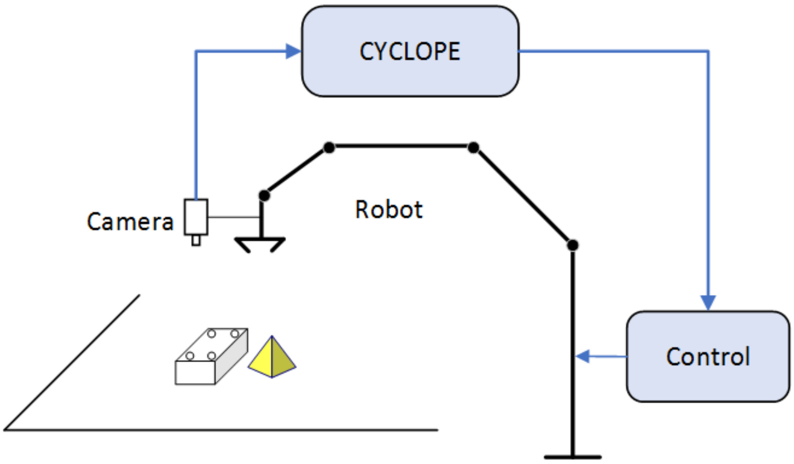


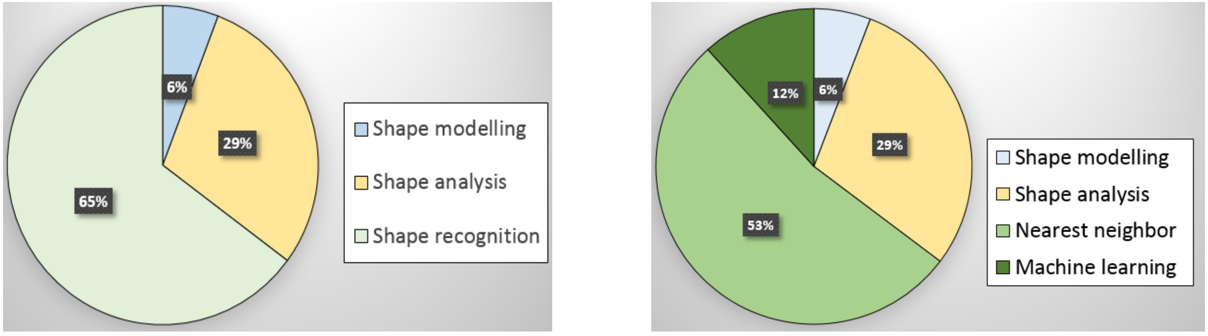
Fig-1: The robot vision system.

During the very first stage of the literature survey, I came across the book "*Vision: A Computational Investigation into the Human representation and Processing of Visual Information*" by David Marr, whom is considered as the founder of the computer vision sciences. Going through the book, I discovered that the computer vision, as a concept, sparked out within a context quite similar to my PhD thesis, long years ago. In the sixties, David Marr, as an expert of the human visual system, has been invited by the artificial intelligence group in the MIT, at Massachusetts, seeking his help for designing a manipulator robot equipped with a visual perception capacity allowing it to sense its environment and enabling it to perform specific basic tasks. Seemingly, this invitation came after several unsuccessful attempts. According to David Marr, the disappointments of the first researchers in robotics resulted from skipping a capital step. They wanted to go directly from the statement of a problem to its solution, without having a prior scientific knowledge to build efficient algorithms on the visual perception. Marr developed his theory around three theses that greatly influenced the thinking of computer vision. The first is that the science of the artificial vision exists, that it must be developed and that once it has progressed sufficiently, the problems posed by the vision in robotics can be solved. The second is that the science of vision is not very different in the case of man's vision of what it will be in that of vision in robotics. The third is that it is as futile to imitate nature in the case of vision as it would have been to build an airplane using the shape and structure of bird feathers. In contrast, the laws of aerodynamics explain the flight of birds and allow to build planes. In his discussion around the early fundamental questions about the potential paradigms for approaching the vision and the algorithms that can emulate the human vision properties (e.g. detection of contours, inference of the depth and the third dimension, etc.), Marr coined the problems related to the nature, the form and the structure of the visual data on which the algorithm acts with the term **"The representation"**. He illustrated the fundamental role of the representation and its implications with arithmetic operations, showcasing how the complexity of such operations increases dramatically when we switch from a given number representation to another. We can exemplify this concept with a simple problem that I love to give to my students in the introductory lecture of my Artificial intelligence class. The example is about performing an algorithm that subtracts 1329 from 2431. The astonishing reaction at a such trivial question quickly dissipates when they are asked to solve this problem using the Roman system for integers representation, i.e. subtract MCCCXXIX from MMCDXXXI. Here, the students realize that the problem is not trivial as they have thought in the beginning, and that the algorithm solving this problem has in fact a quite higher level of difficulty than its counterpart in the first representation. As Marr explained in his book, the difficulty of the problem here does not lie in the nature of the problem itself (perform a subtraction) but rather in the representation adopted for solving it. Marr advocacy on the essential and the subtle role played by the representation can be summarized in this quote "*If one thinks that the purpose of studies in the field of information processing is to formulate and understand particular problems on the treatment of the information, then it is the structure of these problems that is the central problem. not the mechanisms through which they are implemented*".

This exposure to the concept of representation, as advocated by David Marr, at the early stage of my research career, influenced to a large extent the orientation of my subsequent research. Looking at the compilation to the representative works of my research activities up to now, reported in Fig-2, we can easily notice that the majority of my original contribution (column 3) is related to shape representation. It is also noticeable that most of the works related to shape recognition adopted a basic minimum distance classifier (e.g nearest neighbour). This trend reflects the rational that an appropriate representation of the information treated in a given recognition task can bring down the complexity of the classification task, allowing thus the usage of the aforementioned simple classification paradigm. We notice also that machine learning paradigm has not been adopted till a recent date, in the last two works related to 2D face recognition and medical image classification. Actually, the employment of the advance deep learning paradigm, in these works, came in the rational of benefiting of the capacity of such system for deriving an appropriate representation by learning, when the manually crafted representation showed limited potential for addressing related problems.

Reference	Problem	Original contribution	Classification paradigm
[werghi99]	Shape modelling	Model and shape constraints representation	None
[ashbrook98]	Shape matching	Local shape representation (geometric histogram)	NN
[berretti13]	3D face matching	Local shape representation (keypoints)	NN
[werghi06]	3D human body scan segmentation	Global shape representation (Discrete Reeb Graph)	None
[werghi2006a]	Articulated object segmentation	Global shape representation (Discrete Reeb Graph)	None
[werghi02b]	3D posture recognition	3D body shape representation (Spherical harmonics, wavelet descriptors)	NN
[werghi11]	Assessing triangle mesh regularity	Local ordered mesh representation (Ordered Ring Facets, ORF)	None
[werghi11b]	3D face surface analysis	ORF	None
[werghi12]	3D shape analysis	ORF	None
[werghi2015]	Shape texture classification	Local shape representation (mesh-LBP)	NN
[werghi2015a]	Shape texture classification	mesh-LBP	NN
[tortorici2017]	Shape texture classification	mesh-LBP	NN
[werghi2015b]	Shape texture/3D face recognition	mesh-LBP	NN
werghi2016	3D face recognition	mesh-LBP	NN
[aruna14]	Medical image quantification	Surface roughness representation	NN
[hayat17]	2D face recognition	Joint registration and recognition using a deep learning architecture	CNN
[bilal17]	Medical image classification	Transfer learning	CNN
Acronyms: NN: Nearest Neighbour, CNN: Convolution Neural Network, LBP local Binary Patterns			

(a)



(b) (c)

Fig-2: (a): Representative list of my works, the problems they are addressing, the original contributions and the employed classification paradigm when applicable. (b): These works can be mapped into three themes: Shape modelling, shape analysis, and shape recognition. (c): Finer categorization, whereby the shape recognition is split into two sub-categories according the adopted classification paradigm.

1.2 Computer-Aided Design Using Range Data Incorporating Geometric Constraints

During my research fellowship at the University of Edinburgh, I worked on developing novel methods for the semi-automatic construction of CAD object models from 3D images. The objective of this research was to enable CAD designers constructing new models from 3D images by interactively introducing new specifications on the model on the form of geometric constraints on the object model. Current methods at the time of the project investigation employed genetic algorithms. Such approaches facilitate the implementation of the geometric constraints but is ridden with a prohibitive computational time (scale of hours) that the CAD designers cannot afford. We proposed a novel approach blending a vector representation of the object model and the accompanying geometric constraints with an efficient optimization framework. This new paradigm allows to approach the problem with a standard quadratic optimization in which the objective function is defined as follows:

$$F(\vec{p}) + \sum_{k=1}^{m} \left(C_k(\vec{p})\right)^2$$

$$F(\vec{p}) = \vec{p}H\vec{p}; \qquad C_k(\vec{p}) = \vec{p}A_k\vec{p} + B_k\vec{p} + D_k$$

$(\vec{p})$ is the vector encompassing the model parameters. $F(\vec{p})$ is the model-data fitting function, and $C_k(\vec{p})$ is a quadratic vector function representing the kth constraint.

In this paradigm we demonstrated that the above function can be efficiently optimized using the standard Levenberg-Marquardt algorithm, whereby the computation time of the model parameters is brought down from hours to seconds, thus allowing the CAD designer to proceed within an interactive time frame. Findings of this research have been disseminated in [werghi99] and received 138 citations so far.

1.3 Surface correspondences for object registration and recognition using geometric histograms

During my research fellowship at the University of Edinburgh, I also contributed towards developing a new shape descriptor for triangular mesh surfaces. This descriptor, dubbed the geometric histogram, is a 2D accumulator that counts the co-occurrences of two geometrical measurements, namely the angle and the distance between pairs of facets in a given neighbourhood around a central facet (See Fig-3 for a full description). We used this descriptor in surface registration and recognition [ashbrook98]. This work received as many as 107 citations. More recently, we showcased the utility of geometric histograms in 3D face recognition by matching geometric histograms around key points on the facial surface [berretti13].

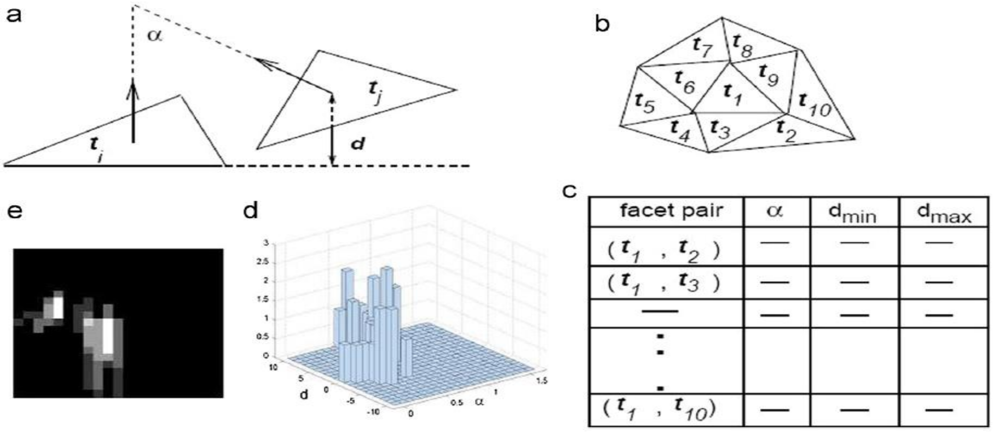
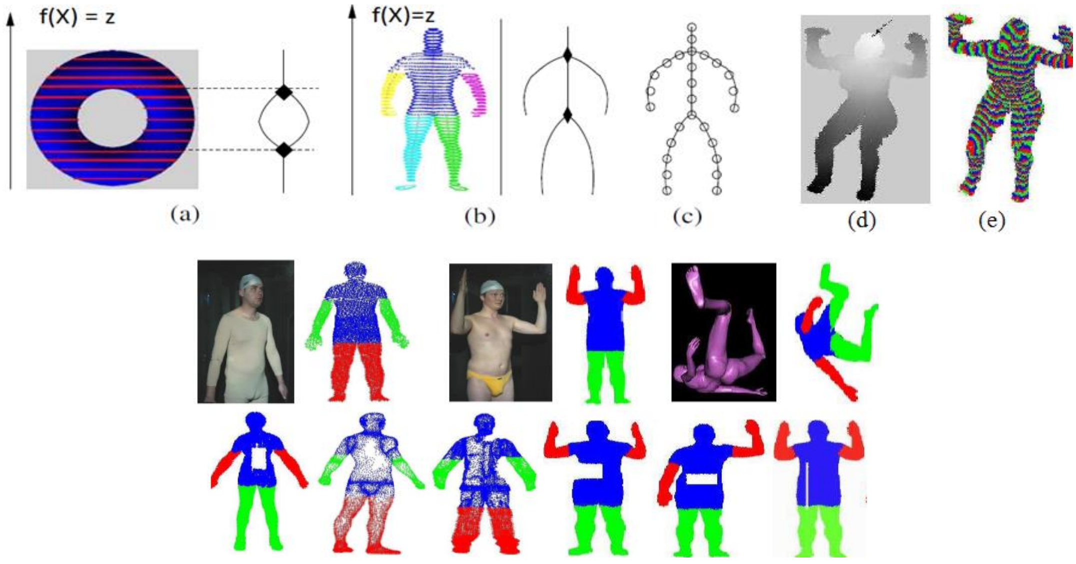


Fig-3 :(a): The geometric measurements used to characterize the relationship between two facets include ti and tj. (b) A facet t1 and its neighbour facets. (c): For each pair (t1, ts), s=1:10, the angle α between the two facets' normals, the minimal and maximal of the perpendicular distance from the plane of t1 to the facet ts are computed. (d): The pairs (α, d) derived from these measurements are entered in a 2D accumulator, thereby obtaining a distribution that characterizes the relationship between the facet t1 and its neighbours. (e): Gray level mapping of the geometric histogram.

1.4 Functional segmentation of objects from point cloud data

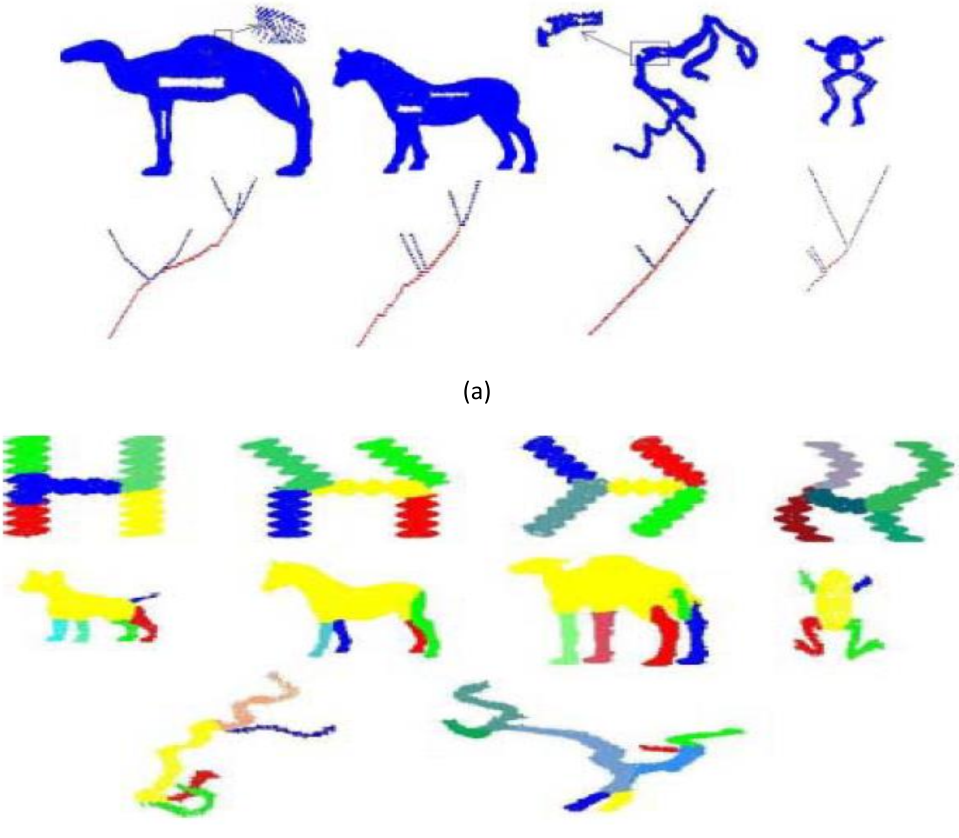
This project, initiated in the department of computing sciences at the University of Glasgow, aimed at designing a robust framework for decomposing a fully scanned object (the scan encapsulates the entire shape) into its functional components. The object comes in a noisy 3D point cloud and exhibiting irregular density. We commenced with the category of human body scans for which, this framework can find applications in the apparel industry, entertainment and medicine. We proposed to address this problem using a topologic framework based on the Reeb-graph. In its continuous form the brancges of the Reeb-graph encode the body parts. Meanwhile in case of its discreet variant, the branches are encoded with nodes representing a connected group of points (level-sets) that share the same value of a given scalar function on the surface manifold (Fig-4). We adopted the geodesic distance from a reference point as a scalar function on the body surface in order to accommodate any arbitrary posture. The noisiness of the data that was causing the ideal discreet Reeb graph (Fig-4-c) to degenerate into a disorganized graph was addressed by defining appropriate topological patterns to differentiate between genuine body joints, and false joints caused by holes and gaps. Such a paradigm makes the segmentation quite robust even for extremely corrupted cases (Fig-4-f). This new paradigm entails the advantage of a linear computational complexity implementation. Part of this research has been disseminated in [werghi06].

In a subsequent work [werghi06a], we generalised this topological framework to accommodate a large class of articulated and tubular-like objects whilst preserving the robustness against data corruption (Fig-5-a) and showcasing its application for point cloud data segmentation for different categories of objects (Fig-5-b).



(f)

Fig-4: (a): Reeb graph of a torus. (b): Reeb graph of a human body shape. (c): Discrete Reeb graph. (d) Geodesic distance function on the body surface. (e) Level sets. (f) Top: Segmentation of HB scans in standing and arbitrary postures. Bottom: Segmentation of corrupted HB scans.



(a)

(b)

Fig-5: (a) Top: Samples of corrupted object points clouds. Bottom: Visualization of topological structure embedded in their associated DRG and encoded as skeleton-like tree structure where each branch denotes a ramification from the main body of the object. (b) Examples of segmented objects.

1.5 Recognizing human body posture from point cloud data

In continuation with the topic on human body scan data, I investigated the problem of recognizing the body posture in a point cloud format. Such semantic information would contribute to bridging the gap between full human body technology and numerous potential applications. One of the proposed approaches was to define an optimal set of 3D descriptors in the sense of Fisher's linear discrimination; that is, set of descriptors minimizing the intra-class distance whilst maximising the inter-class distance. Here, we successfully extended the 2D wavelet shape descriptors developed by Shen and Ip [chen99] to the 3D case. We did this by projecting the 3D spherical harmonics transformation applied to point cloud into a 1D space, obtaining thus a sort of radial function encoding the posture shape. From this function, we derived a set of wavelet transform coefficients (WTCs) computed suing an orthogonal family of wavelets. In the last stage, we selected the best discriminative descriptors, from the WTCs using a discriminative power criterion based on the Fisher's linear discrimination principle. Tested on a set of 32 postures (illustrated in Fig-6), our descriptor representation outperformed two other standard features, namely, the 3D Fourier coefficients (FC) and the 3D Zernike coefficients (ZC). In particular, our proposed features exhibited a remarkable capacity in differentiating between close postures. Table-1 depicts some relevant examples. We disseminated this work in [werghi02b, werghi05].

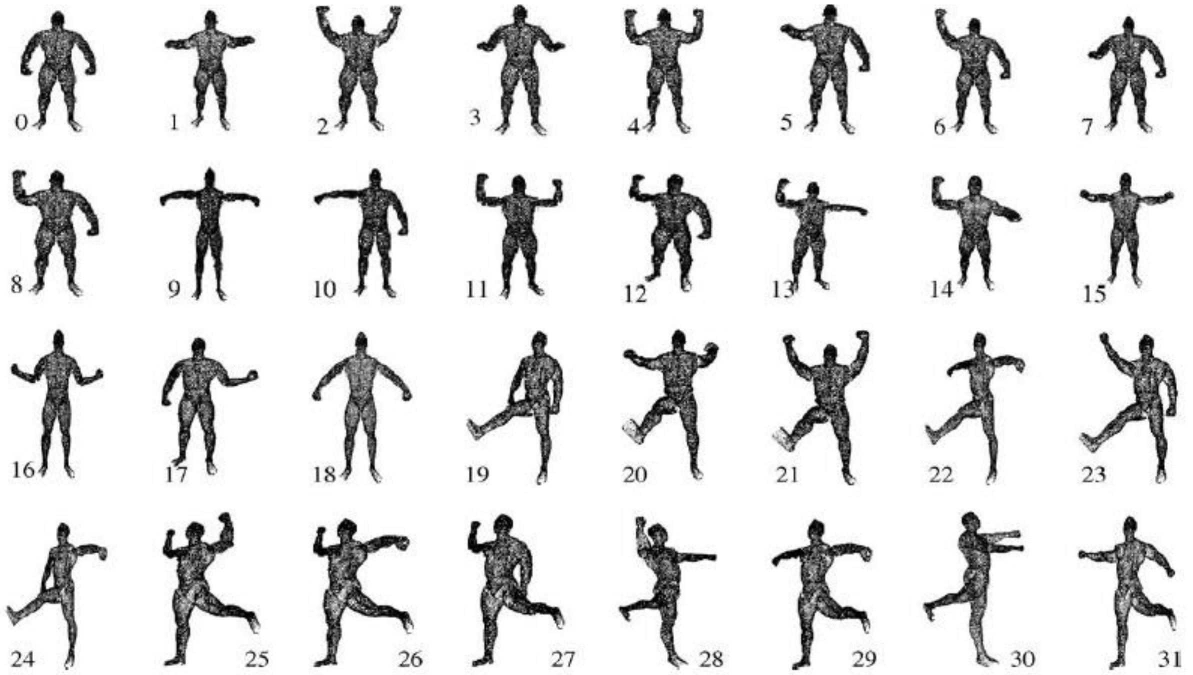


Fig-6: 32 posture models.

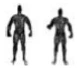
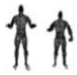
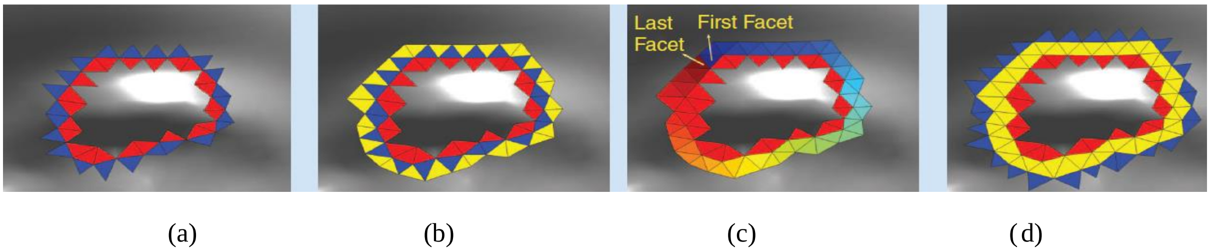
	(0, 7)	(2, 4)	(2, 11)	(3, 7)	(4, 11)	(8, 12)	(9, 15)	(20, 21)
								
FC	16	2	13	0	72	54	82	6
ZC	64	3	438	8	356	7	544	136
WC	201	306	984	45	635	39	533	477

Table-1: Examples of pairs of close postures and their related separations distances for each of the FC, ZC and the WC.

1.6 Shape analysis on the mesh manifold: representation tools and applications

A large portion of my research has been dedicated to the design of an appropriate representation for analysing triangular mesh manifold data. The concept of investigating such a problem was ignited while revisiting the concept of geometric histogram that was previously described in Sec-1.3. The endeavour was to extend this descriptor into a multi-scale analysis tool. However, one major impediment was the lack of intrinsic order in the triangular mesh manifold. In its standard representation, a triangular mesh manifold is encoded in an array of facets and array of vertices. The arrangement of these facets and vertices in the arrays is totally random and thus, do not exhibit any useful spatial structure or order. This is in contrast with the case of pixels in image for which, the arrangement made in row and columns is persevered within the 2D array data structure. This fundamental structural difference between the 2D image and the mesh manifold makes the extension of the 2D image analysis techniques into the mesh manifold quite problematic, unlike its counterpart in the 1D-to-2D case. We thought that Investigating a local, and if possible, global ordered as well as structured support for encoding triangular mesh manifold will open-up new outlets in the analysis of this modality, which currently finds application in myriad fields such as animation, medical imaging, computer-aided design, and remote sensing. We therefore deemed such a new niche of research quite relevant for bridging the gap between 3D shape digitalization technology and its related applications.

In this project, we proposed a novel representation dubbed Order Ring Facets (ORF) [werghi12]. This representation was inspired by observing the arrangement of facets laying on a convex contour (each face has exactly one edge on that contour) on the mesh; in other words, facets that share only one edge with that contour. We noticed that these facets can be segmented into two categories, namely, facets pointing outside the contour (Fout facets), and facets pointing inside the contours (Fin facets). In Fig-6-a, these facets are highlighted in blue and red, respectively[1]. Setting an initial circular arrangement of the Fout facets and considering the pair-wise adjacency between Fout and Fin facets, we repeated the process of filling the gap between each pair of consecutive Fout facets, with a sequence of facets that shared a vertex on the contour (yellow facets in Fig-7-b). In doing so, we obtain a ring of ordered facets, as illustrated in Fig-7c. It is also possible to build a subsequent ring using the outer contour of the existing ring, for which the Fin facets are represented by those obtained by the aforementioned gap filling procedure. Here, the Fout facets are those that are one-to-one adjacent to them (Fig-7-d). By iterating this procedure, we construct a sequence of concentric rings whereby the circular ordering is propagated during each iteration.

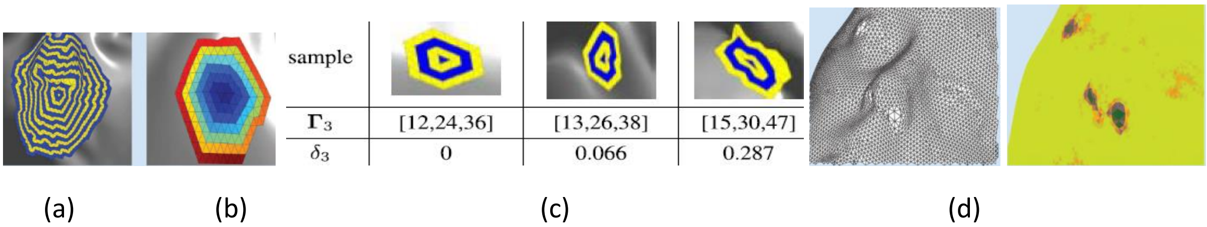


(a) (b) (c) (d)

Fig-7: Ordered Ring Facets construction: (a) Facet laying on a convex contour are segmented into Fout (blue) and Fin (red) which point outside and inside the contour, respectively. (b) We fill the gap between each pair of Fout facets after ordering them in a circular fashion (e.g. clockwise). (c) We obtain a ring of facets that are ordered in circular fashion. (d). Four facets are extracted from outer contour of the constructed ring to be used in the subsequent ring construction.

[11] For sake of simplicity, the notation used here is slightly different than the one mentioned in related papers.

The ORF can be generated around a central facet when setting its edges as the initial contour. The ORF thus forms and denotes an ordered and structured neighbourhood of that facet (Fig-8-a). The facets can also be arranged in spiral-wise fashion (Fig-8-b).



sample			
Γ_3	[12,24,36]	[13,26,38]	[15,30,47]
δ_3	0	0.066	0.287

(a) (b) (c) (d)

Fig-8. (a) ORFs generated around a seed facet. (b) Facet arranged in a spiral-wise fashion. (c) Samples of 3-ring ORFs and their Γ3 and δ3. (d) Computation of δ3 on a facial mesh surface; the irregularities detected at the dark spots areas can be noticed.

One of the interesting aspects of the ORF is that in an ideal regular mesh, (where the valence of each vertex is six), the number of facets ***n*** across the rings follows the following arithmetic progression ***n(i+1) = n(i) + 12***. In a 3-ring ORF, for instance, the progression is [12, 24, 36]. Such a property facilitated the establishment of a simple criterion for evaluating the local regularity of the mesh. Here we proposed the following criterion:

$$\delta r = \|\widehat{\Gamma} r \; - \Gamma r \,\| / \|\widehat{\Gamma} r \,\|$$

where $\widehat{\Gamma} r$ is vector that represents the ideal sequence of the number of facets across *r* rings, and Γr denotes the actual sequence. Fig. 8-c depicts examples of 3-ring ORFs exhibiting different Γ3 and δ3, along with an instance illustrating the detection of irregular tessellation (Fig-8-d). In addition to its simplicity, this criterion has a low computational complexity when compared to other standard methods. We showcased this performance in [werghi11].

We also demonstrated that facets across at ORF rings are located virtually at the same geodesic distance from the seed facet and hence, can be used to extract iso-geodesic contours and compute iso-geodesic distances on the mesh with linear algorithmic complexity O(n), where n denotes the number facets in the ORF rings. The ORF framework has been also adapted in other several facial mesh surface-processing tasks, such as nose detection, cropping, compression, and alignment (see Fig-9). Additional details on these tasks are outlined in the VSAPP2011 conference paper [Werghi2011b], which received the best paper award, as well as in the journal paper [werghi2012].

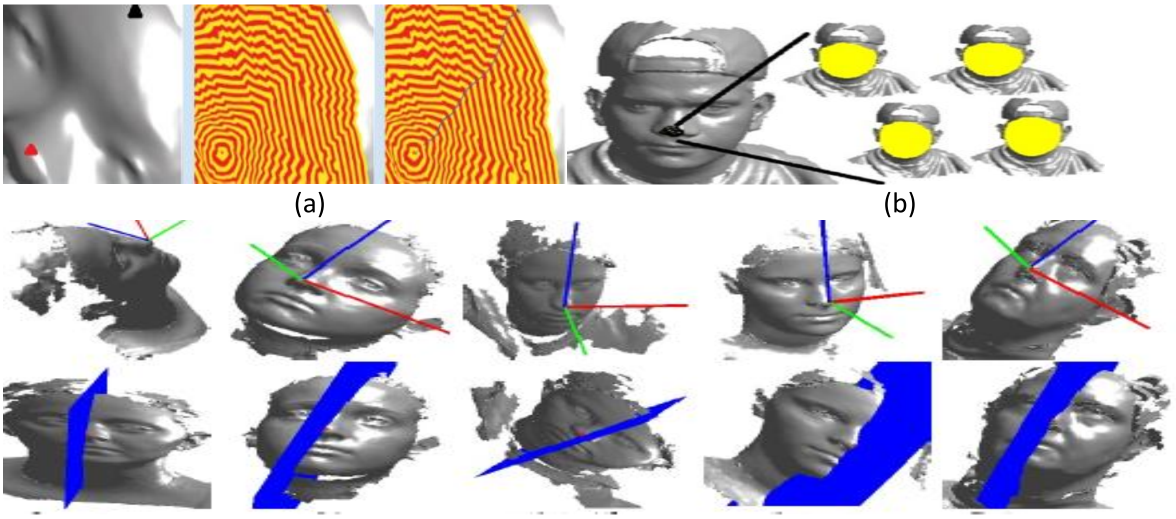


Fig-9. (a) Computation of the geodesic path: From a source facet, the ORFs are expanded till they reach the destination facet. The geodesic path is then extracted by tracing back the source facet. (b) Face cropping is performed by generating several ORFs from the nose tip neighbourhood and merging them. (c) Examples above illustrate facial poses and symmetry planes and are derived by applying the principal component analysis on the ORF rings.

The ORF formed also the foundation for extending the local binary pattern (LBP) [ojala02] to the mesh manifold. In its simplest form, an LBP denotes an 8-bit binary code obtained by comparing a pixel's value with the value of each pixel in its 3x3 neighbourhood. The outcome of this comparison is 1 if the difference between the central pixel's value and its neighbour pixel's counterpart is less or equal than a certain threshold; and 0 otherwise. The local description can be refined and extended at different scales by adopting circular neighbourhoods at diverse radii and using pixel sub-sampling (see Fig-10).

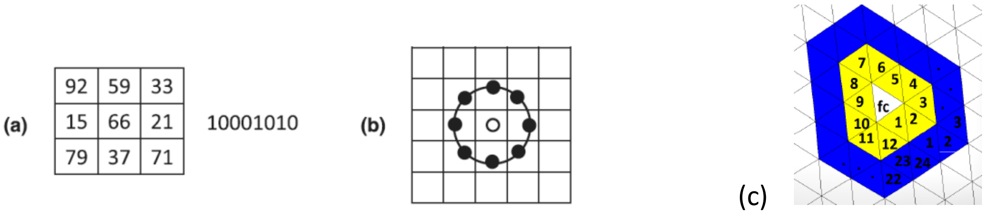


Fig-10: (a) Computation of basic LBP code from a 3x3 neighbourhood. Each pixel, starting from the upper left corner, is compared with the central pixel to produce 1 if its value is greater than a threshold; and 0 otherwise. The result is an 8-bit binary code. (b) Example of a central pixel with circular neighbourhood. (c) Example of 2-ring ORF generated from the central facet fc.

The circular arrangement of facets across the ring sequence within the ORF structure allows a straightforward adaptation of the LBP into the mesh manifold in its generalized multi-resolution format. Let ***h(f)*** denote a scalar function defined on the mesh that can incarnate either a geometric (e.g., curvature) or photometric (e.g., colour) information. The mesh-LBP operator is defined as follows:

$$meshLBP_m^r(f_c) = \sum_{k=0}^{m-1} s(h(f_k^r) - h(f_c)) \cdot \alpha(k)\ ,\ s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where ***r*** is the ring number, and ***m*** represents the number of samples computed at each ring. The parameters ***r*** and ***m*** control the radial resolution and the azimuthal quantisation, respectively. The discrete function ***α(k)*** is introduced for the purpose of deriving different LBP variants. For example,***α(k) = 2^k*** results in the mesh counterpart of the basic LBP operator firstly suggested by Ojala et al. [27]; with ***α(k) = 1***, we obtain the sum of the digits composing the binary pattern. For ***m=12***, the number of mesh-LBP pattern values is 13 and 4096, for ***α(k) = 1*** and ***α(k) = 2^k***, respectively.

The obtained operator, which we dubbed mesh-LBP, presents a novel framework for analysing ***3D geometric texture*** as a property of the surface, distinct from the global shape, and characterized by the presence of repeatable geometric patterns (see Fig-11-a-b-c). These patterns can be viewed as geometric corrugations of the surface that change the local smoothness and appearance of the surface rather than altering the overall shape. Fig-11-d depicts some examples of mesh-LBP pattern computed for two different types of surfaces.

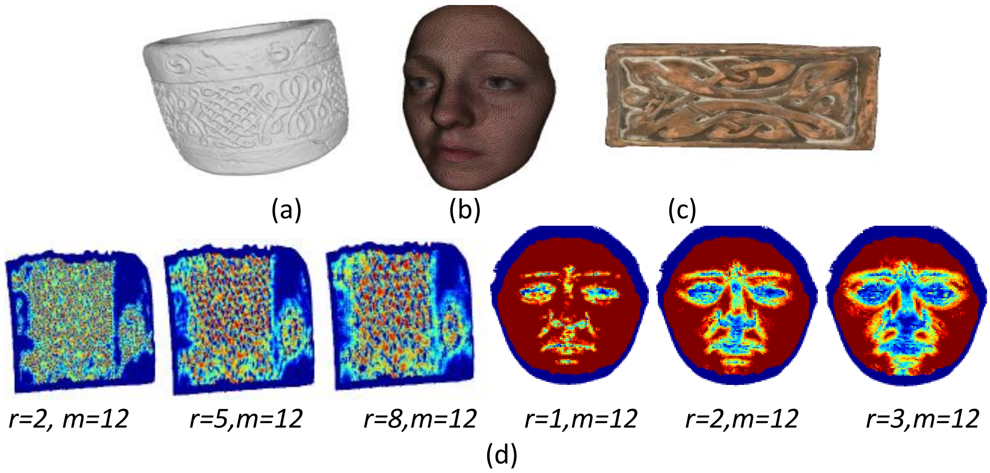


Fig-11: Example 3D objects with different 3D textures: (a) 3D geometric texture characterized by repeatable patterns of the mesh surface; (b) 3D photometric texture attached to the triangular mesh. In this case, the textural information is most present in the photometric appearance of the mesh rather than in the geometric appearance; (c) Combination of 3D geometric and photometric texture on 3D mesh manifold. (d) Examples of three mesh-LBP patterns that are computed on a texture and facial surfaces. The scalar function used is the Gaussian curvature and the mean curvature for the first and second surface, respectively.

We describe surface variations on the manifold over a given surface area with a 2D histogram by stacking 1D histograms of the mesh-LBP values across different radial resolutions ***r.*** Fig-12 depicts some shape texture samples that were collected from a public object dataset.

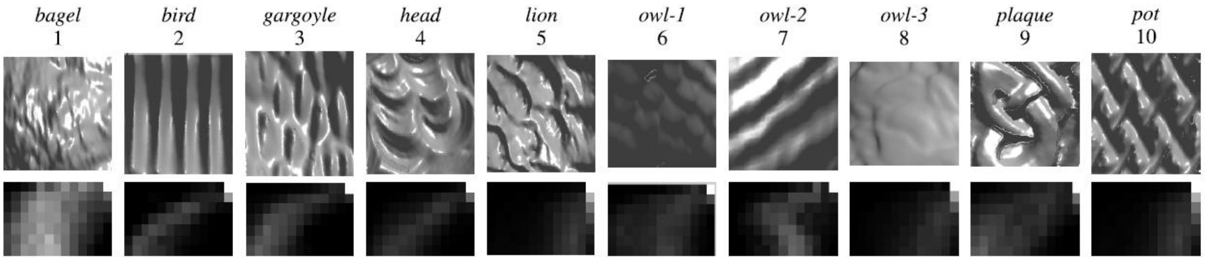


Fig-12: Top: 3D texture samples from ten 3D texture classes. Bottom: The corresponding histograms obtained with ***h(f)*** set to the mean curvature, ***α(f) =1*** and mesh-LBP parameters ***r=1:7*** , ***m=12.*** For each r value, we compute a 13-bin 1D histogram accumulating the frequency of the different mesh-LBP values. By stacking-up the 1D histograms, we obtain a 7x13 2D histogram. Bottom: The 2D histograms are illustrated as a gray-level images.

The mesh-LBP framework preserves the simplicity and the elegance characterizing the original LBP while accommodating the extension of its different variants developed for 2D image analysis to the mesh manifold including closed surfaces. It also relieves the surface data from normalization and registration procedures that are necessitated when using depth images.

Naturally, exploiting mesh-LBP for the classification and retrieval for 3D shape texture patterns was the first application we investigated. In this context, we demonstrated the superiority of mesh-LBP descriptors over other standard shape descriptors in terms of texture discrimination and retrieval.
Fig-13 depicts the confusion matrix pertaining to the classification performance of the mesh-LBP and other standard descriptors when applied to the 10 classes as depicted in Fig-12.

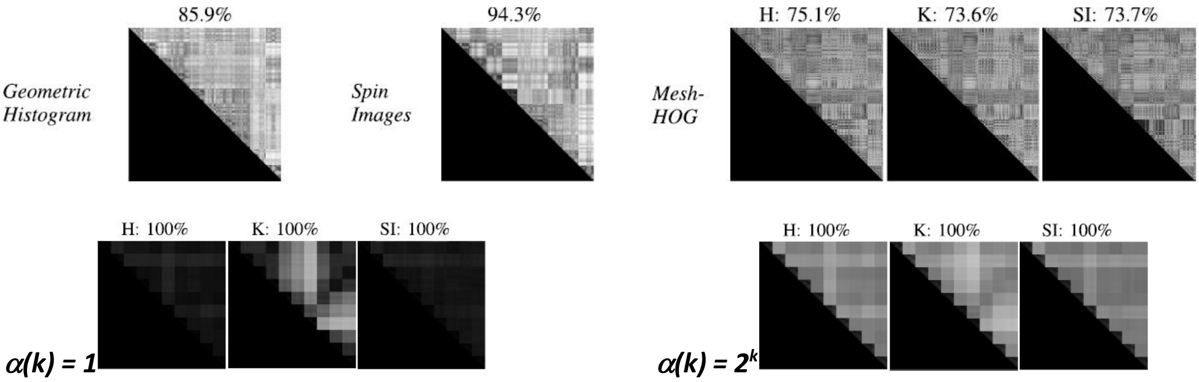


Fig-13: Confusion matrices reporting the distances between all the instances of ten classes shown in Fig-12 (30 instances per class).Top: Diffusion matrix obtained with Geometric histograms, the spin image and the mesh-HOG using the mean curvature (H), the Gaussian curvature (K) and the shape index (SI). Bottom: Confusion matrices obtained with the mesh-LBP using the same descriptors, H, K, and SI as scalar functions. The classification accuracy is reported at the top of each matrix.

The ability of the mesh-LBP to retrieve shape texture has been also compared with other standard descriptors. The experiment entails the search of each probe within a gallery surface, and subsequently assessing the detection as well as retrieval capacity of the different descriptors. Fig-14 shows some samples of texture retrieval results. The results were obtained with ***$\alpha(k) = 2^k$*** using six different scalar functions on the mesh. Additional details on the mesh-LBP are mentioned in [werghi2015, werghi2015a]. Moreover, we made the Mesh-LBP code available for the scientific committee[2].

[2] Code avalable at : http://uk.mathworks.com/matlabcentral/fileexchange/authors/538543

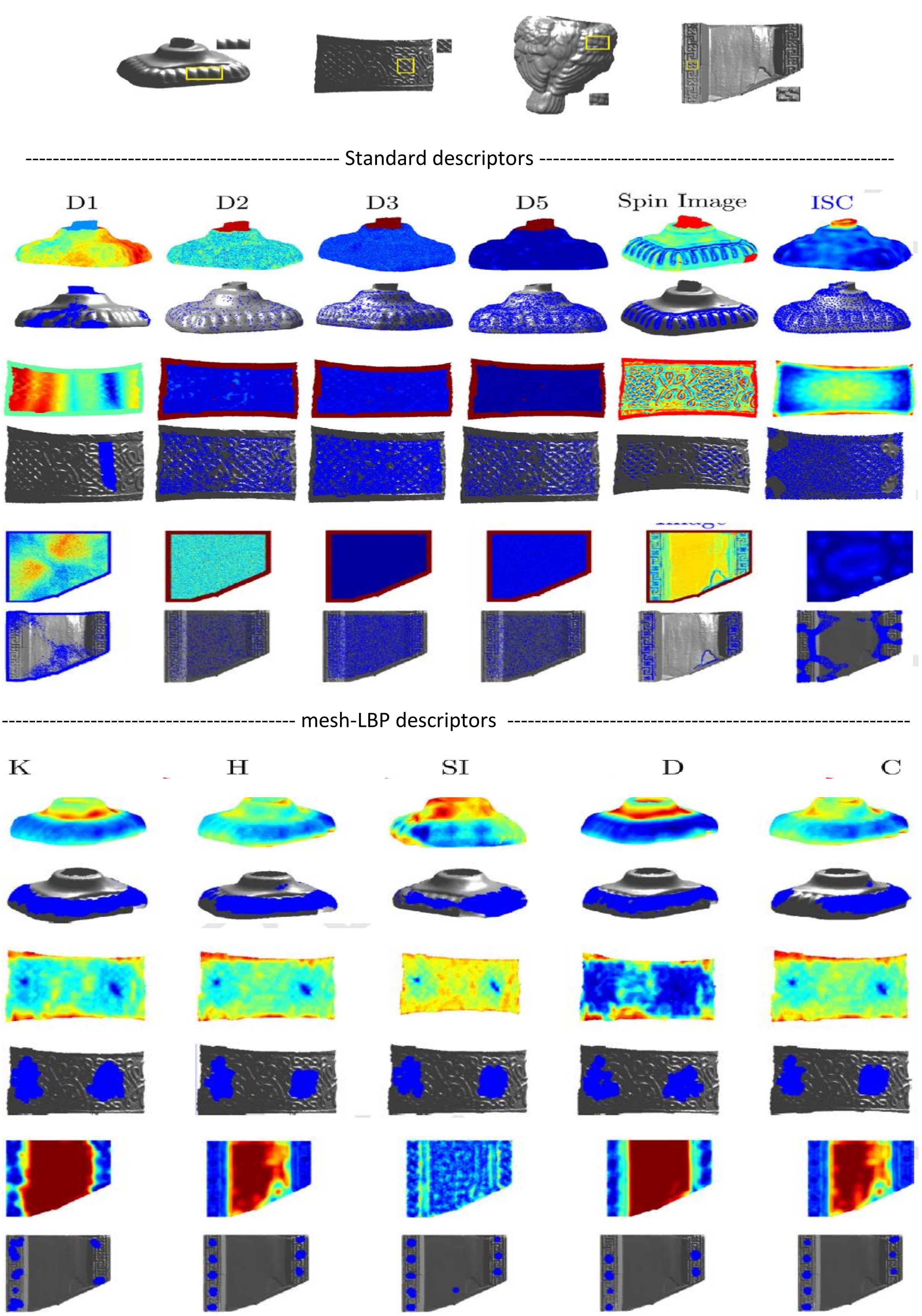


Fig-14: Examples of texture retrieval results. The figure illustrates the mapping of the distance between the probes (framed samples in the first row) and the different object surfaces in the gallery. The detected regions are highlighted in blue.

In a recent work [tortorici2017], we extended several LBP variants, such as the Median LBP, the centre-symmetric LBP, the completed LBP, among many others, to the mesh manifold. We extracted a total set of 48 novel mesh-LBP variants which were evaluated on a public dataset of 3D textured surfaces presented at the SHREC'17 contest [biasotti17]. The dataset consists of 720 surfaces grouped in 15 classes of 48 elements each. Every class was created by acquiring a single pattern in different poses and applying a series of surface deformation and mesh tessellation alteration (Fig-15). The testing of our texture retrieval method using the novel variants demonstrated a quite superior performance when compared to other state-of-the-art methods. We presented this work in the most recent International Workshop on Representation, Analysis and Recognition of Shape and Motion from Image Data 2017 for which, our PhD Student Claudio Tortorici received the best student paper award.

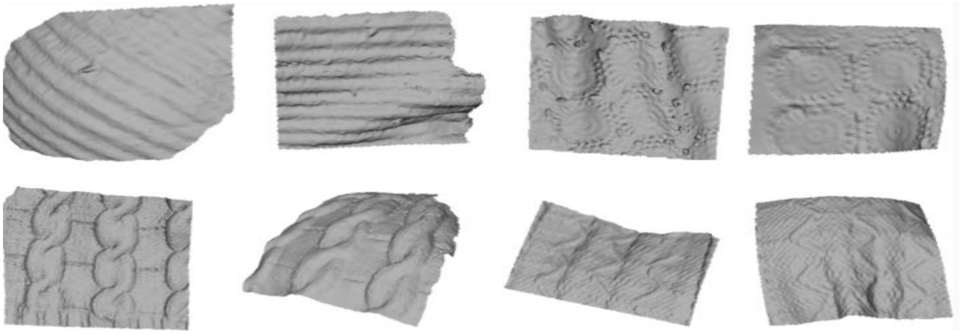
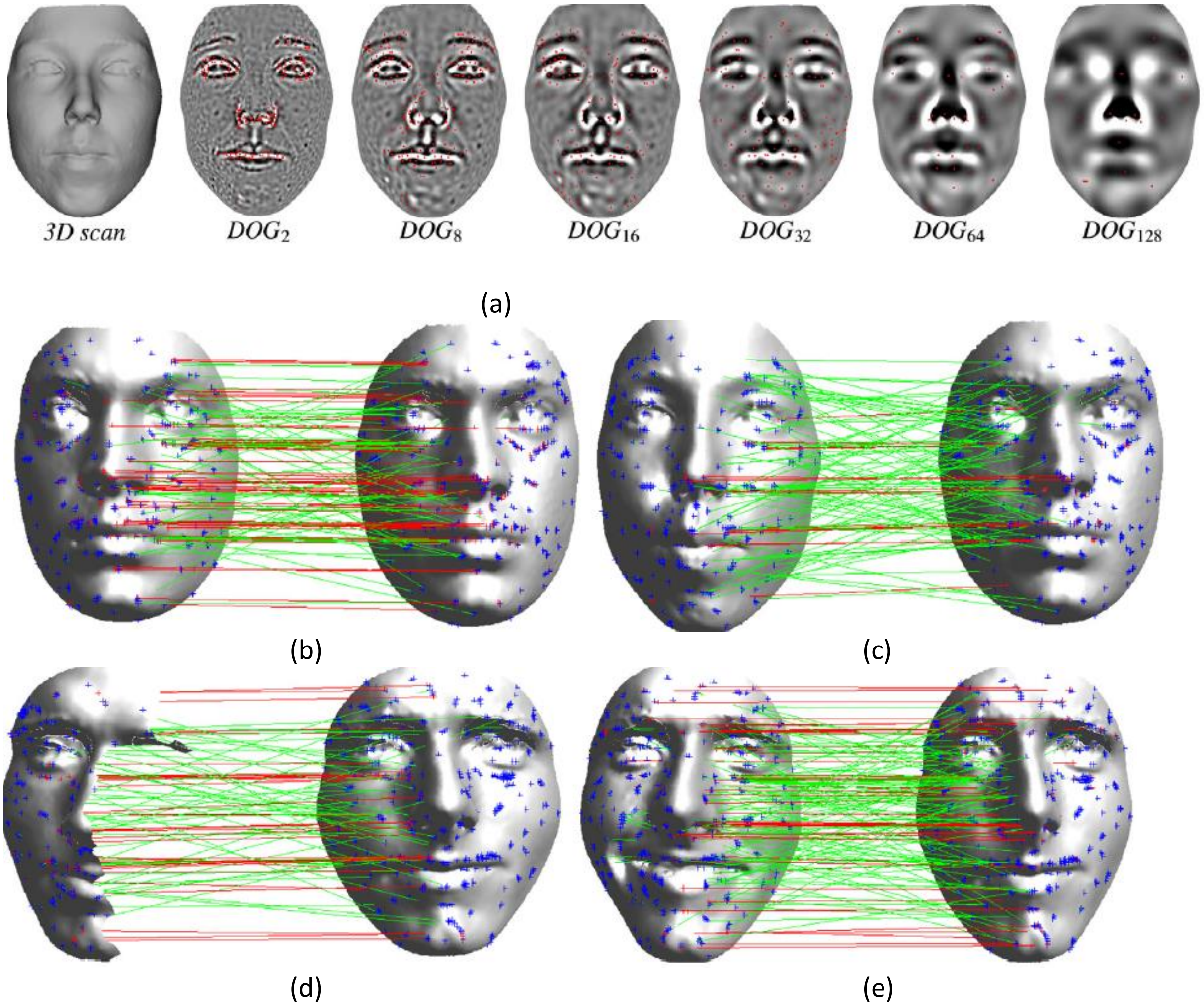


Fig-15: surface samples from the SHREC'17 dataset.

1.7 3D Face recognition

Face has affirmed itself as one of the most important biometric traits owing to the fact that facial images or videos are collectable in an easy and non-intrusive manner. In fact, while their accuracy might not be as high as fingerprint and iris, they do have the capital advantage of not requiring the cooperation of the subject, a requirement which is of great interest in several scenarios such as surveillance and gateless access applications. Notably, automatic face recognition confronts several challenges including pose changes, illumination variations, facial expressions and occlusions. In order to resolve these problems, face recognition using 3D scans has been proposed as an alternative or complementary solution to conventional 2D face recognition approaches that use still images or videos. Nowadays, most face scans (if not all) encompass the facial shape in the form of a triangular mesh surface and the facial appearance in the form of a 2D image mapped to the mesh.

In a first contribution in the 3D face recognition, we developed an original approach based on the idea of capturing local information of the facial surface around a set of 3D keypoints that were detected at multiple scales in accordance to differential surface measurements. The keypoints detection is performed by adapting the meshDOG algorithm to the facial case (Fig-16-a). Subsequently, 3D local descriptors are extracted at the neighbourhood keypoints as local signatures and are employed within a keypoints matching scheme. We compared three types of local descriptors namely, Histogram of Gradients (HOG), the Histogram of Orientations (SHOT), and the Geometric Histograms. The matching scheme does not make any assumption about the correspondence of detected keypoints to specific landmarks on the face; therefore, it can support the comparison of probe and gallery scans even in cases where probe scans represent merely a part of the face. To improve the accuracy of keypoints correspondences, we introduced a spatial constraint using the RANSAC algorithm. The experiments conducted on BU-3DFE, Gavab and FRGC v2.0 datasets proved that our method is capable of competing with state-of-the-art method, evidencing a distinct advantage in cases of probes involving large missing parts. Further details about this work are in [berretti13].



(a)

(b) (c)

(d) (e)

Fig-16: Top: Keypoints detected using the meshHOG at different scales. Matching of scans of the same and different subjects is reported in (b) and (c), respectively. All the detected keypoints are illustrated as "+". Lines indicate matching keypoints (in green), and inliers matching after RANSAC (in red). In case the scan involves the same subject in (b), 61 inlier matches are identified. For scans of different subjects in (c), 18 matches are detected. (d) Identified a match between partial face scan and neutral full scan. (e) Identified a match between a facial expression scan and a neutral full scan (faces are flipped for sake of visualization).

In the second research, we investigated the adaptation of mesh-LBP concept for 3D face recognition. Our proposed paradigm is inspired from the standard LBP-based face representation propounded by Ahonen et al [ahonen06] in the context of 2D face recognition. In their method, the facial image was divided into a grid of rectangular blocks after which, histograms of LBP descriptors are extracted from each block and concatenated to produce a global signature of the face (Fig-17). In order to extend this scheme to the face surface manifold, we part the facial surface into a grid of regions (the counterpart of these blocks in the 2D-LBP), compute their corresponding histograms, and then group them into a single structure. The proposed method entails therefore following stages: 1) Construction of a grid of points on the face surface to obtain an ordered set of regions; 2) Computation of an histogram of the mesh-LBP descriptors over the surface regions centred at each point of the grid; 3) Aggregation of the regional histograms into a structure encoding either a global or partial description of the face; 4) Performing facial matching.

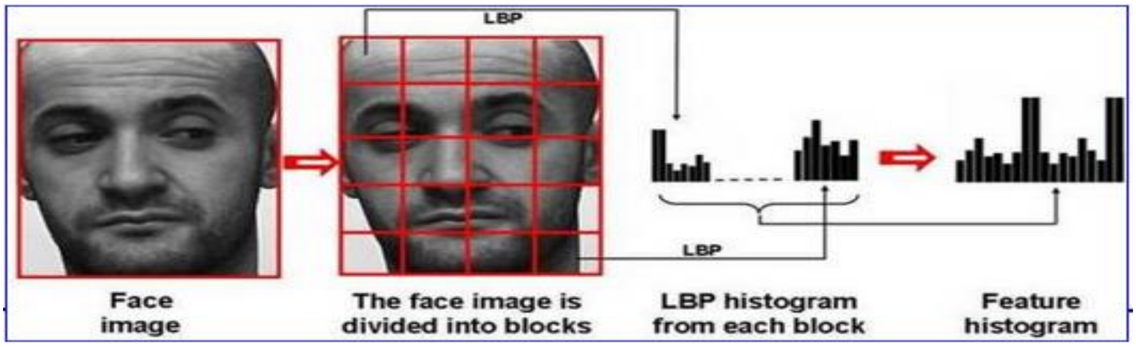

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

Fig-17: LBP-based 2D face signature (source: www.scholarpedia.org/article/Local_Binary_Patterns).

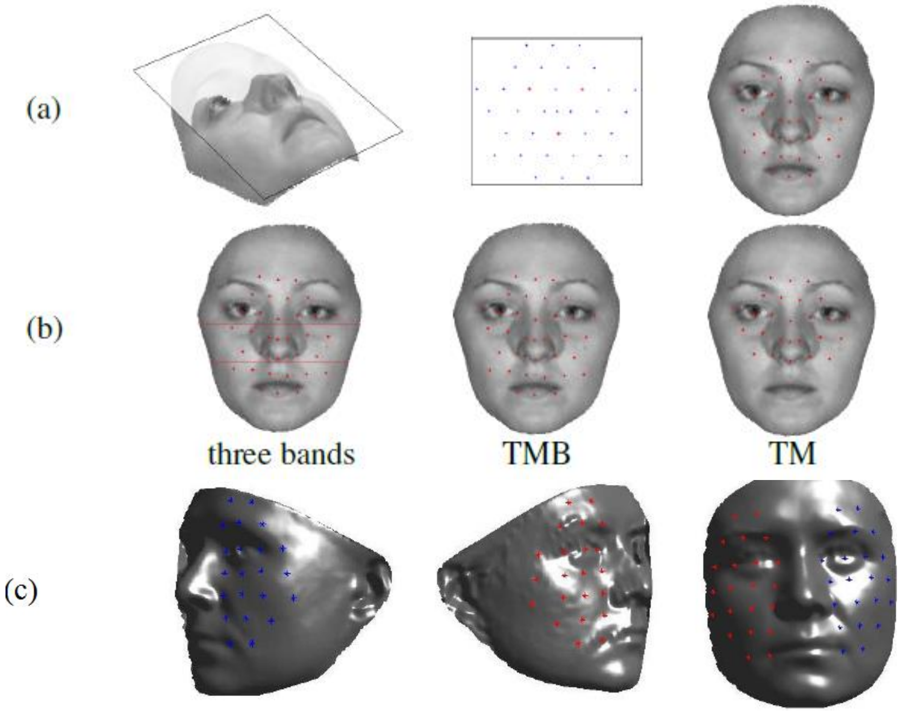


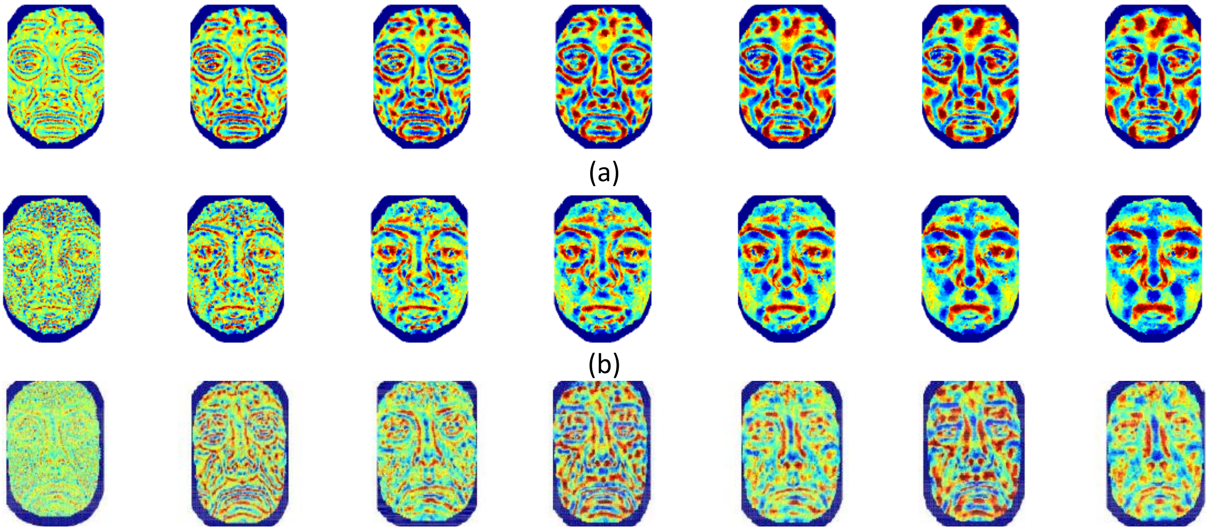
Fig-18: Construction of the face grid; (b) on the left scan, partition of the grid points to a top (T), middle (M) and bottom (B) band. Middle scan: All the points in the three bands (TMB), right scan: Only the points in the top and middle (TM) bands are depicted. (c) Construction of the partial grid on two rotated probe scans and a gallery scan.

Grid-point construction

In the first step (Fig-18-a), we computed the plane formed by the nose tip and the inner-corner of two eyes as landmark points. We choose these three landmarks because they are the most accurate detectable landmarks on the face in addition to their robustness to facial expressions. From these three landmarks, we derive, via a simple geometric computation, an ordered and regularly spaced set of points on that plane (shown in the middle of Fig-18-a; the nose tip and the inner corner of eyes are marked in red). Subsequently, the plane is tilted slightly by a constant amount to augment its alignment with the face orientation. Next, we projected this set of points on the face surface, along the plane's normal direction. The outcome of this procedure is an ordered grid of points which defines an atlas for the regions dividing the facial surface (Fig-18-a-right). To accommodate the effects of facial expressions, we segmented the grid points into three bands (Fig-18-b), dubbed as top (T), middle (M) and bottom (B), so that we can consider the full grid (TMB) or the top and middle bands (TM) only during the face matching. The TM option allows us to neutralise, albeit to a certain extent, the shape changes manifesting at the lower part of the face caused by the mouth in particular. The TMB and the TM grids comprises of 35 and 26 points. Around each grid point, we extract a neighbourhood of facets using the ORF centred at that grid point. For a yaw rotated pose resulting in a partial scan that does not allow for the extraction of either of the two eyes' inner-corner landmarks, we adopted a lateral grid, and constructed upon the plane defined by one eye inner-corner, an eye outer corner and the nose ridge. Covering one side of the face, the grid contains 22 points. For the gallery scans, we constructed TMB grid as well as the left and right lateral grids (Fig-18-c).

Regional Mesh-LBP histograms computation

In the second step, we compute a multi-resolution mesh-LBP descriptor for each facet within the regions, considering different shape-valued and appearance value functions on the mesh. For instance, Fig-19-a-b depicts the mesh-LBP patterns computed for the Mean curvature and the gray level at a different radial resolution, for $\boldsymbol{\alpha(k) = 1}$.


(a)

(b)

(c)

Fig-19: Mesh-LBP computed for an azimuthal resolution m=12 and seven radial resolution r=1..7, with the scalar function Mean curvature (a), Gray-level (b), as well as by interleaving the mesh-LBP of the Mean-curvature and the gray-level (c).

In addition, we considered three different fusion variants of the mesh-LBP patterns. In the first variant, we concatenated the two mesh-LBP regional histograms corresponding to a shape and a gray-level function. For example, considering an azimuthal quantization $m = 12$ and $\boldsymbol{\alpha(k) = 1}$**,** we obtained a 13-bins histogram for each function, thus leading to a one-dimensional 26-bins histogram for each radial resolution r; that is a r ×26 histogram (Fig-20-b). In the second variant, we used a 2-D accumulator that accounts for the co-occurrences of mesh-LBP patterns corresponding to a shape and gray-level function. For the aforementioned parameters' values, we obtained an r × 13 × 13 histogram (Fig-20-c). In the third variant, the fusion was performed at the LBP pattern level, instead of the histogram level, as for the first two. Here, the mesh-LBP pattern was constructed by interleaving digits from the shape function mesh-LBP with a gray-level mesh-LBP (Fig-19-c).

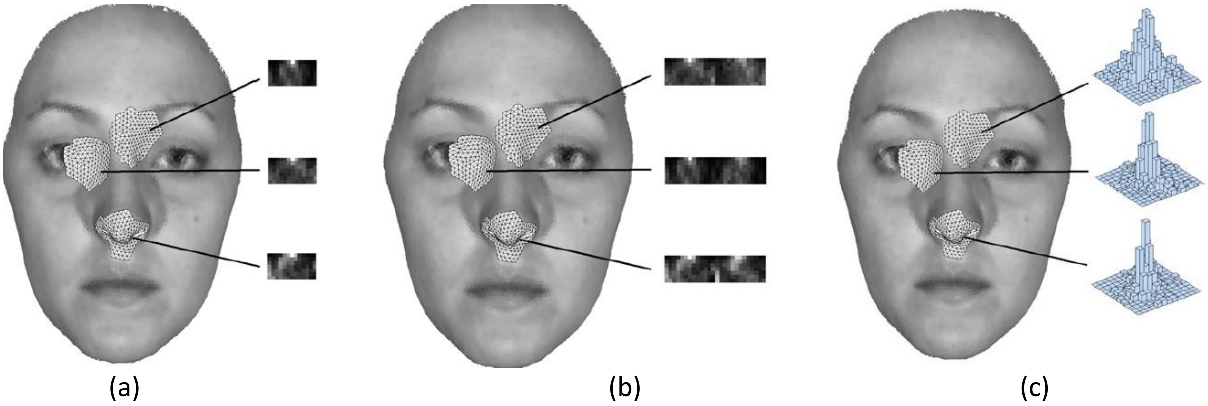


Fig-20: Examples of regional histogram variants obtained with m = 12,r =1..7, and $\alpha(k) = 1$. (a): 7 × 13unimodal histogram corresponding to a shape function. (b): 7 × 26 histogram obtained by concatenating two 7×13 histograms corresponding to a shape function and a gray-level function. (c) A 2D section of a 7 × 13 × 13 histogram obtained with a shape function and a gray-level function.

Global Face Histogram

The global histogram that represents the entire face signature is constructed by aggregating the set of regional histograms computed over the grid regions (Fig-21). Given the 35 regions derived from this face grid, the size of the global histogram in a non-fusion mode is 35 × 13× 7 and 35× 1125× 7 for $\alpha(k) = 1$ and $\alpha(k) = 2^k$, respectively. Using the first fusion variant, the size of the global histogram is 35 x 26 x7. A video demonstrating the construction of a global histogram can be watched in the following link: https://www.youtube.com/watch?v=8UBsIJRKWPM

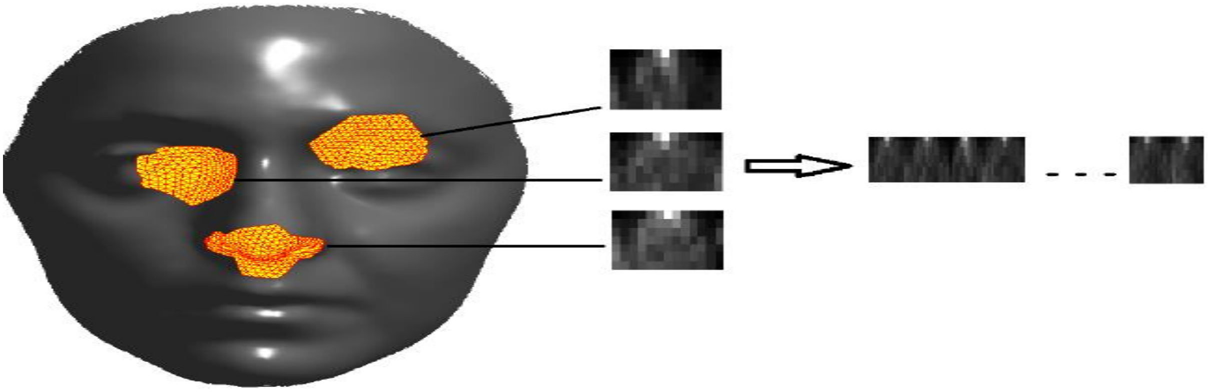


Fig-21: Global histogram is computed by concatenating the regional histograms.

Face Matching

We performed face matching using a very basic, minimum distance classifier that employs a simple distance metrics between histograms (e.g. χ2 and cosine distance). Our aim was to demonstrate the discrimination capability of the mesh-LBP face signature in addition to its great ability to be incorporated in face recognition even without employing machine learning classifier. The performance of our face recognition method was assessed with the BU-3DFE and Bosphorus datasets. The experiments conducted with BU-3DFE database underpinned the enhancement in recognition performance made possible by our fusion framework. They also proved its superiority with regard to the closest approach. Results obtained using the Bosphorus database demonstrated a competitive accuracy as compared to the state of the art solutions, with an increment for some specific expression category subsets. For instance, it attained 100% accuracy for the neutral, surprise, the Upper Face Action Unit, and the Combined Action Unit categories within the Bosphorus databases. The Yaw pose, for which, we could not exceed 72% accuracy, was the only category that presented mediocre performance. Additional details about this research can be found in [werghi2015b] and [werghi2016].

1.8 2D face recognition

While face recognition for frontal and moderate pose variations seems to have attained maturity, recognition of faces in extreme pose variation, exhibiting significant occlusion and missing data, which is often encountered in real-life scenarios, continues to pose a big challenge. The deficient performance of our 3D face recognition for the aforementioned Yaw pose is one such example. Elaborating a manually-crafted face representation that can effectively tackle face images acquired in uncontrolled environments, which often suffers from low quality and extreme head rotation, is not a straightforward task. With the recent advent of deep learning paradigms, we thought of approaching such a problem with a data-driven approach by designing a convolutional neural network (CNN) system that could learn a simultaneous registration and representation of facial data. The proposed CNN is composed of two interconnected modules (Fig-22)). First, a registration module that learns a set of transformation parameters in order to optimally register a facial image. Second, a representation module that learns a distinctive feature encoding of the registered face image. These two modules are connected with the output of the registration module that is being input to the representation module.

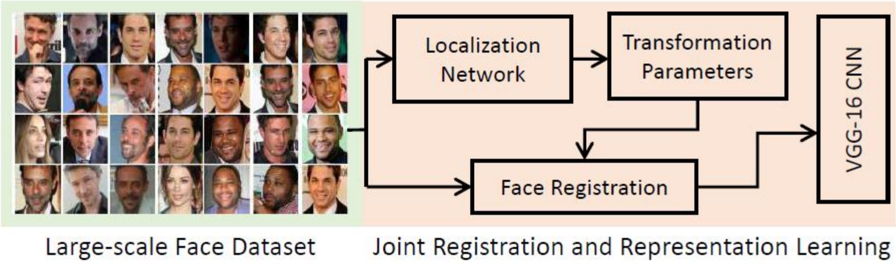
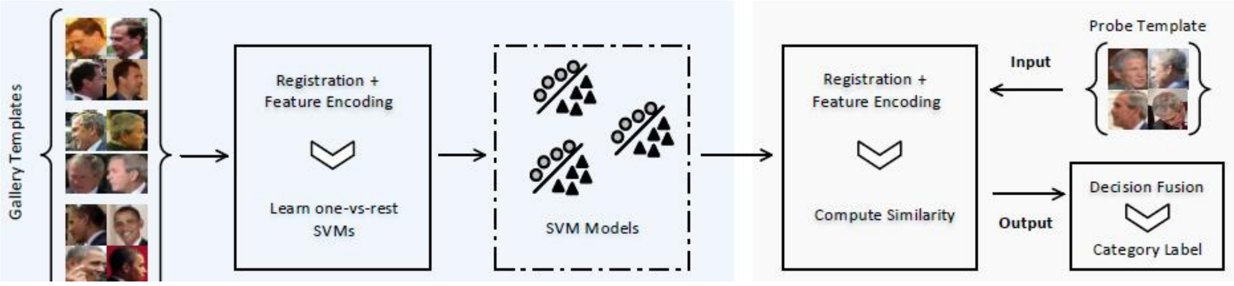

Localization Network
Transformation Parameters
VGG-16 CNN
Face Registration
Large-scale Face Dataset
Joint Registration and Representation Learning

Fig-22: Joint face registration and representation.

The registration module deploys a Spatial Transformer Network [jaderberg15]] that computes the parameters of the entire affine transformation in order to bring the face into its canonical pose. The registered face image then serves an input to the subsequent representation module. Here, we opted for the pre-defined VGG-16 architecture which has proven its superiority by getting tested on public benchmarks. The entire face encoding system comprising of the registration and the representation module was trained using the publicly available face dataset that is reported in [parkhi15].

The person recognition module was designed using a one-versus-all-rest binary support vector machine classifier, so that a discriminative SVM model was learnt for each subject. Thus, a logistic regressor was subsequently used to obtain the decision value in order to evaluate the matching of query face data to the enrolled subjects within the gallery. We also noted that the query and enrolled subject are represented by several media representations (static images, video frames); thus, there is a need to address the aspect of fusion in the classification. Here, we adopted a decision-fusion approach wherein the decision outcomes of different media are combined using the Bayesian

Classifier Combination model proposed in [kim12]. The bloc diagram of the whole system is illustrated in Fig-23.



Training phase Testing phase

Fig-23: Bloc diagram of the system depicting the subject-specific classifier within the training bloc in addition to the decision fusion in the testing bloc.

The system has been tested on three public challenging datasets, namely, the IJB-A, the YouTube Celebrities, and COX datasets. Our proposed system demonstrated neat superiority as compared to state of the art methods, whereby it achieved relative performance boosts of 3.6%, 21.6% and 12.8%, respectively. The other interesting outcome of our work is that it highlights another piece of evidence on the importance of data representation. We showed that our proposed features, which encoded the face registration and the presentation, significantly improve the performance of all other state of the art methods when they substituted their own features. Complete details about this work can be found in [hayat17].

1.9 Assessing Posterior Capsule Opacification

In the field of medical images, I have been working on the problem of assessing Posterior capsule opacification (PCO), which is a common complication arising after cataract surgery in patients who have undergone the extra capsular cataract extraction surgery. PCO is caused by the growth of lens epithelium cells (LECs) that remain within the posterior capsular area of the eye following the cataract surgery. These cells develop as different types of PCO, namely, pearls, fibrosis, and wrinkles as illustrated in Fig-24.

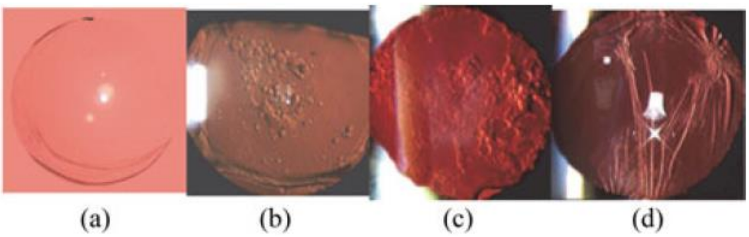
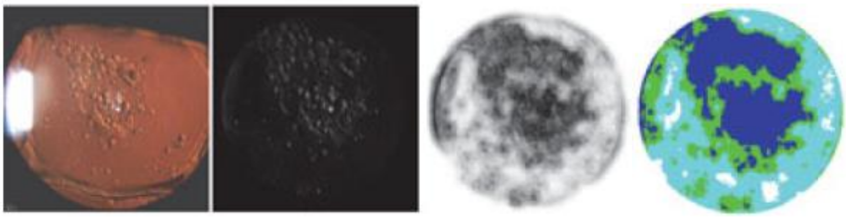


Fig-24: PCO image samples: (a) clear eye capsule, (b) pearls PCO, (c) fibrosis PCO, and (d) wrinkles PCO.

Assessing the efficacy of clinical trials performed to reduce or inhibit PCO requires both a quantitative and qualitative analysis that can accurately evaluate PCO in the eye's capsule. Human assessment is often corrupted by bias, subjectivity and inaccuracy. This tends to happen, for instance, when comparing PCO progression by studying images taken before and after the treatment, and when comparing the severity of PCO. In contrast to previous works which approached the problem by attempting to evaluate the proliferations of LEC cells using segmentation paradigms - which is quite problematic given the high level of irregularity characterizing the PCO texture - we were able to bypass the segmentation problem and proposed a novel PCO quantification based on the concept of "roughness". This concept is defined as multiscale roughness assessment undertaken at each pixel over a neighbourhood of concentric rings. The roughness assessment produces a "roughness" image, that goes into a clustering stage whereby the pixels are classified into four distinct PCO level classes, namely, clear (grade 0), mild (grade1), moderate (grade 2), and severe (grade 3). Subsequently, the PCO score is calculated based on the number of pixels falling within each cluster weighted by the severity grade of that cluster (Fig-25).



(a) (b) (c) (d)

Fig-25: Steps of proposed method: (a) Original PCO image, (b) Pre-processed image, (c) Roughness image, (d) Clustered image, where the blue, green cyan, and white colours indicate severe, moderate, mild, and clear, respectively.

Our method demonstrated a better performance when compared to the state of the art methods, with the exception of EPCO [tetz13] method, for which, our method got a slightly lower score. However, the EPCO method is manually operated, as opposed to our method which is fully automatic. More details about this research are outlined in [aruna14].

1.10 Automatic detection of polyps in colonoscopy images

In this research, we proposed a novel contribution towards the detection of polyps in colonoscopy images. Polyps are protrusions that develop at the intestinal tract (See examples in Fig-26). Their early detection and removal is crucial for a better prevention of colorectal cancer. In this context, manual clinical inspection to detect polyps is currently the preferred technique. However, this technique is plagued with many limitations, such as the dependence on the examiner's level of expertise and the equipment limitations that could lead to false or missed polyps. Here, computer-aided detection system can be a complementary tool to make a more accurate detection wherein it can support the medical expert to have a better validation whilst also helping in the follow-up process.

B.

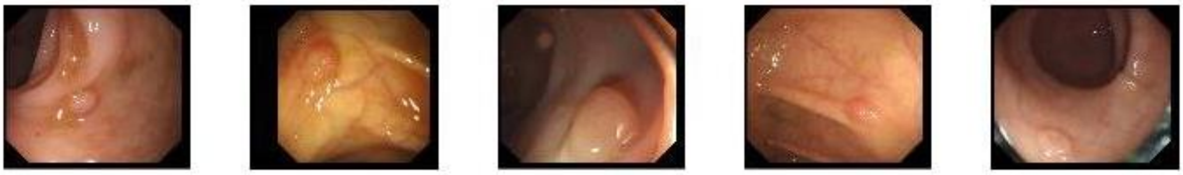


Fig-26: Polyp samples.

In this work, we proposed to leverage a CNN as a Transfer Learning (TL) scheme. The proposed approach relies on pre-trained architectures that have been trained on colossal natural images (ImageNet). In contrast to conventional methods which either perform fine-tuning or train the CNN from scratch, we utilize the output features of CNN as an input in order to train a machine learning classifier such as Support Vector Machine (SVM) and SoftMax. The polyp image samples are first divided into patches. Next, the learned weights from pre-trained nets are utilised to extract deep features from the them to be used in the decision-making process. Moreover, data augmentation approaches have also served as database enlargement techniques where adequate sample sizes for training and testing were obtained. The efficacy of the presented framework is demonstrated on a public database named CVCColonDB[3] wherein the experimental results indicate that our methodology is quite competitive with the state of the art methods; it scores the best recall score of 96% and precision score of 92.7%, which is only slightly lower than the best score (93%) recorded up to the preparation of this document. More details about this work can be found in [bilal17].

Tracking polyps is another aspect that has been investigated in this research. Conventional tracking mechanisms make use of intensity information only for the purpose of tracking. However, in this thesis, we elucidated that adding the colour contribution along with the intensity one could lead to a better tracking system. The algorithm employs three independent colour format and affine transformation for modelling the problem where steepest descent (SD) algorithm is used for optimization. Since the three selected colour signals are independent, each contribution in the algorithm could be implemented separately, such as using three cores implying that the speed of the algorithm is not affected. Results of this research have been disseminated in [bilal17a]

1.11 Detection of cervix cancer using pap-smear images

The transfer learning scheme developed for polyp detection has been tested on other medical image analysis applications, namely, single cell microscopic pap-smear images (Fig-27). The endeavour here is to design a computer-aided diagnosis tool that facilitates the early detection of cervix cancer.

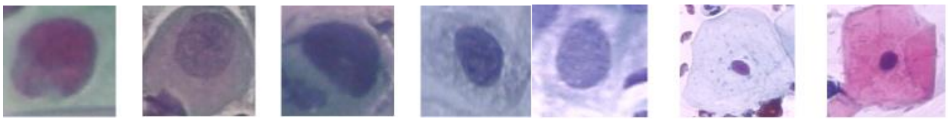


Fig-27: Samples of pap-smear images showing seven different cervix cells

Rather than adopting the conventional process of segmenting the cell into nucleus and cytoplasm and then crafting-out features to train a classifie, we considered the whole cell image as an input for a CNN network. We tested the AlexNet and the VggNet architecture on the public Herlev pap smear database[4]. A comparison with state of the art methods illuminates the superiority of our method across all metrics of two-class classification (normal- abnormal) and the best recall score for thee-class classification (normal-intermediate-abnormal). More details about this work can be found in [bilal17b].

[3] http://mv.cvc.uab.es/projects/colon-qa/cvccolondb

[4] http://mde-lab.aegean.gr/index.php/downloads

2. Current and Future Research

In this section we will expose, current ongoing research, near-future envisaged projects, and envisaged feature in the long term.

2.1 Medical imaging

Currently, we are investigating a medical imaging project that focuses on the detection of prostate cancer from Magnetic Resonance Imaging (MRI) images. Prostate cancer is one of most frequently diagnosed malignant form of cancer and the second leading cause of cancer-induced death in men after lung cancer. Early and accurate detection will enable clinicians to initiate early intervention and begin appropriate treatment in a timely manner, thereby potentially reducing the mortality rate.

We proposed an approach using diffusion-weighted magnetic resonance image (DW-MRI) acquired at different b-values following a classic 3-stage paradigm: 1) prostate area detection; 2) prostate description; and 3) classification. The prostate detection is performed using an active shape model. Then apparent diffusion coefficients (ADC) are computed for each b-value across the different scan slices. Subsequently, a cumulative distribution function (CDF) related to each b-value is constructed from these ADC maps. The so-obtained CDFs constitute a global feature that is used to distinguish between benign and malignant prostate tumours. For this purpose, auto-encoder classifiers are trained and then used for each b-value. The proposed CAD system was tested on datasets of 53 subjects, obtaining an accuracy of 100% on “leave-one-subject-out” mode for the b-700 value. The results of this primary investigation have been disseminated in [reda17].

We are currently investigating a more general framework which can bypass the problematic stage of prostate segmentation. In addition, apart from evaluating the performance of each b-value input separately, we will investigate a score-level fusion of the different classifiers associated with the b-values. Thirdly, we envisage going beyond an overall detection (making decisions as to whether the case presents malignancy scan-wise or slice-wise) towards a more refined and accurate framework with the ability to localise malign regions in each slice. Such a system will be more helpful to the clinician with regard to the selection of the biopsy location.

2.2 Convolution on the mesh manifold

This current research comes with the extension line of **shape analysis on the mesh manifold** work described in section 1.6.

The convolution operation is at the base of many computations in Mathematics, Physics, and Engineering. In particular, its discrete version has found large application in image analysis, where it is naturally used to perform image filtering, in a broad sense. For example, convolution of the image with differently formed masks allows operations that go from edge detection (Sobel mask), to smoothing (Gaussian mask), derivatives (Laplacian mask), and so on. The easiness in performing convolutions on the image domain directly derives from the grid structure of images, which allows the definition of masks and the effective implementation of the convolution with multiplications, sums and shifts.

The capability of the convolution operation of extracting meaningful patterns from an image and its effective computation are also at the base of its extensive use of CNN. Since 2012, when a CNN architecture resulted the best ranked method in the ImageNet Large Scale Visual Recognition Challenge surpassing by a large extent other competitors CNNs have become, de-facto, the standard tool to address a large spectrum of problems in the fields of Computer Vision, Pattern Recognition and

Image Processing. Starting from the above evidences, extensions and adaptations of the CNN model have been tried in other domains [bronstein17]. Among them, the mesh manifold support is of particular interest since it is largely used for modelling 3D objects either obtained synthetically or acquired with 3D scanners. As described earlier, direct application of the convolutional operation to such domain is not possible, due to the lack of the regular grid structure as it is the case in images. This determined the emergence of different solutions that redefined CNN for volumes [wu15], surfaces [sinha:2016,xie15,fang15], and point clouds [qi17].

The common trait of all these architectures is the application of the convolution operation to the input images through a series of convolutional layers using filters with different size, shift amount (stride) and padding. After, non-linearity layers usually introduced with Rectified Linear Units (ReLu). Down-sampling is also performed using some form of pooling (for example, max pooling basically takes a filter and a stride of the same length, and outputs the maximum number in every sub-region that the filter convolves around).

Replicating convolution and pooling on the mesh will be at the base of any CNNs extension to such domain. We envisage developing a new framework that enables computing convolutions on the mesh support. This will open the way to a wide spectrum of filtering operations in this domain. Here, we will capitalize on the Ordered Ring Facets representation (ORF) [werghi12] that, given a facet on the mesh, allowed us to provide a local ordering of its neighbours. With this ordering, and a local reference frame, extension of the convolution becomes possible by emulating the 2D-like shift operation. The possibility to inherit the order from the ORF allows to define a shift operator on the mesh surface. While on the image the neighbours of a pixel are given by the Cartesian coordinates derived by the grid structure of the image itself, with the ORF we use polar coordinates, i.e., radius r and quantized angle θ. Therefore, the convolution between a given mesh M and a filter F is defined as follow:

$$M * F = \sum_{r} \sum_{\theta} m_{r,\theta}.\ f_{r,\theta}$$

Where $m_{r,\theta}$ and $f_{r,\theta}$ are, respectively, a scalar function computed on the mesh and the filter values, both at radius *r* and angle θ. In images, the convolution is performed at each pixel: neighbour pixels are multiplied by the filter values; in our proposed approach, instead, the convolution is performed on the facets, therefore filter values have to be determined at each facet of the ring. Here, we need to address two problems 1) defining a filter function on the mesh, and 2) browsing the filter across the mesh manifold.

For the first problem, a first solution would be to define the filter over a polar coordinate support then mapping it to the ORF structure which benefits from a polar coordinate-like structure, whereby the ring number and the index of the facet in that ring cab be identified to the radius *r* and to the angle θ, respectively. Fig.29 depicts examples of Gabor filter instance mapped on the mesh. Yet there are other issues related to the mesh resolution and the support scale that still need to be investigated.

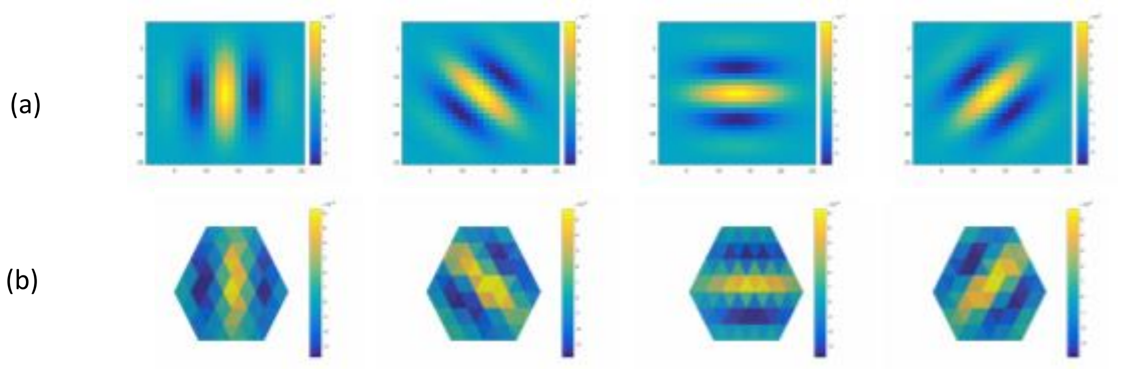


Fig-29: Gabor filter images. (b): corresponding Gabor filters on the mesh mapped on a 3-ring ORF.

Regarding browsing the filter across the mesh manifold, we plan also to capitalize on the ordered structure of the ORF. The ordering of the facet across the rings, allows to slide the filter in a spiral-wise fashion, across the mesh manifold, at a different strand values. Fig-30 depicts an example illustrating this concept.

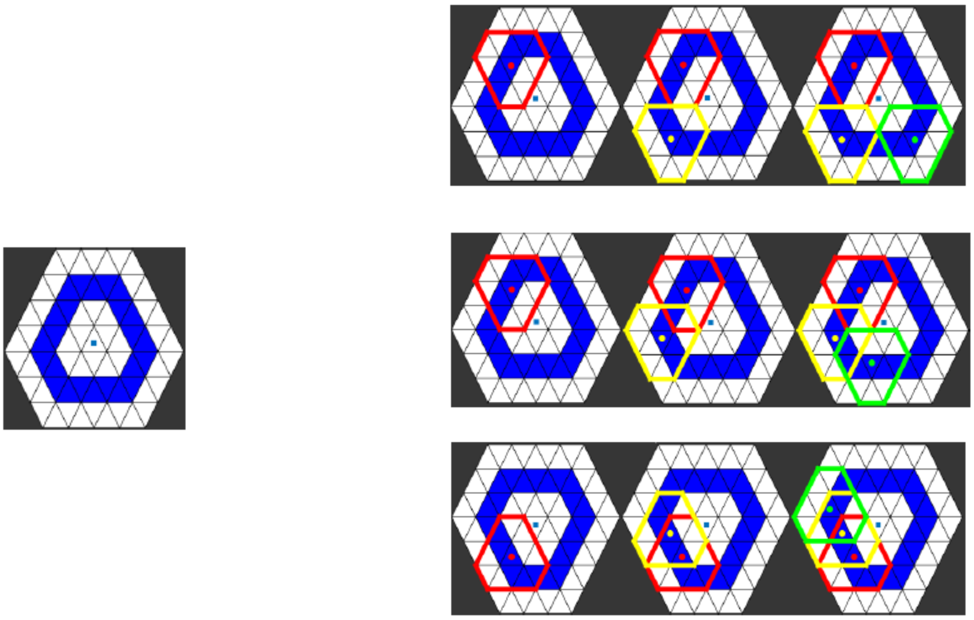


Fig-30: Left: Example of an ORF region of a 3-ring size. Right: sliding of a 1-ring filter support at decreasing strands of 6, 4, and 1, respectively.

2.3 Subject retrieval based on eyewitness's visual description.

In criminology and police investigation, facial sketches (called also facial composite) are commonly used in searching and identifying suspects in crimes, in the absence of the suspect(s) photos [jain12]. In addition to identification, facial composite can be used as additional evidence, to assist investigation at checking leads, and to defuse warning of vulnerable population against serial offenders. Currently, the identification procedure uses legal sketches and composite sketches. Legal sketches are sketches drawn by forensic artists referring to the description provided by a witness. Composite sketches are sketches of faces rather built using software allowing an operator to select and combine different elements of the face.

The current procedure of suspect identification based on witness description as currently adopted by authorities does not yet seem to profit from all the available resources. In particular the face database maintained by legal authorities and which are continuously fed from network of cameras deployed at access control points and public places. Performance-wise, the current procedures suffer from several shortcomings. Legal sketches production is subjective and depends on the artist skills. Facial composite software, while offer comprehensive construction functionalities, they often produce a mismatched outcome. Moreover, both categories use 2D face reconstruction, which does not accurately reflect the actual 3D shape features of the subject. Recently some methods proposed to match the face sketch to mugshots (photos of person taken after being arrested) [klare11, klare13] and composite sketches to mugshots [yuen07, han13]. In both of these two schemes, witness description goes through a human interpretation stage, namely the expert artist for the face sketch, and the software operator for the composite sketch. Both face sketch and composite sketch are therefore subjected to reconstruction error as illustrated in the examples shown in Fig-31

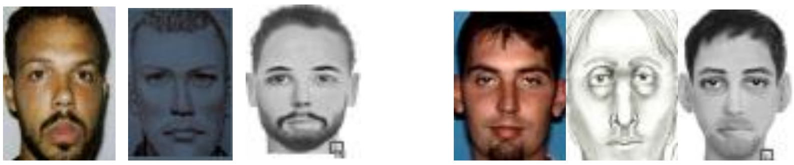

[5]

Fig-31 Two examples showing real face, a face sketch and composite sketch. We can easily notice the difference between these two types of reconstruction and the real instances.

We plan to develop a new framework, whereby rather than adopting a face reconstruction, we propose a face retrieval approach where the input is set of textual description compiling the group of facial feature and traits provided by an eye-witness. These will be used to interrogate a face database and retrieve a set of potential suspects. The system pipeline is as shown in Fig-32.

The rationale behind this novel approach is that performing a direct match between the verbal description and the face database eliminates the reconstruction error produced in the face sketch and the composite sketch. In addition, by considering a 3D face image database, the approach has higher capacity in retrieving facial trait and features that are not preserved in 2D images because of the loss of geometry by projection.

Achieving such system requires addressing several challenges. First, the representation of witness verbal description into appropriate numerical descriptors and defining the mechanisms for in such a conversion. Second, how to perform the automatic annotation of a large face database? That is, given a specific subject, how we can automatically map the 3D face data to the different facial attributes? Third, how to perform the comparison between the query descriptors and the database counterparts and what optimal metric can be adopted for this purpose?

[5] Courtesy S.Klum et al , ''Sketch Based Face Recognition: Forensic vs. Composite Sketches'' Int. Conf. Biometrics 2013

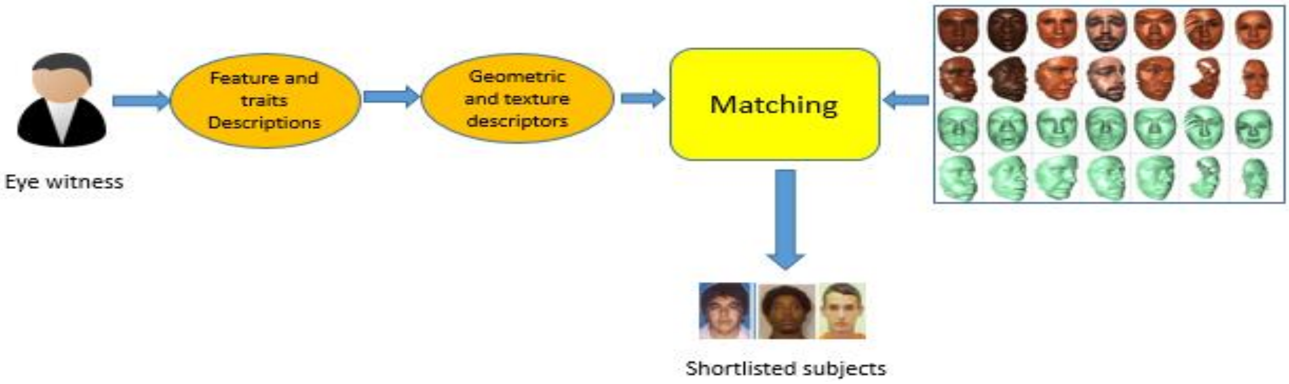

Eye witness
Feature and traits Descriptions
Geometric and texture descriptors
Matching
Shortlisted subjects

Fig-32: Suspect retrieval process pipeline.

2.4 Middle and long-term research projects.

For the next stage, I plan to carry on with a multi-disciplinary research spanning 2D/3D data analysis and interpretation as well as targeting innovative human-machine interface models and applications. Considering the recent advancements in 3D video acquisition and their continuous cost reduction, this technology can potentially be deployed on several machine-people interaction scenarios in a cost-effective manner. Potential areas of application include identity management, medical rehabilitation, entertainment, and elderly care, among many others. With the rapid proliferation of smart phones, smart multi-modal mobile applications integrating and customizing the aforementioned interface models is another promising area of research and development. Such an application would be of great interest for people residing in remote areas as well as for emergency consultations.

Retrieval of content-based multimedia information is expected to be a topic that generates great interest in the near future. In effect, with the widespread availability of 3D digitizers and the exponentially growing field of multimedia technology, large collections of hybrid multimedia models can be readily built and connected to the internet for different applications and in different sectors. Developing properly customized structures and mechanisms for the purpose of interrogating as well as retrieving information from such heterogeneous databases will present a worthwhile challenge for researchers.

3. Conclusion

Coming to the conclusion of this report, I hope I have shown how central is the problem of representation for a wide spectrum of problems and applications in computer vision. I introduced at the beginning this concept and reported the motivation and the rational that guided me during my research activities, hoping that it would help the reader grasp the logic and the nature of my research contributions.

While I did not elaborate much on the reasons explaining the diversity of the contexts and the applications that characterize my research, I presume that one can easily link it to the different academic environments I have been in throughout my academic career as well as the adaptations made in response to funding opportunity requirements.

Although I tried to position the representation at the heart of my contributions, one cannot go without having the impression that, for the most recent contributions, the representation concept tended to be shadowed by the machine learning paradigms. I believe, however, that this trend actually reflects a shift ***to some extent*** from the expert-designed representation to the *engineered representation* or *learned representation*. This shift, from my perspective, did not come about by choice, but rather as a constraint for the sake of addressing the related challenges and pushing the frontiers of the state of the art, noticeably with regard to the performance. This shift will remain bounded, as highlighted in the aforementioned statement, by many factors which, in my opinion, will keep data representation, and the manual feature design quite relevant, and *a fortiori* indispensable. Indeed, the design of deep learning architecture, is a manual process, where some components represent hard coded feature extraction (e.g. pooling layer) set by the architecture designer and are not meant to be learned by training. Second, looking at the different applications and their related data, for example in our current research projects, described in sections 2.1 and 2.2, it is evident that a deep learning treatment of this data in its raw format is not feasible, and that working out an appropriate presentation of the data is actually needed. The data representation becomes imperative when dealing with multi-modality problems requiring a proper fusion and aggregation framework.

References

[werghi99] N. Werghi, R. Fisher, C. Robertson and A. Ashbrook, “Object reconstruction by incorporating geometric constraints in reverse engineering”, *Computer-Aided Design,* vol.31, no.6, pp.363-399, 1999.

[ashbrook98] A. P. Ashbrook, R. B. Fisher, C. Robertson, N. Werghi, “Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms”, *Proc. European Conference on Computer Vision*, 1998, Friburg, Germany, June, pp. 674-686.

[berretti13] S. Berretti, N. Werghi, A.D. Bimbo and P. Pala, “Matching 3D Face Scans using Interest Points and Local Histogram Descriptors”, *Computers & Graphics*, vol.37, no.5, pp.509-525, 2013.

[werghi06] N. Werghi, Y. Xiao and P. Siebert , “A Functional-Based Segmentation of Human Body Scans in Arbitrary Postures,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 36, No.1, pp. 153-165, 2006.

[werghi2006a] N. Werghi, “A Robust Approach for Constructing a Graph Representation of Articulated and Tubular-like Objects from 3D Scattered Data”, *Pattern Recognition Letters*, vol. 27, no.6, pp. 643-651, 2006.

[shen99] D. Shen and H. IP, “Discriminative wavelet shape descriptors for recognition of 2-D patterns”, *Pattern Recognition*, vol.32, no.2, pp.151-165, 1999.

[werghi02b] N. Werghi, “Recognition of Human Body Posture from a Cloud of 3D Data Points Using Wavelet Transform Coefficients”, *Proc. International Conference on Automatic Face and Gesture Recognition Washington*, 2002, USA, May.

[werghi05] N. Werghi, “A Discriminative 3D Wavelet-based Descriptors: Application to the Recognition of Human Body Postures”, *Pattern recognition letters*, vol.26, no.5, pp. 663-677, 2005.

[werghi11] N.Werghi, “Assessing the Regularity of 3D Triangular Mesh Tessellation Using a Topological Structured Pattern”, *Computer-Aided Design and Applications*, vol. 8, no.5, pp. 633-648, 2011.

[werghi11b] N. Werghi, H.Bhaskar, Y.Meguebli and H.Boukadia, “The Spiral Facets: A Unified Framework for the Analysis and Description of 3D Facial Mesh Surfaces”, *Proc. Int. Conference Computer Vision Theory and Applications (VISAPP),* 2011, Algrave, Portugal, pp. 30-39. **BEST PAPER AWARD**.

[werghi12] N. Werghi, M. Rahayem and J. Kjellander. “An ordered topological representation of 3D triangular mesh facial surface: Concept and applications”, *EURASIP Journal on Advances in Signal Processing*, pp.144, 2012.

[ojala02] T. Ojala, M. Pietikäinen and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[werghi2015] N. Werghi, C. Tortorici, S. Berretti, A. Del Bimbo, “Representing 3D Texture on Mesh Manifolds for Retrieval and Recognition Applications”, *IEEE Conf. Computer Vision and Pattern Recognition*, CVPR2015, Boston, USA, 2015, pp. 2521-2530.

[werghi2015a] N. Werghi, S. Berretti and A. Del Bimbo, “The mesh-LBP: a Framework for Extracting Local Binary Patterns from Discrete Manifolds”, *IEEE transactions on image processing*, vol. 24, no. 1, pp.220-235, 2015.

[tortorici2017] C. Tortorici, N. Werghi, S. Berretti, “Defining Mesh-LBP Variants for 3D Relief Patterns Classification”, *7th International Workshop on Representation, analysis and recognition of shape and motion from Image data*, 2017, Savoi, France. **BEST STUDENT PAPER AWARD**.

[biasotti17] S. Biasotti, "Retrieval of Surfaces with Similar Relief Patterns", *European Association for Computer Graphics*, 2017. [Online]. Available: https://diglib.eg.org/handle/10.2312/3dor20171058.

[ahonen2006] T. Ahonen, A. Hadid and M. Pietik¨ainen, “Face description with local binary patterns: Application to face recognition”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.28, no.12, pp.2037–2041, 2006.

[werghi2015b] N. Werghi, C. Tortorici, S. Berretti and A. Del Bimbo, “Local Binary Patterns on Triangular Meshes: Concept and Applications”, *Computer Vision and Image Understanding*, 2015.

[werghi2016] N. Werghi, C. Tortorici, S. Berretti and A. Del Bimbo, ‘’ Boosting 3D LBP-based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh’’, *IEEE Transactions on Information Forensics & Security*, pp. 964 – 979, 2016.

[jaderberg15] M. Jaderberg, K. Simonyan, A. Zisserman et al., “Spatial transformer networks”, in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[parkhi15] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition”, in *British Machine Vision Confernce*, vol. 1, 2015, p. 6.

[kim12] H.C. Kim and Z. Ghahramani. “Bayesian classifier combination”, in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 619–627.

[hayat17] M. Hayat, S. H. Khan, N.Werghi and R. Goecke, “Joint Registration and Representation Learning for Unconstrained Face Identification”, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2767-2776

[tetz13] M. R. Tetz, “EPCO2000: Software for the evaluation of posterior capsule opacification”, 2013. [Online]. Available: www.epco2000.de.

[aruna14] A. Vivekanand, N. Werghi and H. Al-Ahmad, “Multi-scale Roughness Approach for Assessing Posterior Capsule Opacification”, *IEEE Journal of Biomedical and Health Informatics*, vol.18, no.6, pp. 1923 – 1931, 2014.

[bilal17] Taha, J. Dias and N.Werghi, “Convolutional Neural Network as a Feature Extractor for Automatic Polyp Detection’’, *IEEE Int. Conf. Image Processing*, Beijing, China, 2017.

[bilal17a] B. Taha, C.Doignon, N.Werghi and J.Dias, “Fast polyps tracking with color image regions alignment”, *Surgetica* 2017, Strasbourg France.

[bilal17b] B. Taha, J. Dias and N. Werghi, “'Screening cervical-cancer using pap-smear images: A convolutional neural network Approach”, *Medical Image Understanding and Analysis (MIUA)* 2017, Edinburgh, UK, 2017.

[reda17] I. Reda, A. Shalaby, M. Elmogy, A. Abou Elfotouh, F. Khalifa , M. Abou El-Ghar, E. Hosseini-Asl, G. Gimel'farb, N. Wergi and A. El-Baz, "A Comprehensive Non-invasive Framework for Diagnosing Prostate Cancer," *Computers in Biology and Medicine*, vol. 81, pp. 148-158, 2017.

Appendix

Representative papers

ELSEVIER

Computer-Aided Design 31 (1999) 363–399

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Object reconstruction by incorporating geometric constraints in reverse engineering

N. Werghi*, R. Fisher, C. Robertson, A. Ashbrook

Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, UK

Received 25 April 1998; received in revised form 3 December 1998; accepted 16 April 1999

Abstract

This paper deals with the constrained reconstruction of 3D geometric models of objects from range data. It describes a new technique of global shape improvement based upon feature positions and geometric constraints. It suggests a general incremental framework whereby constraints can be added and integrated in the model reconstruction process, resulting in an optimal trade-off between minimization of the shape fitting error and the constraint tolerances. After defining sets of constraints for planar and special case quadric surface classes based on feature coincidence, position and shape, the paper shows through application on synthetic model that our scheme is well behaved. The approach is then validated through experiments on different real parts. This work is the first to give such a large framework for the integration of geometric relationships in object modelling. The technique is expected to have a great impact in reverse engineering applications and manufactured object modelling where the majority of parts are designed with intended feature relationships. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Reverse engineering; Geometric constraints; Constrained shape reconstruction; Shape optimization

1. Introduction and related work

The use of constraints in object modelling is an important topic in the CAD literature. In this area, engineering concepts and shape constraints are transformed into shape models through mechanisms of checking, incorporating and solving constraints in the modelling process. Constraints in this area include specification of the geometric relationships between object features as well as engineering constraints (dimensions, material strength and machining parameters) [1,2].

Finding geometric configurations that satisfy the constraints is the crucial issue and much research has been dedicated to different mechanisms for constraint solving. There are two main strategies for solving constraint problems according to the classification mentioned in Ref. [3]. The first strategy, referred as the instance solver, uses specific values of the constraints and looks for geometric configurations satisfying these constraints. In the second strategy, the generic solver investigates first whether the geometric elements could be placed given the constraints independently of their values. After checking that the problem is well-constrained, the specific placements of the geometric elements are then determined. In CAD literature, these two strategies have been implemented through different approaches.

The numerical approaches given in Refs. [4–7] are typical instance solvers. Constraints are translated into a set of algebraic equations and are usually solved simultaneously by means of iterative techniques, for instance the Newton–Raphson algorithm. This approach can deal with general cases, over-constrained systems and inconsistent constraint problems. A good initial value is required for such solvers and the algorithm should be applied with care since it may face an ill-conditioned problem.

Symbolic methods [8–11] are hybrid methods in the sense that they can involve both the generic solver strategy and instance solver strategy. These methods also transform the geometric constraints into algebraic equations but instead of numerical techniques, general symbolic methods are first used to put the set of equations into a new form which is easy to solve. The set of equations is sequentially reduced by solving the simplest one at each step as far as possible. The final set can be then solved numerically. Compared to numerical approaches, they are not subject to numerical instabilities and can locate all solutions to

* Corresponding author. Tel.: + 44-131-650-4504; fax: + 44-131-650-68999.

E-mail address: naoufelw@dai.ed.ac.uk (N. Werghi)

0010-4485/99/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved.
PII: S0010-4485(99)00038-X

the constraint equations. However, they tend to be computationally expensive. This often restricts the types of geometric elements and types of constraints allowed to be involved.

A more recent approach solves the constraints through sequential geometric constructions, as most configurations in engineering drawing are solvable by ruler, compass and protractor. These approaches can be roughly divided into two categories: the rule-based [12–14] and graph-based [15–20] approaches. In the first category constraints are expressed by rules or predicates. The procedure starts from an initial set of predicates defining the constraints and sequentially derives a new set of predicates by applying logical reasoning techniques, with the predicates converging towards defined positions for all the characteristic features. However since only constructive geometries can be handled by these methods they may not be very efficient for large systems of constraints.

The graph-based approaches handle the problem in a more methodical way. They start by forming a graph representation of the problem. In this graph each node represents a geometric element and the edges linking these nodes indicate the constraints between the associated geometric elements. Each edge is labelled with the constraint's type. In a first phase the graph is analysed and if it is well-constrained a set of sequential construction steps are derived from it. This phase depends only on the type and the number of constraints, so it is considered a generic constraint solver. In the second phase the construction steps are carried out integrating the actual values of the constraints to derive the solution shape.

In the Computer Vision community, constraints are mainly used in model-based recognition and localization of objects or environments more generally. They are used as a priori information to reduce the search space between, for example, the model features (already stored and known CAD models) and the extracted features from visual sensor output (grey level image, 3D range data, etc.) [21–26]. Some of the approaches for object recognition in particular [27,28] use a notion of graph representation close to the one used in the graph-based approaches for constraint solving, where the nodes represent object primitives (e.g. points, lines, etc.) and the arcs present geometric relationships between them (e.g. adjacency, parallelism, perpendicularity, etc.).

Constraints can be defined over the geometric and topological relationships between the object model features (the a priori information) and the extracted features from the input data. These relationships are derived either from the properties of the geometric transformation between the vision sensor frame and the scene frame or the transformation between two vision sensor frames (stereo-vision) or the intrinsic structure of the objects [29].

So we can conclude that when computer vision applications deal with model-based recognition and localization, the definition and the concept of constraints are wider than those considered in CAD applications, although they may share the same terminology.

There is one area where Computer Aided Design and Computer Vision share a similar interpretation of geometric constraints, namely reverse engineering referred to as 3D geometric model reconstruction within the vision community. Reverse engineering is typically concerned with parts and industrial objects, whereas 3D geometric model reconstruction is a larger field which includes built environments. But the two terms point to the same goal, which is the transformation of a real object (in the large sense of the word) to a model and concept. In parts manufacturing reverse engineering deals with measuring an existing object so that a surface or solid model can be deduced in order to take advantage of CAD/CAM technologies. It is also often necessary to produce a copy of a part when no original drawings or documentation are available. In other cases we may want to re-engineer an existing part, when analysis and modifications are required to construct a new improved product. Even though it is possible to turn to a computer-aided design to fashion a new part, it is only after the real object is made and evaluated that we can see if the object fits with real world. For this reason designers rely on real 3D objects (real scale wood or clay models) as starting points. This procedure is particularly important to areas involving aesthetic design e.g. automobile industry or generation of custom fits to human surfaces such as helmets, space suits or prostheses.

A review of the main research in the CAD community [30–33] and the Vision community [34–36] (for reconstruction from single range images) and [37–40] (for reconstruction from multiple range images) revealed that the exploitation of geometric constraints has not been fully investigated. This lack was noted in the survey work of Varady et al. [41].

The first motivation behind considering geometric constraints in this work is that models needed by industry are generally designed with intended feature relationships so this aspect should be exploited rather than ignored. The consideration of these relationships is actually necessary because some attributes of the object would have no sense if the object modelling scheme did not take into account these constraints. For example, take the case when we want to estimate the distance between two parallel planes: if the plane fitting results gave two planes which are not parallel, then the distance measured between them would have no significance. Furthermore exploiting the available known relationships would be useful for reducing the effects of registration errors and mis-calibration, thus improving the accuracy of the estimated part features' parameters and consequently the quality of the modelling.

The second motivation is that generally in the manufacturing process, once the part is produced many improvements are carried manually to optimize the part and make it fit with the real world (e.g. fit with another part, adjust the part to fit particular customer). These improvements could be represented by new constraints on the shape of the part. By integrating these constraints into the CAD process the work piece optimization would be reduced and hence many cycles in the part production process would be saved. In other cases, such improvement could not be achieved by hand due to the complexity of the object or when we want to extend the application of the process to complex environments such as buildings or industrial plants.

From a CAD viewpoint the way with which the constraint problem is handled is close to the numerical constraint solver. However it differs radically from this scope on two levels. First on the level of the components of the problem. In our case we have already a real object whose shape we are trying to reconstruct, hence the object real data is used to constraint the shape. Thus, the solution has to satisfy proximity to measured points as well as the constraints. Second the numerical technique used to find the solution overcomes ill-conditioning problems.

The approach for incorporating geometric relationships in object modelling has to tackle two problems. The first is how to represent the constraints. The second is how to integrate these constraints into the shape fitting process. These two aspects are not entirely independent, the shape fitting technique imposes restrictions on the constraint representation and vice versa.

A first step in the direction of incorporating constraints for assuring the consistency of the reconstruction was done by Porrill [42]. He linearized a set of nonlinear constraints and combined them with a Kalman filter, as applied to wire frame model construction. Porrill’s method takes advantage of the recursive linear estimation of the Kalman filter, but guarantees satisfaction of the constraints only to linearized first order. Additional iterations are needed at each step if more accuracy is required. This last condition has been taken into account in the work of De Geeter et al. [43] by defining a “Smoothly Constrained Kalman Filter”. The key idea of their approach is to replace a nonlinear constraint by a set of linear constraints applied iteratively and updated by new measurements in order to reduce the linearization error. However, the characteristics of Kalman filtering make these methods essentially adapted for iteratively acquired data and many data samples. Moreover, there was no mechanism for determining how successfully the constraints were satisfied and only lines and planes were considered in both of the above works.

The constraints considered by Bolle et al. [44] in their approach to 3D object position covered only the shape of the surfaces. They chose a specific representation for the treated features: plane, cylinder and sphere.

Compared to Porrill’s and De Geeter’s work, our approach avoids the drawbacks of linearization, since the constraints are completely implemented. Moreover, our approach covers a larger category of feature shapes. Regarding the work of Bolle [44], the type of constraints which can be held by our approach go beyond the restricted set of surface shapes and cover also the geometric relationships between object features. The proposed approach has been successfully applied first on polyhedral objects [45]. To our knowledge the work appears the first to give such a large framework for the integration of geometric relationships for object reconstruction in the field of reverse engineering.

Although this work is mainly intended for object modelling, it can also find many other many useful applications, e.g. in object localisation. In registration tasks, the features represented in different views need to be put into a single reference frame. For this purpose the transformation between different views is recovered by matching between the related frames. Since a reference frame is built from object features, e.g. normals of surfaces which are supposed to be orthogonal, the estimation of the surfaces has to satisfy the orthogonality constraints. The proposed paradigm may be extended as well to any constrained built environment application like creating “as built” CAD models of an industrial plant for planning new building work. A current method uses a motorised camera head to create highly detailed panoramic images which are then used to extract CAD models. Since the different captured parts of a plant (pipes, reservoirs, etc.) have many geometric relationships between them, using these constraints in the reconstruction process will help to have a consistent whole model. The same is true as well for modelling different compartments of buildings or cities. The current methods of extracting, matching and estimation of large scale buildings’ features from aerial images have reached reasonable level. This make the application of our method for modelling different compartments of buildings or cities possible as well.

The organisation of the rest of paper will be as follows: the next section gives some preliminaries on planes and quadric surfaces and gives the parameterization of such surfaces. The aim is to make clear the relationship between the constraint formulation and the surface representations. We then state the problem and develop the proposed approach. Next we define and classify the different types of constraints. Lastly, we demonstrate the process on several synthetic and real objects to evaluate the accuracy, the convergence, repeatability and consistency of the approach.

2. Preliminaries

This section gives a brief overview about constraining planes, general quadrics and some particular quadric shapes. A full treatment of these surfaces can be found in Ref. [46]. While the material contained here is largely elementary geometry, we present it in order to make clear how the set of constraints used for each surface type and relationship relate to the parameters of the generic quadric.

2.1. The line

A line is defined by the following equations:

$$\frac{x - x_0}{l} = \frac{y - y_0}{m} = \frac{z - z_0}{n} \tag{1}$$

where $\vec{X}_0 = [x_0, y_0, z_0]^{\mathrm{T}}$ is an arbitrary point of the line and the vector $\vec{p} = [l, m, n]^{\mathrm{T}}$ defines the orientation of the line.

2.2. The plane

A plane surface can be represented by this equation:

$$f(x,y,z) = n_x x + n_y y + n_z z + d = 0 \tag{2}$$

where $\vec{n} = [n_x, n_y, n_z]^{\mathrm{T}}$ is the unit normal vector to the plane and d is the distance to the origin. A plane can have two different representations $(\vec{n}, d)$ and $(-\vec{n}, -d)$. This ambiguity is easily removed by orienting the normal toward the outside of the object.

Given N data points the best parameters which satisfy (2) in the least squares sense are those minimizing the criterion:

$$\sum_{i=1}^{N} f(x_i, y_i, z_i)^2 = \vec{p}^{\mathrm{T}} H \vec{p} \tag{3}$$

where $\vec{p} = [n_x, n_y, n_z, d]^{\mathrm{T}}$ is the parameter vector and H is the data matrix defined by

$$H = \sum_{i=1}^{N} \vec{h}_i \vec{h}_i^{\mathrm{T}}, \qquad \vec{h}_i = [x_i, y_i, z_i, 1]^{\mathrm{T}} \tag{4}$$

H is symmetric and positive definite.

2.3. The quadrics

A general quadric surface is represented by the following quadratic equation:

$$f(x,y,z) = ax^2 + by^2 + cz^2 + 2hxy + 2gxz + 2fyz + 2ux + 2vy + 2wz + d = 0 \tag{5}$$

which can be written:

$$X^{\mathrm{T}} A X + 2X^{\mathrm{T}} B + C = 0 \tag{6}$$

where

$$A = \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix}, \qquad B = [u, v, w]^{\mathrm{T}}, \qquad C = d; \qquad X = [x, y, z]^{\mathrm{T}}. \tag{7}$$

The type of the quadric depends on the discriminant of the quadric Δ, the cubic discriminant $\mathscr{D}$:

$$\Delta = \begin{vmatrix} a & h & g & u \\ h & b & f & v \\ g & f & c & w \\ u & v & w & d \end{vmatrix} \qquad \mathscr{D} = \begin{vmatrix} a & h & g \\ h & b & f \\ g & f & c \end{vmatrix} \tag{8}$$

and the cofactors of $\mathscr{D}$:

$$\mathscr{A} = bc - f^2, \quad \mathscr{B} = ac - g^2 \quad \mathscr{C} = ab - h^2$$
$$\mathscr{F} = gh - af, \quad \mathscr{G} = hf - bg, \quad \mathscr{H} = gf - ch. \tag{9}$$

Similarly to the plane case, the best parameters which satisfy (5) for N data points in the least squares sense are those minimizing the criterion:

$$\sum_{i=1}^{N} f(x_i, y_i, z_i)^2 = \vec{p}^{\mathrm{T}} \left(\sum_{i=1}^{N} \vec{h}_i \vec{h}_i^{\mathrm{T}} \right) \vec{p} = \vec{p}^{\mathrm{T}} H \vec{p} \tag{10}$$

where $\vec{p} = [a, b, c, h, g, f, u, v, w, d]^{\mathrm{T}}$ and $h_i^{\mathrm{T}} = [x_i^2,\ y_i^2,\ z_i^2,\ 2x_i y_i,\ 2x_i z_i,\ 2y_i z_i,\ 2x_i,\ 2y_i,\ 2z_i,\ 1]$.

2.4. The cylinder

The quadric is a cylinder when $\Delta = \mathscr{D} = 0, u\mathscr{A} + v\mathscr{H} + w\mathscr{G} = 0$ and $\mathscr{A} + \mathscr{B} + \mathscr{C} > 0$. The equation of the cylinder axis is

$$\frac{x - \dfrac{uf}{\mathscr{F}}}{1/\mathscr{F}} = \frac{y - \dfrac{vg}{\mathscr{G}}}{1/\mathscr{G}} = \frac{z - \dfrac{wh}{\mathscr{H}}}{1/\mathscr{H}}. \tag{11}$$

This means that the cylinder axis has the direction vector $[1/\mathscr{F}, 1/\mathscr{G}, 1/\mathscr{H}]^{\mathrm{T}}$ and passes through the point $\vec{X}_0 = [(uf/\mathscr{F}), (vg/\mathscr{G}), (wh/\mathscr{H})]^{\mathrm{T}}$. The axis orientation corresponds to the eigenvector related to the null eigenvalue of the matrix A. The two other eigenvalues are positive.

2.4.1. The circular cylinder

For a circular cylinder, we can show that the parameters of the quadric should also satisfy the following conditions:

$$agh + f(g^2 + h^2) = 0 \qquad cfg + h(f^2 + g^2) = 0$$
$$bhf + g(h^2 + f^2) = 0 \qquad \frac{u}{f} + \frac{v}{g} + \frac{w}{h} = 0. \tag{12}$$

The matrix A (see (7)) has two identical eigenvalues λ and the radius can be expressed by

$$r^2 = (u^2 f/\mathscr{F} + v^2 g/\mathscr{G} + w^2 h/\mathscr{H} + d)/\lambda. \tag{13}$$

A circular cylinder may be also represented by the canonical form:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - (n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0))^2 - r^2 = 0 \tag{14}$$

where $\vec{X}_0 = [x_0, y_0, z_0]^{\mathrm{T}}$ is an arbitrary point on the axis, $\vec{n} = [n_x, n_z, n_y]^{\mathrm{T}}$ is a unit vector along the axis and r is the radius of the cylinder.

This form has the advantage of having a minimal number of parameters. However its implementation in the optimization algorithm may cause some complexity, indeed it is not possible with this form to get separate terms for the data and the parameters as in (10) (which allows the data terms to be

Table 1
Relationships between features

	Point	Line	Plane	Quadric surface
Point	Coincident separation	Inclusion separation	Inclusion separation	Inclusion separation
Line	–	Coincident relative orientation separation	Inclusion relative orientation separation	Inclusion relative/orientation separation
Plane	–	–	Coincident relative orientation separation	Relative orientation/separation
Quadric surface	–	–	–	Coincident relative/orientation separation

computed off line). Consequently this may increase the computational cost dramatically.

The expansion of (14) and the identification with (5) yields

$$a = 1 - n_x^2 \qquad b = 1 - n_y^2 \qquad c = 1 - n_z^2 \qquad h = -n_x n_y$$

$$g = -n_x n_z \qquad f = -n_y n_z. \tag{15}$$

2.5. The cone

A cone surface satisfies $\Delta \neq 0$, $\mathscr{D} = 0$. The apex of the cone is given by:

$$\vec{X}_0 = A^{-1}B. \tag{16}$$

The axis of the cone corresponds to the eigenvector related to the negative eigenvalue of the matrix A. The two other eigenvalues are positive.

2.5.1. Circular cone

For a circular cone the parameters of the quadric equation have to satisfy the following conditions:

$$\frac{af - gh}{f} = \frac{bg - hf}{g} = \frac{ch - fg}{h}. \tag{17}$$

As for the cylinder case, a circular cone equation has a more compact form:

$$[(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2]\cos^2(\alpha) - [n_x(x - x_0) + n_y(y - y_0) + n_z(z - z_0)]^2 = 0 \tag{18}$$

where $[x_0, y_0, z_0]^{\mathrm{T}}$ is the apex of the cone, $[n_x, n_y, n_z]^{\mathrm{T}}$ is the unit vector defining the orientation of the cone axis and α is the semi-vertical angle. The quadric equation parameters can thus be expressed explicitly as a function of the above terms by:

$$a = n_x^2 - \cos^2\alpha \qquad b = n_y^2 - \cos^2\alpha \qquad c = n_z^2 - \cos^2\alpha$$

$$h = n_x n_y \qquad g = n_x n_z \qquad f = n_y n_z. \tag{19}$$

For the same reasons as mentioned in the cylinder case, the compact form of the cone equation is not adequate for the optimization algorithm. Nevertheless it is useful to implicitly impose the conic circularity constraints.

2.6. The sphere

A sphere is characterized by equal coefficients for x^2, y^2 and z^2 terms and vanishing coefficients for the cross product terms xy, xz and yz, so the parameters h, g and f are all equal to zero. The equation of a sphere can be written as:

$$a(x^2 + y^2 + z^2) + 2ux + 2vy + 2wz + d = 0. \tag{20}$$

The centre of the sphere is:

$$\vec{X}_0 = [-u/a, -v/a, -w/a]^{\mathrm{T}} \tag{21}$$

and the radius is:

$$r^2 = \frac{u^2 + v^2 + w^2 - ad}{a^2}. \tag{22}$$

3. The geometric constraints

The set of constraints associated with a given object can be divided mainly into two categories. The first one is the surface intrinsic constraints covering the geometric properties which reflect the specific shapes of the surfaces. Examples of these constraints will be given in the next subsection. The second category named the feature extrinsic constraints, defines the geometric and topological relationships between the different object features.

3.1. Specific shape constraints

In the text below, when we say that an equation (or set of equations) can be used as a constraint, we mean that the property $f(\vec{p}) = 0$ can be used to define a constraint $C(\vec{p})$ on the object parameters $\vec{p}$ by letting

$$C(\vec{p}) = f(\vec{p}).$$

3.1.1. Circularity of a cylinder

The circularity of a cylinder can be imposed using either Eq. (12) or (15). The last equations have the advantage of

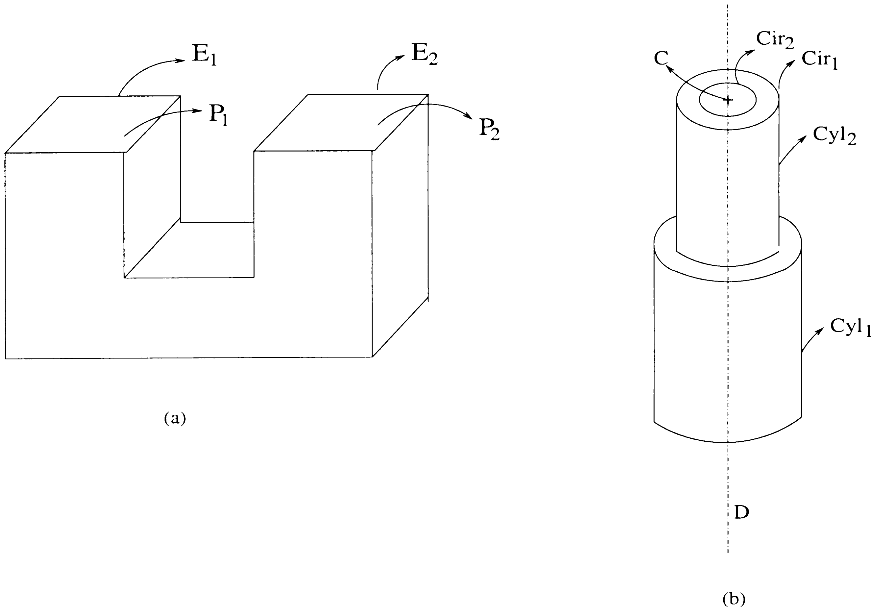


Fig. 1. (a) The two edges E_1 and E_2 belong to the same infinite line. The two faces P_1 and P_2 lie in the same infinite plane. (b) The centres of the circles Cir_1 and Cir_2 coincide at the same point C. The cylinders Cyl_1 and Cyl_2 have a common axis.

imposing implicitly the circularity constraints of the cylinder and avoid the problem when one of the parameters (f, g, h) vanishes. Besides, they make concrete the geometric relationships between the cylinder and other object features as we will see in Section 5.5 (the half cylinder).

3.1.2. Circularity of a cone

This property can be expressed using either Eq. (17) or (19). Similarly to the cylinder case the last equations are more convenient.

3.1.3. Sphere constraint

To require that an ellipsoidal patch represents a perfect sphere, Eq. (20) can be used.

3.2. Feature extrinsic constraints

These constraints reflect the geometric or topological relationships between the different features of one object. Table 1 summarizes the relationships that we have considered. We notice here that points and lines in this table may be either physical features of the object like cone apexes and edges or implicit features like centres, axes of symmetry. This list is not exhaustive and the classification may not be unique. Nevertheless it covers a large number of constraints in manufactured objects.

3.2.1. Coincidence constraints

It is common that a part contains features which are associated with the same geometric entity (Fig. 1(a)) or which coincide at the same position (Fig. 1(b)). In the first case these constraints are implicitly imposed by considering the same parameters for each feature. In the second case the parameters associated to each feature are equated and the resulting equations have then to be satisfied.

3.2.2. Inclusion constraints

A particular feature point may be included in an object feature e.g. line, plane or quadric patch. The inclusion constraint requires that the point satisfies the feature's equation.

A feature line may be included in a plane or a particular quadric surface. Fig. 2 shows an example of this in cylinders. By considering Eqs. (1) and (2), the condition that a line should lie in a plane is:

$$\begin{cases} n_x l + n_y m + n_z n = 0 \\ n_x x_0 + n_y y_0 + n_z z_0 + d = 0 \end{cases}. \tag{23}$$

A necessary and sufficient condition that a line be

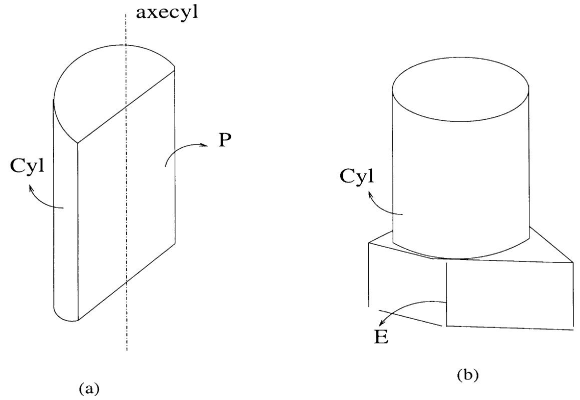


Fig. 2. (a) The axis of the cylinder patch Cyl is included in the plane P. (b) The line associated with the edge E is included in the cylinder Cyl.

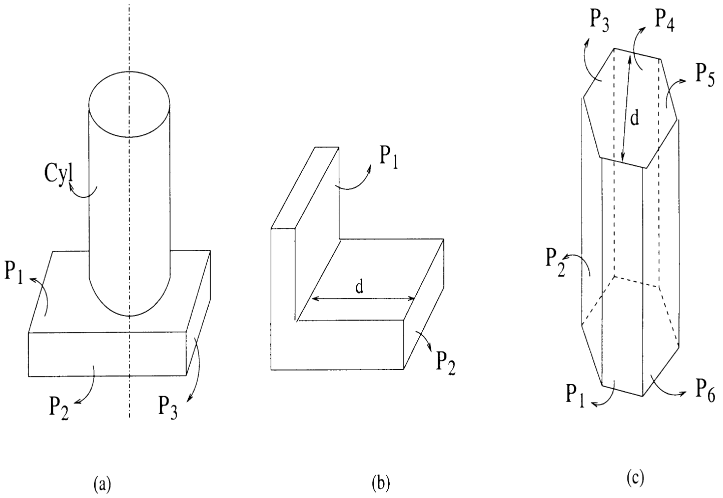


Fig. 3. (a) Each pair of planes (P_1,P_2,P_3) makes an angle of 90°, the axis of the cylinder *Cyl* is orthogonal to P_1. (b) The planes (P_1,P_2) are separated by distance *d*. (c) Each pair of parallel planes of the hexagonal prism is separated by the same distance.

included in a cylinder surface is that the line and the cylinder have the same orientation and an arbitrary point of the line $(X_0, Y_0, Z_0)^{\mathrm{T}}$ satisfies the cylinder equation. Thus, from Eqs. (1), (5) and (11) these conditions can be expressed by

$$\begin{cases} [l, m, n]^{\mathrm{T}} = [1/\mathscr{F}, 1/\mathscr{G}, 1/\mathscr{H}]^{\mathrm{T}} \\ \quad f(X_0, Y_0, Z_0)^{\mathrm{T}} = 0. \end{cases} \quad (24)$$

A line is included in a cone if and only if the orientation vector of the line satisfies the homogeneous equation of the cone (Eq. (5)) without the *u*, *v*, *w* and *d* terms) and it passes through the cone summit. This is formulated then by

$$\begin{cases} \quad f_{\text{homogeneous}}(\vec{p}) = 0 \\ (X_0, Y_0, Z_0) = \text{cone summit} \end{cases} . \quad (25)$$

3.2.3. Relative orientation constraint

There are many orientation relationships which can be deduced and exploited in a given part. In particular, the two common particular cases of parallelism and orthogonality (Fig. 3(a)). The presence of these two characteristics is easily detected in an object. More generally, given a pair of features (F_i,F_j) whose orientations are defined respectively by two vectors $(\vec{n}_i, \vec{n}_j)$ which make an angle α, the relative orientation constraint is expressed by:

$$\vec{n}_i^{\mathrm{T}} \vec{n}_j = \cos(\alpha). \quad (26)$$

3.2.4. Relative separation constraint

The relative separation between features can be exploited when the distance between parallel features (Fig. 3(b)) is already known or needs to be imposed or when the object presents a symmetry aspect leading to some separation distance relationships (Fig. 3(c)). We will take as example the case of planes. Given a pair of parallel planes (P_i,P_j) separated by the algebraic distance *d* (Fig. 3(b)), this constraint is expressed by:

$$d_i + d_j = d \quad (27)$$

where d_i and d_j are the distance parameters associated respectively to P_i and P_j. The planes are oriented in opposite directions.

Given two pairs of parallel planes (P_i,P_j) and (P_k,P_l) separated by the same distance (Fig. 3(c)), the constraint is expressed then by:

$$d_i + d_j = d_k + d_l. \quad (28)$$

3.3. Other constraints

There are also other type of constraints like those imposed directly on the surface parameters as a consequence of the surface representation e.g. the representation of a plane by Eq. (2) requires that the sum of the squared elements of the normal be equal to one. Such constraints will be referenced as the unit constraints.

4. Optimization of shape satisfying the constraints

Given sets of 3D measurement points representing surfaces belonging to a certain object, we want to estimate the different surface parameters, taking into account the geometric relationships between these surfaces and the specific shapes of surfaces as well.

A state vector $\vec{p}$ is associated to the object, which includes all set of parameters related to the different patches. The vector $\vec{p}$ has to best fit the data while satisfying the constraints. Consider $F(\vec{p})$ to be an objective function defining the relationship between the measured data points and the parameters. Such function is generally a minimization criterion (e.g. sum of least squares residuals, maximum likelihood function, etc.).

Consider $C_k(\vec{p}), k = 1 \ldots M$, the set of constraint functions defining the geometric constraints where $C_k(\vec{p})$ is a vector function associated with constraint k. The problem can then be stated as follows:

$$\begin{aligned} &\text{minimize} && F(\vec{p}), \\ &\text{subject to the constraints} && C_k(\vec{p}) \le \tau_k, \qquad k = 1 \ldots M. \end{aligned} \tag{29}$$

Here τ_k represents the tolerance related to the constraint C_k. Ideally the tolerances have zero values, but practically, for geometric constraints they are assigned certain values which reflects the geometric inaccuracies in the relative locations and shapes of features. It is up to the designer to set the tolerances, however an appropriate definition of the tolerances for a given object can be set up by using the scheme developed by Requicha [47].

When faced with an optimization problem it is necessary to know the characteristics of the components of the problem since techniques that solve the problem more efficiently depend mainly on these characteristics. The components of the problem are the objective function and the constraint functions. The characteristics to be investigated are the properties of these functions which include, linearity, smoothness or continuity, differentiability and up to what degree and the form of these functions, quadratic, sum of squared terms, etc.

The computation time of the technique should be taken into account as well. For a reverse engineering task that uses an interactive user environment, designers could not afford to spend hours waiting to get the optimized shape. So a reasonable processing time (in the order of minutes) is a necessary requirement for the optimization technique.

In order to define the appropriate approach let us examine first the components of the problem, the objective function and the constraint functions.

4.1. The objective function

Consider $S_1, \ldots, S_N$ to be the set of surfaces and $\vec{p}_1, \ldots, \vec{p}_N$ the set of parameter vectors related to them. Each vector $\vec{p}_i$ has to minimize a given surface fit error criterion J_i associated with the surface S_i. The set of the parameter vectors has then to minimize the following object function:

$$J = J_1 + J_2 + \cdots J_N. \tag{30}$$

By considering a polynomial description of the surfaces, each surface S_i can be represented by:

$$\vec{h}_i^{\mathrm{T}} \vec{p}_i = 0 \tag{31}$$

where $\vec{h}_i$ is the measurement vector with each component of the form $x^\alpha y^\beta z^\gamma$ for some (α, β, γ).

The advantage of this formulation is that it leads to a compact quadric expression of the objective function because of the linearity (with respect to the parameters) of surface (Eq. (31)). Indeed, given m_i measurements, the least squares criterion related to this equation is

$$J_i = \sum_{l=1}^{m_i} (\vec{h}_i^{l^{\mathrm{T}}} \vec{p}_i)^2 = \vec{p}_i^{\mathrm{T}} H_i \vec{p}_i, \qquad H_i = \sum_{l=1}^{m_i} (\vec{h}_i^l \vec{h}_i^{l^{\mathrm{T}}}) \tag{32}$$

where H_i represents the sample covariance matrix of the surface S_i. By concatenating all the vectors $\vec{p}_i^{\mathrm{T}}$ into one vector $\vec{p} = [\vec{p}_1^{\mathrm{T}}, \vec{p}_2^{\mathrm{T}}, \ldots, \vec{p}_N^{\mathrm{T}}]^{\mathrm{T}}$ Eq. (30) can be written as a function of the parameter vector $\vec{p}$ and we get the following objective function:

$$F(\vec{p}) = J = \vec{p}^{\mathrm{T}} \mathscr{H} \vec{p}, \qquad \mathscr{H} = \begin{bmatrix} H_1 & (0) & \cdot & (0) \\ (0) & H_2 & \cdot & (0) \\ (0) & \cdot & \cdot & (0) \\ (0) & \cdot & (0) & H_N \end{bmatrix}. \tag{33}$$

Under the above form, the objective equation contains separate terms for the data and the parameters. The data matrix $\mathscr{H}$ can be thus computed off-line before the optimization.

The inconvenience of the polynomial representation (31) of the surfaces is that it may over-parametrize the surface. For example a circular cone and circular cylinder have 10 parameters if they are represented by the quadric equation (5) whereas they actually need only 7 parameters (see (14) and (18)). Furthermore, the reduced representation imposes implicitly the circularity constraint consequently there is no need to formulate this constraint within a constraint function. However, the implementation of the reduced form in the optimization algorithm may cause some complexity, indeed because of the nonlinearity of the these forms, it has not been possible to get an objective function with separated terms for the data and the parameters. Thus, the data terms could not be computed off-line. This may increase the computational cost dramatically.

The objective function could be taken as the likelihood of the range data given the parameters (with a negative sign since we want to minimize). The likelihood function has the advantage of accounting for the statistical aspect of the measurements. As a first step, we have chosen the least squares function. The integration of the data noise characteristics in the LS function can be done afterwards with no particular difficulty, leading to the same estimation of the likelihood function in the case of the Gaussian distribution.

4.2. The constraint functions

The geometric constraints include some linear constraints (e.g. the relative separation constraint) and mainly nonlinear constraints (e.g. relative orientation constraint).

A matrix representation can hold all the types of the constraints mentioned earlier. It leads to a compact form and avoid expressions with many variables. As it will be shown later in the experiment sections, a close examination of the nonlinear constraints shows that they can be represented by expressions containing cross-product terms of at most 2 parameters. Thus they can represented by the quadratic vector function:

$$\vec{p}^{\mathrm{T}} A \vec{p} + B^{\mathrm{T}} \vec{p} + C \tag{34}$$

where A and B are, respectively, a square matrix and a vector having the same dimension than the parameter vector $\vec{p}$, C is a scalar. This representation can also include linear constraints by setting the matrix A to zero. In the following sections the constraint functions will use the matrix and vector notation defined in Appendix A.

4.2.1. Example

The slot shown in Fig. 4 contains three surfaces. The two parallel surfaces (S_1,S_2) have been associated with a single normal vector $\vec{n}_1$ and the surface S_3 is oriented by the normal $\vec{n}_3$. The three surfaces are then defined, respectively, by $(\vec{n}_1, d_1), (\vec{n}_1, d_2)$ and $(\vec{n}_3, d_3)$. The parameters of the slot can be then encapsulated in the vector $\vec{p} = [\vec{n}_1^{\mathrm{T}}, d_1, d_2, \vec{n}_3^{\mathrm{T}}, d_3]^{\mathrm{T}}$. The fixed distance constraint between the surfaces S_1 and S_2 and the orthogonality constraint between (S_1,S_2) and S_3 are represented, respectively, by:

$$d_2 - d_1 = d$$

$$\vec{n}_1^{\mathrm{T}} \vec{n}_3 = 0.$$

The first constraint is linear and can be put into the form

$$B^{\mathrm{T}} \vec{p} + C = 0, \qquad B = [0, 0, 0, -1, 1, 0, 0, 0, 0]^{\mathrm{T}},$$

$$A = [0], \qquad C = -d.$$

The second constraint is nonlinear and can written under the quadratic form:

$$\vec{p}^{\mathrm{T}} A \vec{p} = 0 \; A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = [0], \qquad C = 0.$$

4.3. The optimization techniques

Optimization techniques fall into two broad branches namely, Operation Research techniques and the recent evolutionary techniques.

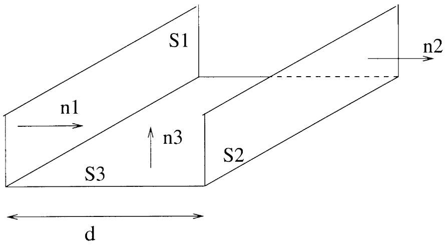


Fig. 4. A slot with two parallel planes orthogonal to a third plane.

Evolutionary computation techniques [48,49] have been having increasing attraction for their potential to solve complex problems. In short they are stochastic optimization methods. They are conveniently presented using the metaphor of natural evolution: they start from a randomly generated set of points or solutions of the search space (population of individuals). Then this set evolves following a process close the natural selection principle. At each stage a new set of population is generated using simulated genetic operations such as mutation or crossover. The probability of survival of the new solutions depends on how well they fit a given evaluation function. The best are kept with high probability and the worst are discarded. This process is repeated until the set of solutions converges to the one best fitting the evaluation function.

The main advantages of the evolutionary techniques is that they do not have much mathematical requirements about the optimization problem. They are 0-order methods, in the sense that they operate only on the objective function and they can handle linear or nonlinear problems, constrained or unconstrained.

The main drawback of these techniques is that they are highly time consuming. This is due to the fact that to ensure convergence, the number of generated solutions has to be high, and at each iteration all the solutions have to be evaluated. This increases the computation time dramatically.

In CAD applications these techniques, and in particular, the genetic algorithms have been used in product shape design [50], manufacturing feature extraction [51], description capture from range data [52] and design specification and evaluation [53].

The second branch of the optimization techniques are the classical operation research techniques. They are more mature than the evolutionary techniques. They involve search techniques, numerical analysis and differential tools. Most of these techniques use an iterative scheme. A reasonable initialisation causes significant speedup in convergence. A detailed review and analysis of these optimization techniques could be found in Refs. [54,55]. Descent methods, for instance the Newton–Raphson minimization was used in constraint solving [5,6] and surface meshing [56]. Quadratic programming and sequential quadratic programming were used for curve and surface optimization [57,58].

4.3.1. Which technique should be adopted?

We believe that the evolutionary techniques are suitable mainly to the optimization cases where objective functions and constraints are very complex, presenting hard-handled aspects such nonlinearity, non-differentiability, or do have not explicit forms. Indeed the earlier mentioned characteristics of these techniques allow them to by-pass these problems.

As our optimization problem does not have these problems, the operational research techniques are more appropriate. This argument is supported by the time-consuming characteristic of the evolutionary techniques, where the average scale of the processing time is of the order of hours. This characteristic makes these methods not appropriate for interactive user environment and impractical for a static verification and checking of the results when experiments have to be repeated many times. The other important reason for opting for search techniques is that we can obtain a reasonable initial estimate of the model parameters. This initial solution is the estimation of the model parameters without considering the constraints. This estimation is not far away from the optimal one since it is obtained from the real object prototype.

4.3.2. The optimization algorithm

Theoretically a solution of the problem stated in (29) is given by finding the set $(\vec{p}, \lambda_1, \lambda_2, \ldots, \lambda_M)$ minimizing the following equation:

$$E(\vec{p}) = F(\vec{p}) + \sum_{k=1}^{M} \lambda_k C_k(\vec{p}) \tag{35}$$

$$F(\vec{p}) = \vec{p}^{\mathrm{T}} \mathscr{H} \vec{p}$$

$$C_k(\vec{p}) = \vec{p}^{\mathrm{T}} A_k \vec{p} + B_k^{\mathrm{T}} \vec{P} + C_k.$$

Under the Khun–Tucker conditions [54, Chapter 9], namely that the objective function and the constraint functions are continuously differentiable and the gradients of the constraint functions are linearly independent, the optimal set $(\vec{p}, \lambda_1, \lambda_2, \ldots, \lambda_M)$ minimizing (35) is solution of the system:

$$\frac{\partial F}{\partial \vec{p}} + \sum_{k=1}^{M} \lambda_k \frac{\partial C_k}{\partial \vec{p}} = 0. \tag{36}$$

In some particular cases it is possible to get a closed form solution for (36) such as the generalized eigenvalues methods. This depends on the characteristics of the constraint functions and whether it is possible to combine them efficiently with the objective function. When the constraints are linear (having the form $A\vec{p} + B = 0$) the standard quadratic programming methods could be applied to solve this system.

However the geometric constraints are mainly nonlinear. Generally it is not trivial to develop an analytical solution for such problem. In this case an algorithmic numerical approach could be of great help taking into account the increasing capabilities of computing.

Now if we look to the objective function and the constraint functions in (35) we see that they are explicitly defined in function of the parameters, they are smooth, differentiable and they both have a quadratic structure. From (32) we can notice that each submatrix H_{i} of $\mathscr{H}$ in (33) is the sum of cross-product terms $\vec{h}_i^l \vec{h}_i^{l^{\mathrm{T}}}$. Thus H_i as well as $\mathscr{H}$ is positive definite. Consequently the objective function is convex. Such functions could be efficiently minimized. Besides it has the important property that its minimum is global. If the constraint functions are also convex, the optimization problem (35) would be a convex optimization problem for $\lambda_k > 0$. For such problem an optimal solution exists, moreover this solution corresponds to the solution of the system (36) defined by the Khun–Tucker conditions [59, Sections 27 and 28].

The constraint functions are not necessarily convex since their related matrix A is not necessarily positive definite. However the squared constraint function will have a Hessian matrix which is positive and definite, so is a convex function. The whole optimization function $E(\vec{p})$ in (35) will be then a convex function. So by considering the squared constraint function the problem would be to determine the set $(\vec{p}, \lambda_1, \lambda_2, \ldots, \lambda_M)$ minimizing:

$$E(\vec{p}) = F(\vec{p}) + \sum_{k=1}^{M} \lambda_k (C_k(\vec{p})^2), \qquad \lambda_k > 0. \tag{37}$$

To provide a numerical solution of this problem we have been investigating an approach in the framework of sequential unconstrained minimization. The basic idea is to attach different penalty functions to the objective function $F(\vec{p})$ in such a way that the optimal solutions of successive unconstrained problems approach the optimal solution of the problem (37). Indeed the term $\sum_{k=1}^{M} \lambda_k (C_k(\vec{p})^2)$ could be seen as a penalty function controlling the constraints satisfaction. The scheme is then increment the set of λ_k iteratively, at each step minimize (37) by a standard non-constrained technique, update the solution $\vec{p}$, and repeat the process until the constraints are satisfied. For equal values of λ_k, Fiacco and McCormick [60] have shown that the solutions of (37) converge towards the same solution of the problem (29) when λ_k tends to infinity.

In more detail the proposed algorithm is: We start with a parameter vector $\vec{p}^{[0]}$ that minimizes the least squares objective function and attempt to find a nearby vector $\vec{p}^{[1]}$ that minimizes (37) for small values of λ_k. Then we iteratively increase the set of λ_k slightly and solve for a new optimal parameter $\vec{p}^{[n+1]}$ using the previous $\vec{p}^{[n]}$. At each iteration n, the algorithm increases each λ_k by a certain amount and a new $\vec{p}^{[n]}$ is found such that the optimization function is minimized by means of the standard Levenberg–Marquardt algorithm (see Appendix C). The parameter vector $\vec{p}^{[n]}$ is then updated to the new estimate $\vec{p}^{[n+1]}$ which becomes the initial estimate at the next values of λ_k. The algorithm stops

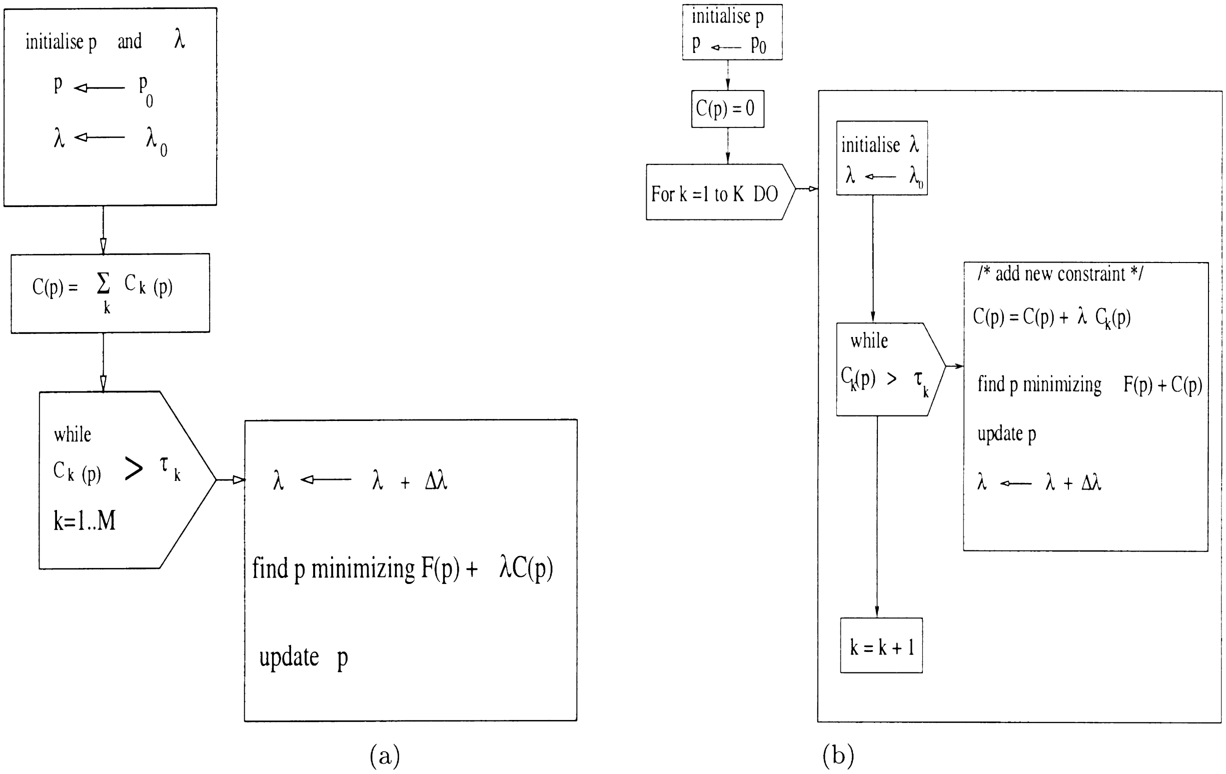


Fig. 5. (a) *optim*1–batch constraint optimization algorithm. (b) *optim*2–sequential constraint introduction optimization algorithm.

when the constraints are satisfied to the desired degree or when the parameter vector remains stable for a certain number of iterations. A simplified version of the algorithm is illustrated in Fig. 5(a) in which a single λ is associated to the constraints.

A computational problem associated with this algorithm emerges when λ_k becomes too large. This problem arises in the Hessian matrix of the optimization function (37) involved in Levenberg–Marquardt algorithm. This matrix become ill-conditioned for high values of λ_k. This aspect could be detected from the expression of this matrix:

$$\mathrm{Hess}(E(\vec{p})) = 2\mathscr{H} + \sum_{k=1}^{M} 4\lambda_k C_k(\vec{p}) A_k + R^{\mathrm{T}} D R \tag{38}$$

where

$$R^{\mathrm{T}} = \left[\frac{\partial C_1}{\partial \vec{p}} \frac{\partial C_2}{\partial \vec{p}} \cdots \frac{\partial C_M}{\partial \vec{p}} \right]$$

$$D = \begin{bmatrix} 2\lambda_1 & 0 & \ldots & 0 \\ 0 & 2\lambda_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 2\lambda_M \end{bmatrix}.$$

The rank of R is equal to M since we assume that the derivatives of the constraint functions are linearly independent. $R^{\mathrm{T}}DR$ will have also a rank equal to M and since D is a diagonal matrix, the M non-null eigenvalues values of $R^{\mathrm{T}}DR$ will depend on λ_k. More exactly, each eigenvalue has the form $\sigma_k\lambda_k$ where σ_k is some coefficient. Thus the norm of $R^{\mathrm{T}}DR$ will increase as λ_k increases. This is not the case with the other terms of the Hessian matrix (38). Indeed, $\mathscr{H}$ is independent of λ_k and the product $\lambda_k C_k(\vec{p})$ in the other term is expected either to vanish or to remain stable since the constraint value $C_k(\vec{p})$ decreases as λ_k increases. So M eigenvalues of the Hessian matrix (38) will increase with λ_k whereas the others $N - M$ remain independent and not affected. Consequently as λ_k values increase and become large the condition number of the Hessian matrix of the optimization function increases and the matrix become ill-conditioned. Consequently the computation of the inverse of the Hessian matrix in the Levenberg–Marquardt algorithm will suffer from high numerical instability and this approach will no longer be appropriate. Broyden et al. [61] have developed a method to overcome this numerical problem. Their method is applied with penalty function having equal weight for all the constraints. We have extended the application of this method to different weights of the constraints. The details are developed in Appendix D.

The initialization of the parameter vector is crucial to guarantee the convergence of the algorithm to the desired solution. For this reason the initial vector was the one which best fitted the set of data in the absence of constraints. This vector can be obtained by estimating each surface's parameter vector separately and then concatenating the vectors into a single one. Naturally, the option of minimizing the objective function $F(\vec{p})$ alone has to be avoided since it leads to the trivial null vector solution. On the other hand, the initial values λ_k have to be not too small in order to avoid the above trivial solution and to give the constraints a certain weight. Practically this condition should be applied only to the unit constraints (e.g. the normals of the plane surfaces or quadric axis have to be unit). A convenient value for the initial λ_k is:

$$\lambda_k^0 = \frac{F(\vec{p}^{[0]})}{C_k(\vec{p}^{[0]})} \tag{39}$$

where $\vec{p}^{[0]}$ is the initial parameter estimation obtained by concatenating the unconstrained estimates. This ensures the objective function and the penalty functions have similar values at the first minimization.

Another option of the algorithm consists of adding the constraints incrementally. At each step a new constraint is added to the constraint function $C(\vec{p})$ and then the optimal value of $\vec{p}$ is found according to the scheme shown in Fig. 5(b). For each new added constraint $C_k(\vec{p})$, λ_k is initialized at λ_k^0, whereas $\vec{p}$ is kept at its current value.

5. Experiments

The experiments were carried out on both synthetic and real data. The real data was acquired from test objects with a 3D triangulation range sensor. The range measurements were already segmented into point sets associated with surfaces by means of the RANGESEG [62] program.

The first experiments aimed to check the behaviour and the convergence of the algorithm. These experiments were applied on surfaces extracted from a single view of polyhedral objects. Through these experiments the performances of the batch version of the algorithm (*optim*1) and the sequential version (*optim*2) were compared (see section “The step model object”).

In the second series of experiments (second subsection) we have gone further in complexity, firstly on the level of types of features. Thus, objects containing quadric features were examined. Secondly the range data was collected and registered from different views. Thus, the data was additionally corrupted by the registration errors. So one objective was to test the robustness of the algorithm toward the complexity of the features (thus the diversity of the constraints) and the registration errors.

At first, objects containing single quadric feature were studied. Section 5.5 (half cylinder) deals with the cylinder case. Multi-quadric objects were examined afterwards (Sections 5.7 (multi-quadric object 1) and 5.8 (multi-quadric object 2)). For the first category we have compared results issued from a single view with those obtained from multiple views. For both categories we checked the impact of constraint satisfaction on the quality of object shape attribute estimation.

Other tests were carried out in order to give answers to the following questions:

1. What happens when some features are left unconstrained? What is the impact on the other constrained features and more generally on the object shape estimation? Reciprocally what is the impact of the constrained features on the non-constrained ones?
2. How stable is the algorithm?
3. How optimal is the solution?
4. What happens if some constraints are invalid or inconsistent?

Experiments carried on the synthetic polyhedral object (step model object) will give preliminary answers to question 1. Trials on real multi-quadric objects (Sections 5.7 (multi-quadric object 1) and 5.8 (multi-quadric object 2)) will bring additional confirmation.

Answers to questions 2, 3, 4 will be developed in the experiments of Section 5.8 (multi-quadric object 2).

5.1. Application to polyhedral objects

Polyhedral objects involve mainly relative orientation constraints and relative separation constraints. Consider N plane surfaces defining a polyhedral object represented by a parameter vector $\vec{p}$.

Given a pair of planes (P_i, P_j) whose normals $(\vec{n}_i, \vec{n}_j)$ make an angle $\alpha_{i,j}$, the angle constraint (26) is expressed by:

$$C_{\text{angle}_{(i,j)}}(\vec{p}) = (\vec{p}^{\mathrm{T}} \mathscr{A}_{i,j} \vec{p} - 2\cos(\alpha_{i,j}))^2 = 0 \tag{40}$$

where $\mathscr{A}_{i,j}$ is an $N \times N$ matrix which according to the notation of Appendix A is defined by $\mathscr{A}_{i,j} = L_{(r,s,2)}$ where r and s are, respectively, the indices of the first element of $\vec{n}_i$ and $\vec{n}_j$ in the parameter vector $\vec{p}$.

The separation constraints (27) and (28) are, respectively, expressed by (see Appendix A):

$$C_{\text{dist}_{(i,j)}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(r,s)} \vec{p} - d)^2 = 0 \tag{41}$$

$$C_{\text{dist}_{(i,j,k,l)}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(r,s,t,l)} \vec{p})^2 = 0 \tag{42}$$

where r,s,t,l represent the indices of the distance parameters d_i, d_j, d_k, d_l in the parameter vector $\vec{p}$.

Additionally, the unit constraint has to be taken into account since the plane’s orientation is defined by a unit normal. For a given surface plane P_i, whose normal is $\vec{n}_i$, this constraint is expressed by:

$$C_{\text{unit}_i}(\vec{p}) = (\vec{p}^{\mathrm{T}} \mathscr{U}_i \vec{p} - 1)^2 = 0 \tag{43}$$

where $\mathscr{U}_i$ is an $N \times N$ matrix which, according to the notation of Appendix A is defined by $\mathscr{U}_i = U_{(r,r+2)}$, where r is the index of the first element of $\vec{n}_i$ in the parameter vector $\vec{p}$.

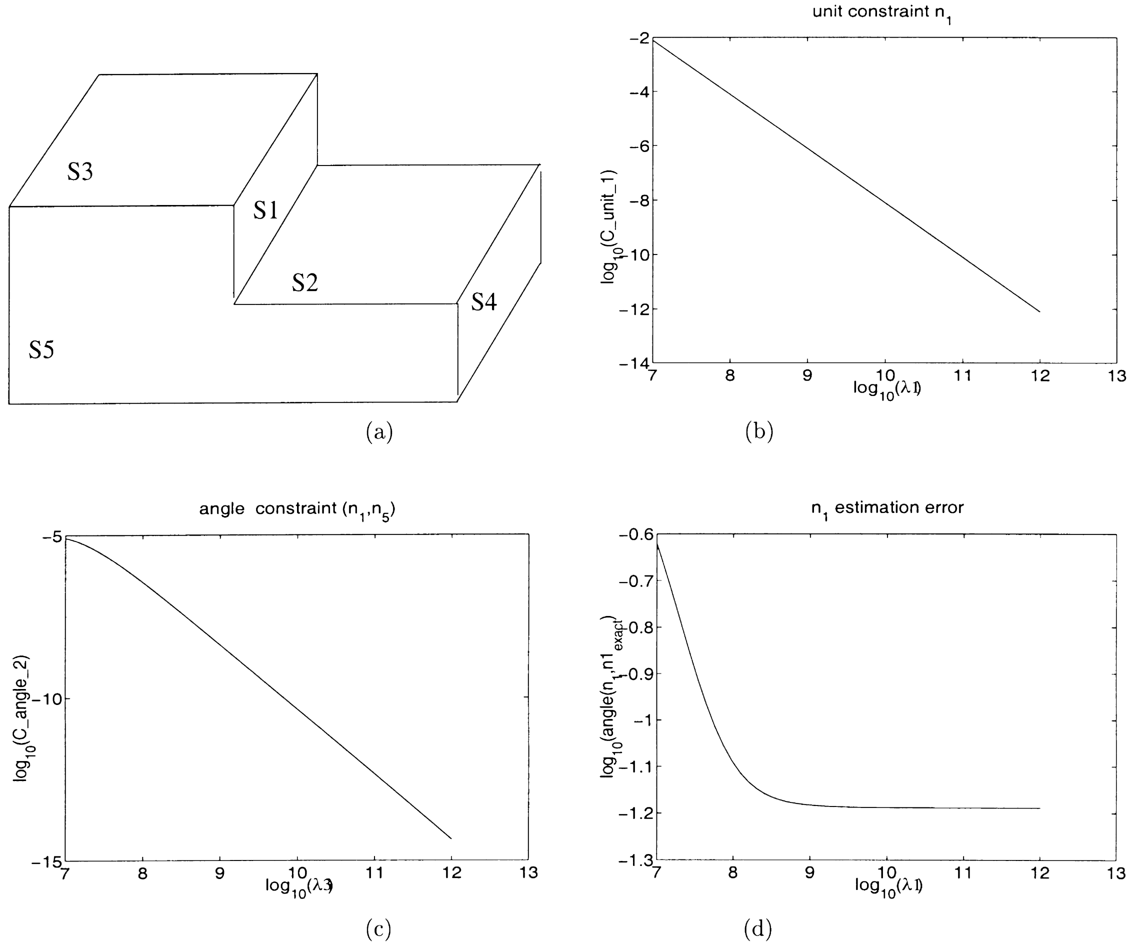


Fig. 6. (a) The step model object. (b) Variation of the unit constraint error as a function of the related λ. (c) Variation of the angle constraint error $(\vec{n}_1, \vec{n}_2)$ as a function of the related λ. (d) Error of the estimated orientation of the normal $\vec{n}_1$ (the error is represented by the angle between the estimated normal and the actual one).

5.2. The step model object

The first series of tests used a synthetic step model object. This object contains sets of parallel planes. The prototype object is composed of eight faces. We have studied the case when five faces are visible (Fig. 6(a)). For this view we assigned a single normal to each set of parallel planes. Three normals $\vec{n}_1, \vec{n}_2, \vec{n}_5$ are associated, respectively, to surfaces (S_1,S_4), (S_2,S_3), and S_5. So, there are three angle constraints (orthogonality of each two normals) and the three unit vector constraints.

The set of visible surfaces is defined by the parameter vector

$$\vec{p} = [\vec{n}_1^{\mathrm{T}}, d_1, d_4, \vec{n}_2^{\mathrm{T}}, d_2, d_3, \vec{n}_5^{\mathrm{T}}, d_5]^{\mathrm{T}}.$$

Using the paradigm the Section 4.1 (representation of the objective function), the objective function associated with this object is expressed by:

$$F(\vec{p}) = \vec{p}^{\mathrm{T}}\mathscr{H}\vec{p}, \qquad \mathscr{H} = \begin{bmatrix} G_{1,4} & (0)_{5,5} & (0)_{5,4} \\ (0)_{5,5} & G_{2,3} & (0)_{5,4} \\ (0)_{4,5} & (0)_{4,5} & G_5 \end{bmatrix} \tag{44}$$

where

$$G_{1,4} = \begin{bmatrix} H_1 + H_4 & h_1 & h_4 \\ h_1^{\mathrm{T}} & N_1 & 0 \\ h_4^{\mathrm{T}} & 0 & N_4 \end{bmatrix},$$
$$G_{2,3} = \begin{bmatrix} H_2 + H_3 & h_2 & h_3 \\ h_2^{\mathrm{T}} & N_2 & 0 \\ h_3^{\mathrm{T}} & 0 & N_3 \end{bmatrix} \qquad G_5 = \begin{bmatrix} H_5 & h_5 \\ h_5^{\mathrm{T}} & N_5 \end{bmatrix} \tag{45}$$

and $H_k = \sum_i^{N_k} (X_i^k)(X_i^k)^{\mathrm{T}}, h_k = \sum_i^{N_k} X_i^k$.

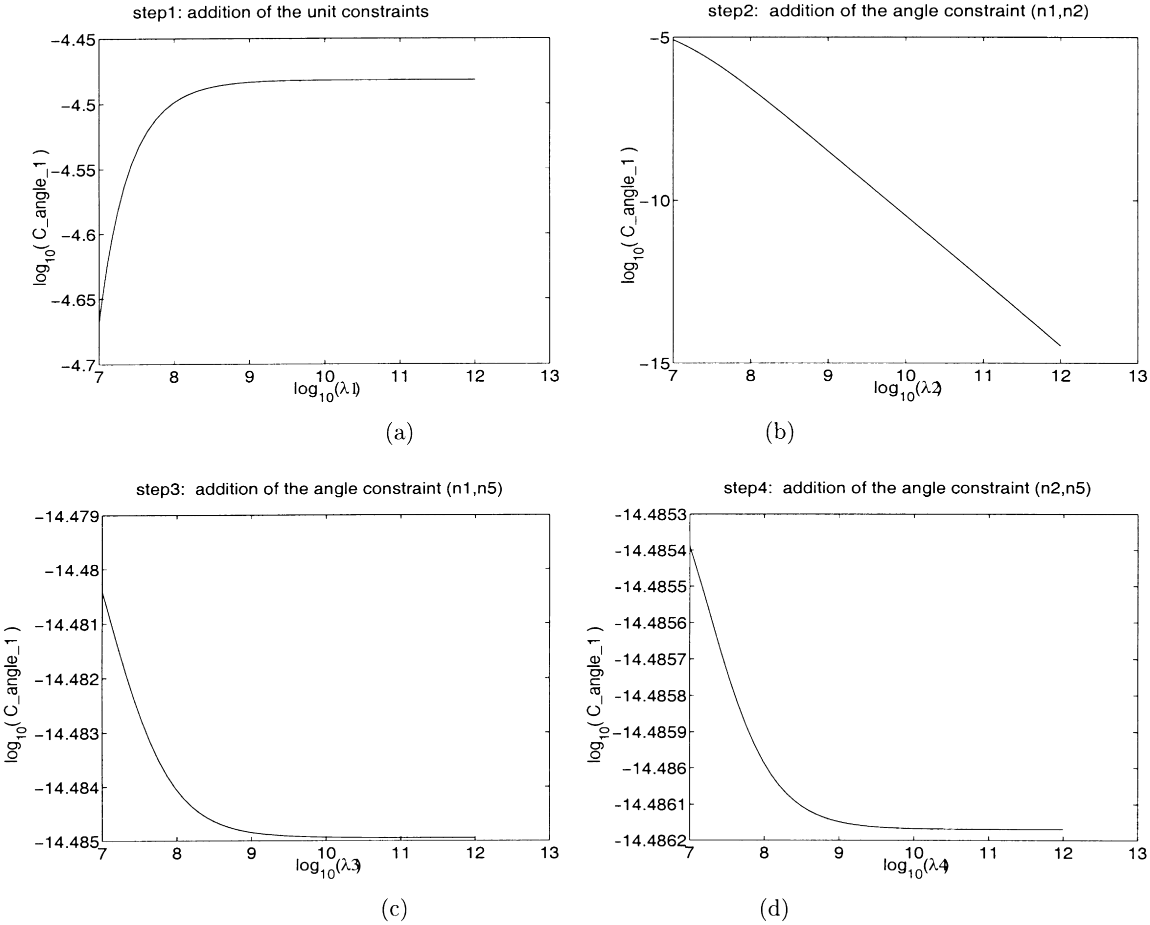
X_i^k is a data point which belongs to the plane surface S_k and N_k is the number of points of the plane S_k.

The normals $\vec{n}_1, \vec{n}_2, \vec{n}_5$ are orthogonal and have to be unit so we set the following constraint functions:

$$C_{\mathrm{unit}_1}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{U}_1\vec{p} - 1)^2 = 0 \tag{46}$$

$$C_{\mathrm{unit}_2}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{U}_2\vec{p} - 1)^2 = 0 \tag{47}$$

$$C_{\mathrm{unit}_5}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{U}_5\vec{p} - 1)^2 = 0 \tag{48}$$



(a) (b) (c) (d)

Fig. 7. Variation of the angle constraint value C_{angle_1} (50) at the four steps of the algorithm. Step 1 (a) Only the unit constraints are considered in the optimization function (see (52)). Step 2 (b) The angle constraint function C_{angle_1} is added to the optimization function (see (53)). Step 3 (c) Addition of the constraint function C_{angle_2} and Step 4 (d) Addition of the constraint function C_{angle_3}.

$$C_{\mathrm{angle}_1}(\vec{p}) = C_{\mathrm{angle}_{(1,2)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{A}_{1,2}\vec{p} - 2\cos(\pi/2))^2 = 0 \tag{49}$$

$$C_{\mathrm{angle}_2}(\vec{p}) = C_{\mathrm{angle}_{(1,5)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{A}_{1,5}\vec{p} - 2\cos(\pi/2))^2 = 0 \tag{50}$$

$$C_{\mathrm{angle}_3}(\vec{p}) = C_{\mathrm{angle}_{(2,5)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathscr{A}_{2,5}\vec{p} - 2\cos(\pi/2))^2 = 0. \tag{51}$$

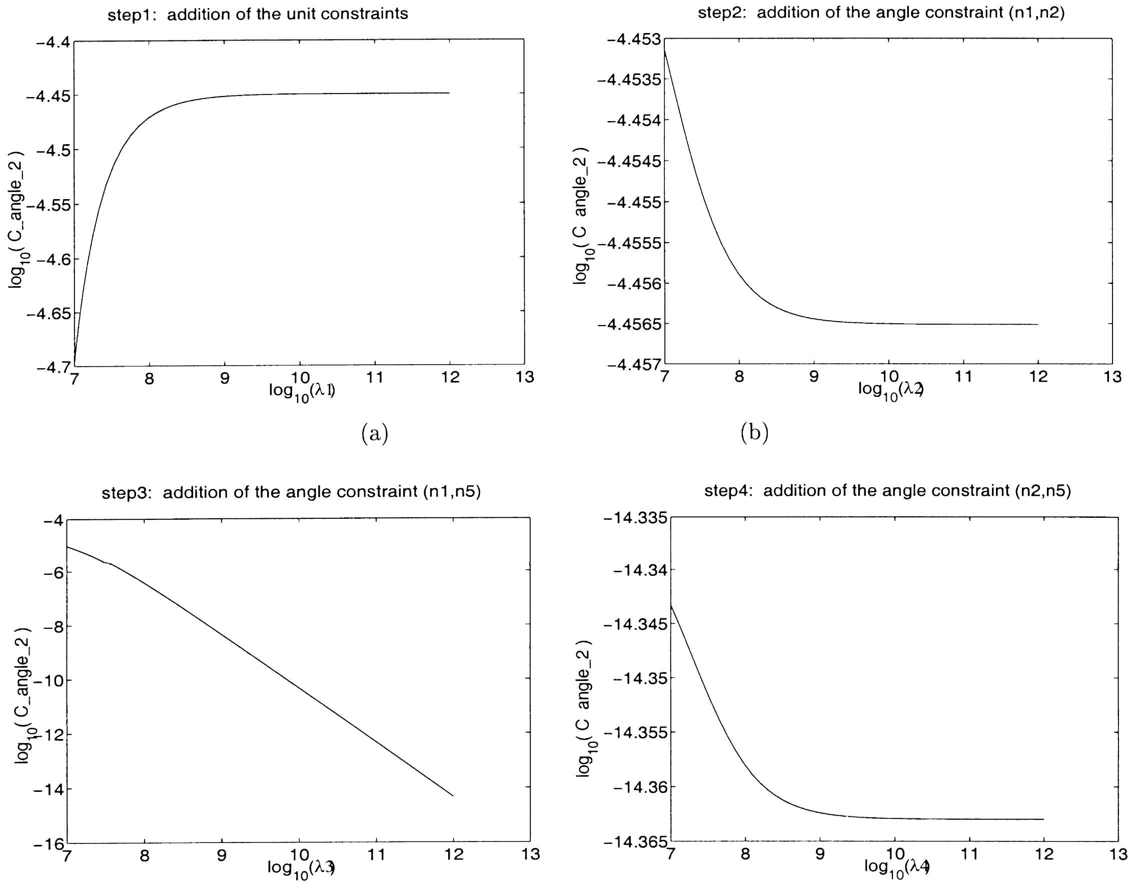
Since the unit constraints are used mainly to avoid the null solution, a single λ is associated to them. The optimization function is then

$$E(\vec{p}) = \vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} + \lambda_1(C_{\mathrm{unit}_1} + C_{\mathrm{unit}_2} + C_{\mathrm{unit}_5})(\vec{p}) + \lambda_2 C_{\mathrm{angle}_1}(\vec{p}) + \lambda_3 C_{\mathrm{angle}_2}(\vec{p}) + \lambda_4 C_{\mathrm{angle}_3}(\vec{p}).$$

The results shown below are the average of 100 trials. At each trial the surfaces' points are randomly corrupted with a Gaussian noise of 3 mm standard deviation. Then *optim*1 and *optim*2 are applied to the same set of data points.

Fig. 6 shows some results obtained with the algorithm *optim*1. These results represent the variation and the behaviour of some constraint functions during the algorithm with respect of their associated λ. Other results represent the variation of the estimation error of some of the object parameters e.g. one surface's normal. The actual normals are known since the object is simulated.

Fig. 6(b) shows the decrease of the unit constraint function (46) as λ_1 increases, similarly for the angle constraint function (50) which decreases as the associated weight λ_3 increases in Fig. 6(c). We notice that both functions are decreasing nearly linearly at a logarithmic scale. This suggests that it is possible to enforce the constraint to any level of tolerance until the numerical accuracy of the algorithm is compromised. The orientation error related to the surface normal $\vec{n}_1$ and represented here by the angle formed by the actual normal and the estimated one decreases and stabilizes to a relatively low value (around 0.06°) in Fig. 6(d). This error is computed as follows: at each iteration of the algorithm *optim*1 the estimated normal $\vec{n}_1$ is extracted from the solution $\vec{p}$ and then the error with respect to the actual simulated $\vec{n}_1$ is computed. At each iteration the values of the different λ change, but the orientation error is mapped



(a) (b)

(c) (d)

Fig. 8. Variation of the angle constraint value C_{angle_2} (50) at the four steps of the algorithm *optim*2.

as function of λ_1 just to show its variation although it is not depending on λ_1 in particular.

Similar behaviour is observed for the other parameter vectors but they are not shown here to save space. This first observation of the constraints behaviour and the parameter estimation is encouraging because it means that the part's shape and position stabilizes as a whole. This fact will be confirmed in next experiments with other objects.

The three figures Figs. 7–9 illustrate, respectively, the variation of the angle constraints values C_{angle_1} (49), C_{angle_2} (50) and C_{angle_3} (51) during the application of the sequential version *optim*2. The optimization process has four steps, first the unit constraints are inserted then the three angle constraints are applied one by one. So that at the first step the optimization function is:

$$E(\vec{p}) = \vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} + \lambda_1(C_{\text{unit}_1} + C_{\text{unit}_2} + C_{\text{unit}_5})(\vec{p}). \tag{52}$$

In the second step it will be

$$E(\vec{p}) = \vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} + \lambda_1(C_{\text{unit}_1} + C_{\text{unit}_2} + C_{\text{unit}_5})(\vec{p}) + \lambda_2 C_{\text{angle}_1}(\vec{p}) \tag{53}$$

and so on.

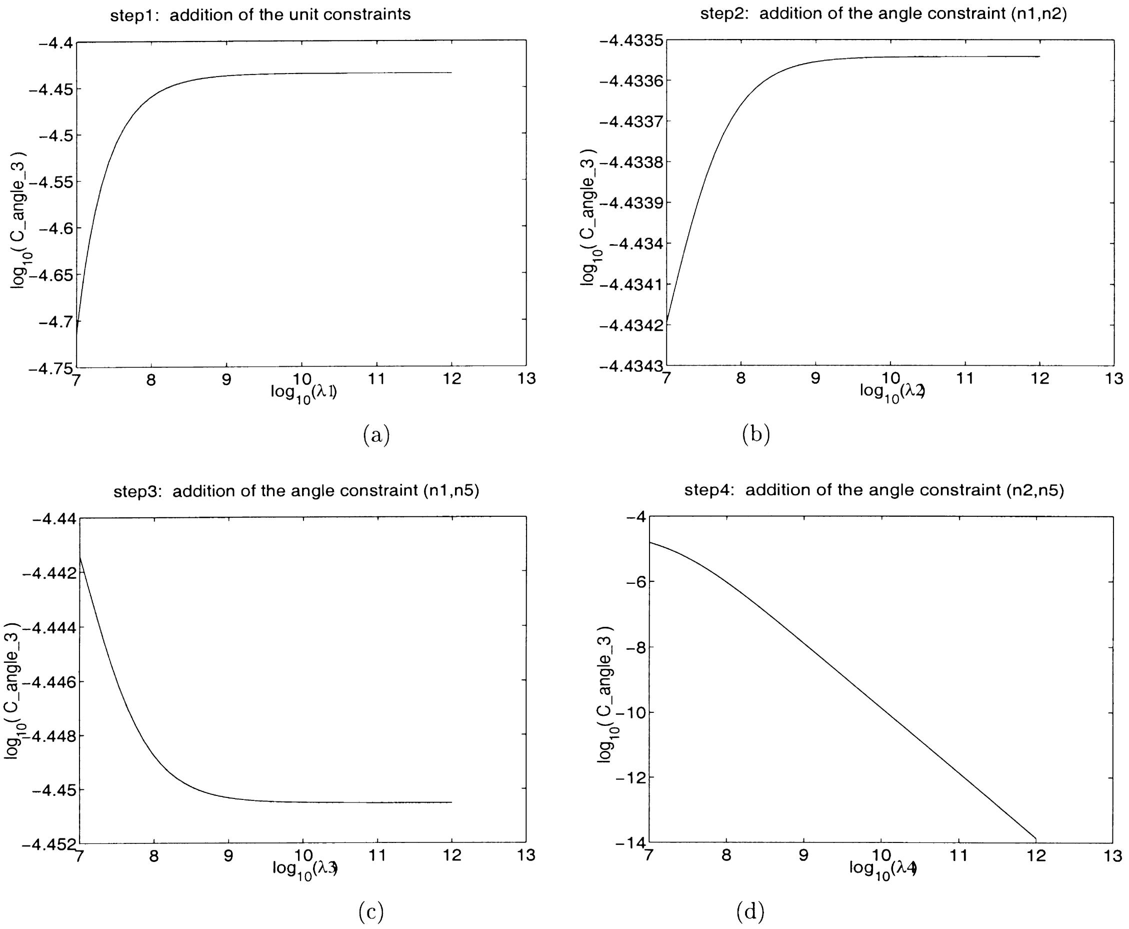
The figures shows clearly the significant decrease of the constraint value when the related constraint function is added to the optimization function. It is seen also that once the constraint is satisfied the addition of the other constraints only affects the level to tolerance previously reached by a very small degree.

In Fig. 7, it is noticed that at the end of step 2 (Fig. 7(b)) the constraint C_{angle_1} is well satisfied although the two others are not yet. Similarly, Fig. 8(c) shows that at the end of step 3 the constraint C_{angle_2} is well satisfied while the constraint C_{angle_3} is not yet implemented.

Fig. 9 shows that during steps 2 and 3 (when C_{angle_1} and C_{angle_2} are applied) the constraint C_{angle_3} almost keeps stable at a reasonable value. This means that the satisfaction of some constraints is not performed at much cost to the unconstrained features.

Fig. 10 shows the variation of the estimation error of one normal $\vec{n}_1$ along the four steps of the algorithm. Similar results are obtained for the other normals. Similar to experiment with *optim*1, Fig. 10(d) shows that at the end of the optimization the error in $\vec{n}_1$ estimation stabilizes at a low value. The same is noticed for the other normals.

So the experiments carried out on the step model object have provided evidence of the applicability of both versions



(a) (b)

(c) (d)

Fig. 9. Variation of the angle constraint value C_{angle_3} (51) at the four steps of the algorithm *optim*2.

*optim*1 and *optim*2 of the algorithm. Both versions offer high satisfaction of the constraints, moreover the estimated orientation of the object surfaces extracted from the algorithm's solutions are close to the actual one in both versions. This goes towards saying that the satisfaction of object shape requirements is not performed at the expense of object localization, although the purpose of the algorithm is not to recover the object localization.

However, *optim*2 is more time-consuming than *optim*1 (around N times, where N is the number of constraints). So, since both estimation of *optim*1 and *optim*2 are acceptable, we have preferred to use *optim*1 for the rest of the work.

5.3. The tetrahedron

The second polyhedral object tested is a real tetrahedron. The data has been extracted from a view representing three visible faces S_1, S_2, S_3 (Fig. 11). The parameter vector is $\vec{p} = [\vec{p}_1^{\mathrm{T}}, \vec{p}_2^{\mathrm{T}}, \vec{p}_3^{\mathrm{T}}]^{\mathrm{T}}$.

In this view, the object surfaces have three angle constraints represented by the three angles 90°, 90° and 120° between the three surface normals, as well as the unit vector constraints. So we define the following constraint functions:

$$C_{\text{unit}_1}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{U}_1\vec{p} - 1)^2 = 0 \tag{54}$$

$$C_{\text{unit}_2}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{U}_2\vec{p} - 1)^2 = 0 \tag{55}$$

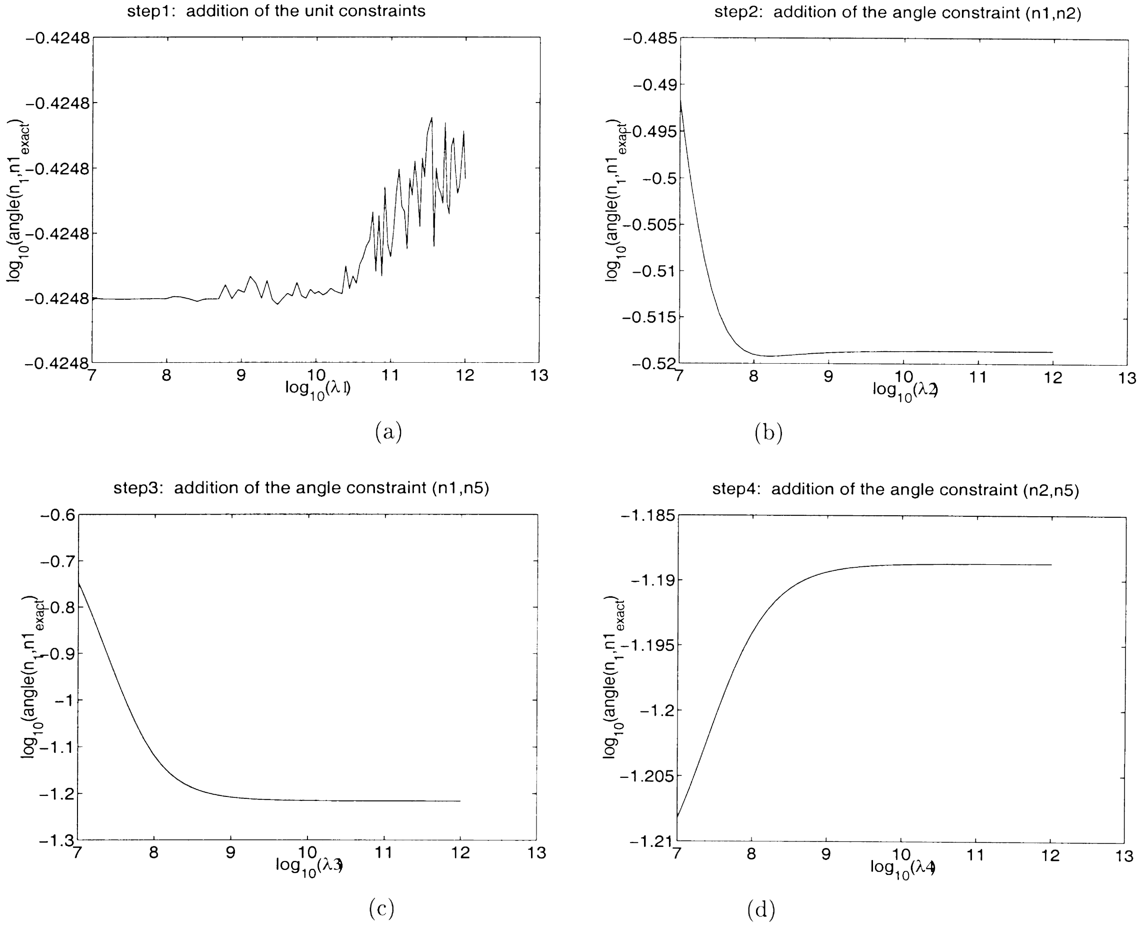
$$C_{\text{unit}_3}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{U}_3\vec{p} - 1)^2 = 0 \tag{56}$$

$$C_{\text{angle}_1}(\vec{p}) = C_{\text{angle}_{(1,2)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{A}_{1,2}\vec{p} - 2\cos(2\pi/3))^2 = 0 \tag{57}$$

$$C_{\text{angle}_2}(\vec{p}) = C_{\text{angle}_{(1,3)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{A}_{1,3}\vec{p} - 2\cos(\pi/2))^2 = 0 \tag{58}$$

$$C_{\text{angle}_3}(\vec{p}) = C_{\text{angle}_{(2,3)}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathcal{A}_{2,3}\vec{p} - 2\cos(\pi/2))^2 = 0. \tag{59}$$

The application of the paradigm developed in Section 4.1 (representation of the objective function) is straightforward for this object and we get easily the following optimization



(a)

(b)

(c)

(d)

Fig. 10. Variation of the orientation error in the estimation of ($\vec{n}_1$) at the four steps of the algorithm *optim*2.

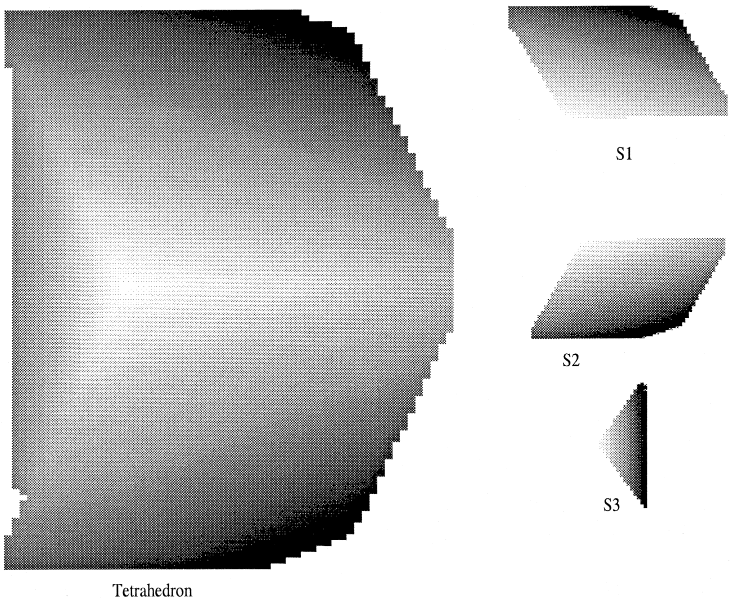
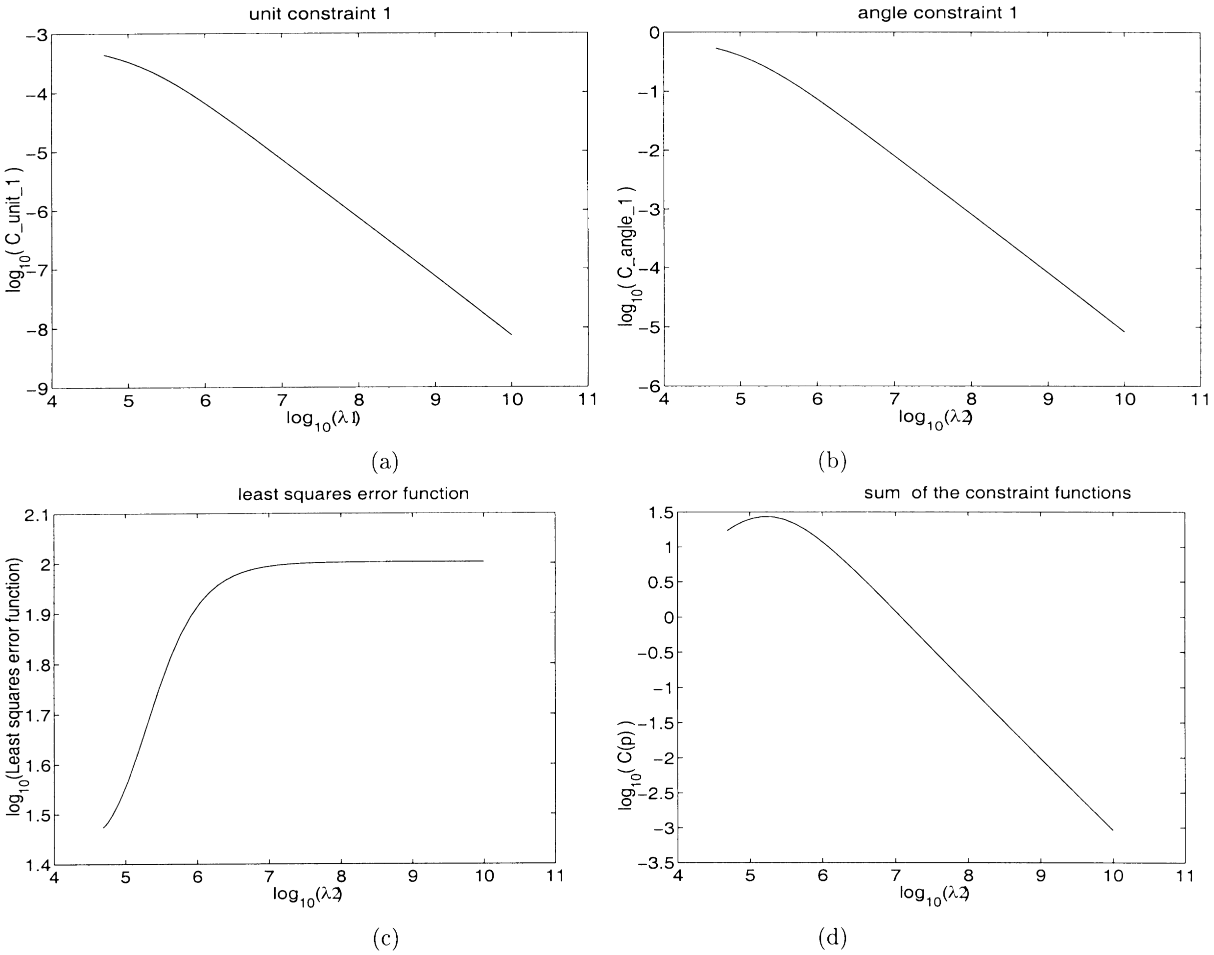


Fig. 11. A top view of the tetrahedron and the extracted surfaces.



(a)

(b)

(c)

(d)

Fig. 12. (a) Decrease of the unit constraint function (54) with respect to λ_1. (b) Decrease of the angle constraint function (57) with respect to λ_2. (c) and (d) variation of the objective function $\vec{p}^{\mathrm{T}}\mathscr{H}\vec{p}$ and the sum of all the constraint functions $C(\vec{p}) = \sum_{l=1}^{3} \mathrm{Unit}_l(\vec{p}) + \sum_{l=2}^{4} \mathrm{Angle}_{l-1}(\vec{p})$ during the optimization. These functions are mapped in function of λ_2 just to show their evolution all along the optimization process but they do not depend specifically on λ_2.

function:

$$\vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} + \lambda_1 \sum_{l=1}^{3} \mathrm{Unit}_1(\vec{p}) + \sum_{l=2}^{4} \lambda_l C_{\mathrm{angle}_{l-1}}(\vec{p}) \tag{60}$$

where

$$\mathscr{H} = \begin{bmatrix} G_1 & (0)_4 & (0)_4 \\ (0)_4 & G_2 & (0)_4 \\ (0)_4 & (0)_4 & G_3 \end{bmatrix}$$

and G_k have the same structure as in Eq. (45). All the constraints were applied simultaneously according to algorithm *optim*1. The results are the average of 100 trials. At each trial the initial vector $\vec{p}^{[0]}$ is corrupted by a uniform deviation of scale 5%. These 100 trials are a quantitative criterion for testing the stability of the algorithm with respect to the perturbations in the initial value of the solution. Here again all the different constraints values decrease during the optimization. This is illustrated through the two examples shown in Fig. 12(a) and (b) where the unit constraint C_{unit_1} (54) and the angle constraints C_{angle_1} (57) are mapped in function of their associated weighting values λ_1 and λ_2. Fig. 12(c) represents the variation of the objective function (the least squares function) $\vec{p}^{\mathrm{T}}\mathscr{H}\vec{p}$ during the optimization process; it increases slightly then it stabilizes. Fig. 12(d) shows the evolution of the sum of all the constraints during the algorithm application. Since at each iteration of the algorithm *optim*1 the λ_k values increase, the variation of the objective function and the sum of the constraints during the optimization is mapped in function of one of the λ_k (λ_2).

It is seen that the sum of the constraint values converges to zero at the end of the optimization. It is also noticed that the constraint values could be decreased further while the least squares error remains stable. Thus, the final part shape now satisfies the shape constraints at a slight increase in the least squares fitting error.

5.4. Application to surfaces having quadric surfaces

Compared to polyhedral objects this category has more complex constraints since the objects contain different types

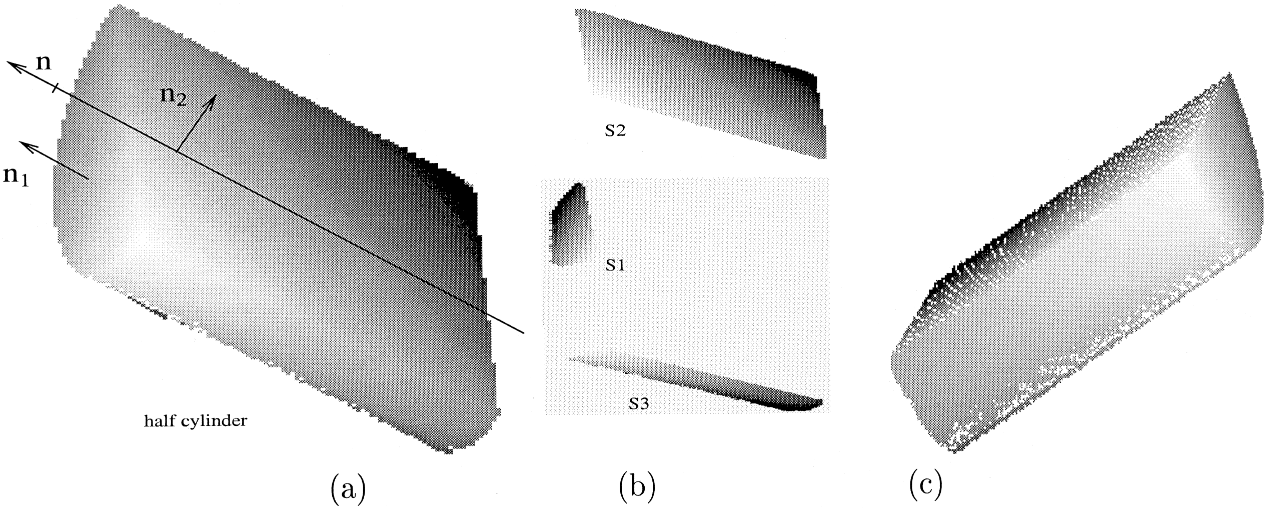


Fig. 13. Two views of the half cylinder and the extracted surfaces.

of surfaces and consequently more geometric features. So, besides the constraints related to the plane surfaces other constraints defining properties and relationships between quadric features could be defined as well as relationships between quadric features and plane features. The objects studied in this section contain cylindrical, conical and spherical patches. In this section, the constraints' expressions will use the notation of Appendix A.

Also, for all objects, the results of the proposed approach have been compared with object estimation methods which do not consider constraints, in particular the least squares technique applied to each surface separately.

5.5. The half cylinder

This object is composed of four surfaces. Three patches S_1, S_2 and S_3 have been extracted from two views represented in Fig. 13(a) and (c). These surfaces correspond respectively to the base plane S_2, lateral plane S_1 and the cylindrical surface S_3 (Fig. 13(b)). The parameter vector is $\vec{p} = [\vec{p}_1^{\mathrm{T}}, \vec{p}_2^{\mathrm{T}}, \vec{p}_3^{\mathrm{T}}]^{\mathrm{T}}$, where $\vec{p}_1 = [\vec{n}_1^{\mathrm{T}}, d_1]^{\mathrm{T}}$, $\vec{p}_2 = [\vec{n}_2^{\mathrm{T}}, d_2]^{\mathrm{T}}$ and $\vec{p}_3 = [a, b, c, h, g, f, u, v, w, d]^{\mathrm{T}}$. The least squares error function is given by:

$$F(\vec{p}) = \vec{p}^{\mathrm{T}} H \vec{p}, \qquad H = \begin{bmatrix} H_1 & O_{(4,4)} & O_{(4,10)} \\ O_{(4,4)} & H_2 & O_{(4,10)} \\ O_{(4,10)}^{\mathrm{T}} & O_{(4,10)}^{\mathrm{T}} & H_3 \end{bmatrix} \tag{61}$$

where H_1, H_2 and H_3 are the data matrices related respectively to S_1, S_2 and S_3. This object has the following constraints:

1. S_1 and S_2 are perpendicular.
2. The cylinder axis is parallel to S_1's normal.
3. The cylinder axis lies on the surface S_2.
4. The cylinder is circular.

Constraint 1 is expressed by the following condition:

$$C_{\mathrm{ang}}(\vec{p}) = (\vec{n}_1^{\mathrm{T}} \vec{n}_2)^2 = (\vec{p}^{\mathrm{T}} L_{(1,5,2)} \vec{p})^2 = 0.$$

Constraint 2 is satisfied by equating the unit vector $\vec{n}$ in (14) to S_1's normal $\vec{n}_1$. Constraints 3 and 4 are represented, respectively, by:

$$C_{\mathrm{axe}}(\vec{p}) = (\vec{i}_8^{\mathrm{T}} \vec{p} - \vec{p}^{\mathrm{T}} L_{(5,15,2)} \vec{p})^2 = 0$$

$$C_{\mathrm{circ}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_k}(\vec{p}) = 0.$$

See Appendix B for details.

Finally the normals $\vec{n}_1$ and $\vec{n}_2$ have to be unit. This is represented by:

$$C_{\mathrm{unit}}(\vec{p}) = (\vec{p}^{\mathrm{T}} U_{(1,3)} \vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}} U_{(5,7)} \vec{p} - 1)^2 = 0.$$

Thus, the optimisation function is:

$$E(\vec{p}) = \vec{p}^{\mathrm{T}} H \vec{p} + \lambda_1 C_{\mathrm{unit}}(\vec{p}) + \lambda_2 C_{\mathrm{ang}}(\vec{p}) + \lambda_3 C_{\mathrm{axe}}(\vec{p}) + \lambda_4 C_{\mathrm{circ}}(\vec{p}).$$

5.5.1. Experiments

In the first test, algorithm *optim*1 has been applied to data extracted from a single view (Fig. 13(c)). In Fig. 14 the behaviour of the different constraints during the optimization have been mapped as a function of the associated λ_k as well as the least squares residual (61) and the sum of the constraint functions. The figures show a nearly linear logarithmic decrease of the constraints. It is also noticed that at the end of the optimization all the constraints are highly satisfied. The least squares error converges to a stable value and the constraint function vanishes at the end of the optimization. The figures also show that it is possible to continue the optimization further until a higher tolerance

Table 2
Improvement in shape and placement parameters with and without constraints from data from single view of the half cylinder object

View2	Angle (S_1,S_2) (degree)	Distance (X_o,S_2) (mm)	Radius (mm)
Without constraints	90.84	6.32	26.98
With constraints	90.00[a]	0.00[a]	29.68
Actual values	90	0	30

[a] Means that the estimated value has been constrained to be the true value.

Table 3
Improvement in shape and placement parameters with and without constraints from data merged from two views of the half cylinder object

Registered view1 and view2	Angle (S_1,S_2) (degree)	Distance (X_o,S_2) (mm)	Radius (mm)
Without constraints	89.28	2.23	30.81
With constraints	90.00[a]	0.00[a]	30.06
Actual values	90	0	30

[a] Means that the estimated value has been constrained to be the true value.

is reached, however this is limited by the numerical accuracy of the machine.

In the second test, registered data from view 1 (Fig. 13(a)) and view 2 (Fig. 13(c)) was used. The registration was carried out by hand. Results similar to the first test were obtained for the constraints.

Tables 2 and 3 present the values of some object characteristics obtained from an estimation without considering the constraints and from the presented optimization algorithm. These are shown for the first and second test respectively.

The characteristics examined are the angle between plane S_1 and plane S_2, the distance between the cylinder axis's point X_0 (see Section 2.4 (the cylinder) (14)) and the plane S_2 and the radius of the cylinder. The comparison of the tables' values for the two approaches show the clear improvement made by the proposed technique. This is noticed in particular for the radius for which the actual value is 30 mm, although the extracted surface covers considerably less than a half of a cylinder. As we constrained the angle and distance relations, we expect these to be satisfied and they are to almost an arbitrarily high tolerance, as seen in Fig. 14. The radius was not constrained but the other constraints on the cylinder have allowed the least squares fitting of the unconstrained parameters to achieve a much more accurate estimation of the cylinder radius in both cases.

5.6. Multi-quadric objects

The third series of tests have been carried out on more complicated objects with several quadric surface patches. For these objects, all of the surfaces have been considered. The registration of the different views was done manually, thus the registered is expected to be corrupted by an additionally systematic error. By this way we can judge the performances of the algorithm in the presence of such errors.

5.7. Multi-quadric object 1

This object (Fig. 15) comprises two lateral planes S_1 and S_2, a back plane S_3, a bottom plane S_4, a cylindrical surface S_5 and a conic surface S_6. The cylindrical patch is less than a half cylinder (40% arc), the conic patch occupies a small area of the whole cone (less than 30%).

The vector parameter for this object is $\vec{p}^{\mathrm{T}} = [\vec{p}_1^{\mathrm{T}}, \vec{p}_2^{\mathrm{T}}, \vec{p}_3^{\mathrm{T}}, \vec{p}_4^{\mathrm{T}}, \vec{p}_5^{\mathrm{T}}, \vec{p}_6^{\mathrm{T}}]$ where $\vec{p}_i$ is the parameter vector associated to the surface S_i.

The surfaces of the object have the following constraints:

1. S_1 makes an angle of 120°[1] with S_2.
2. S_1 and S_2 are perpendicular to S_3.
3. S_1 and S_2 make an angle of 120° with S_4.
4. S_3 is perpendicular to S_4.
5. The axis of the cylindrical patch S_5 is parallel to S_3's normal.
6. The axis of the cone patch S_6 is parallel to S_4's normal.
7. The cylindrical patch is circular.
8. The cone patch is circular.

The first four angle constraints are grouped into a single angle constraint function:

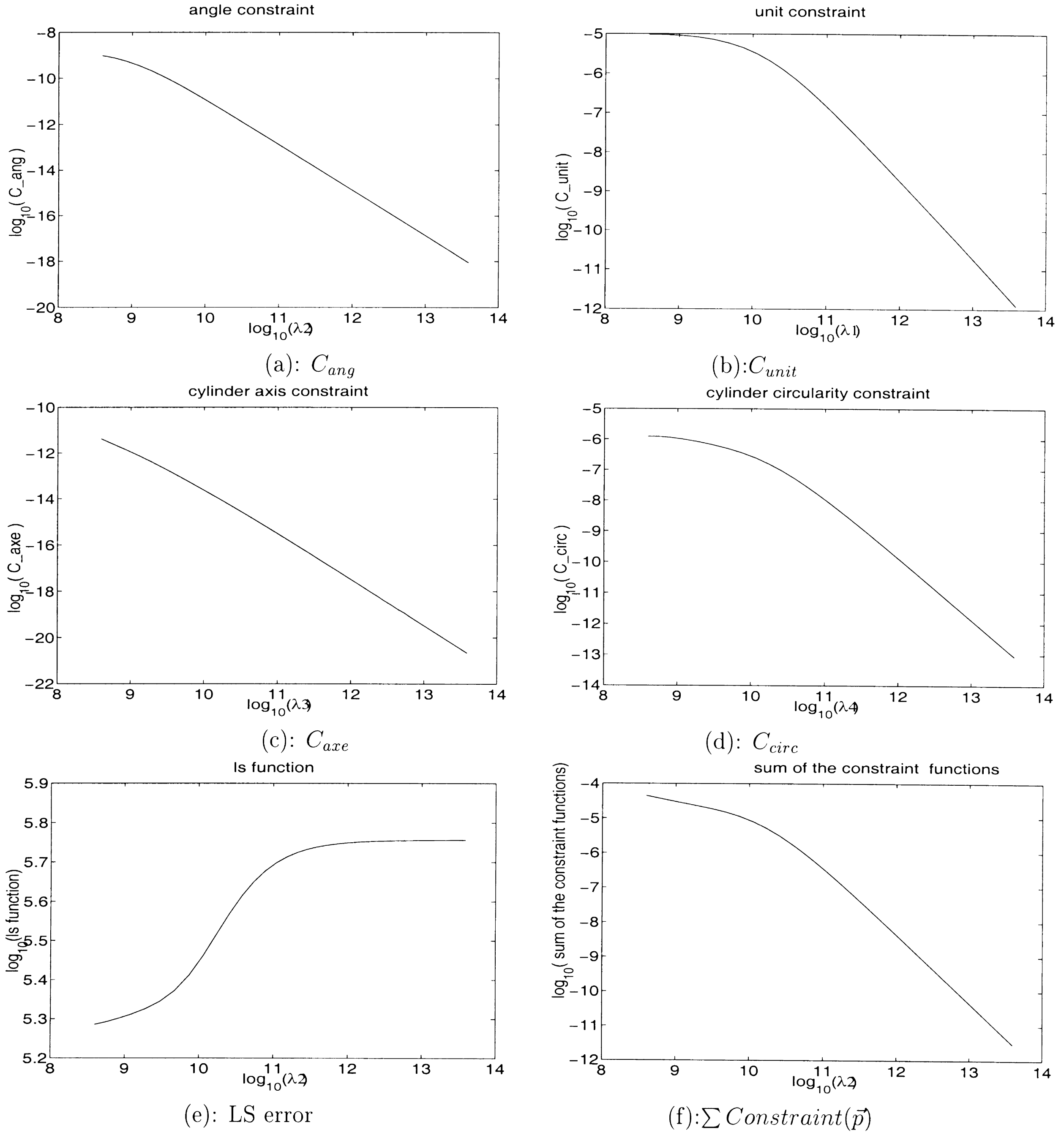
$$C_{\mathrm{angl}}(\vec{p}) = \sum_{i=1}^{4} C_{\mathrm{angl}_i}(\vec{p}).$$

Constraints 5 and 6 are imposed by associating the normals $\vec{n}_3$ and $\vec{n}_4$, respectively, to the unit vectors of the cylinder axis and the cone axis (see paragraphs circular cylinder and circular cone in Section 2 (preliminaries).

The circularity of the cylinder and the cone are expressed, respectively, by:

$$C_{\mathrm{circ}_{\mathrm{cyl}}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_{\mathrm{cyl}_k}}(\vec{p})$$

[1] We consider the angle between normals.



(a): C_{ang}

(b):C_{unit}

(c): C_{axe}

(d): C_{circ}

(e): LS error

(f):$\sum Constraint(\vec{p})$

Fig. 14. Shape optimization of the half cylinder object. (a)–(d): Decrease of the different constraints with respect to the related λ; (e), (f): variation of least squares function and the constraint function.

$$C_{\mathrm{circ}_{\mathrm{cone}}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_{\mathrm{cone}_k}}(\vec{p}).$$

See Appendix B for the development of all these constraints.

Finally the unit constraints on the surface normals have to be taken into account. This leads to the following unit constraint function:

$$C_{\mathrm{unit}}(\vec{p}) = (\vec{p}^{\mathrm{T}} U_{(1,3)} \vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}} U_{(5,7)} \vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}} U_{(9,11)} \vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}} U_{(13,15)} \vec{p} - 1)^2.$$

The complete optimisation function is then given by the expression:

$$E(\vec{p}) = \vec{p}^{\mathrm{T}} H \vec{p} + \lambda_1 C_{\mathrm{unit}}(\vec{p}) + \lambda_2 C_{\mathrm{ang}}(\vec{p}) + \lambda_3 C_{\mathrm{circ}_{\mathrm{cyl}}}(\vec{p}) + \lambda_4 C_{\mathrm{circ}_{\mathrm{cone}}}(\vec{p}).$$

5.7.1. Experiments

Since the surfaces cannot be recovered from a single view, four views (Fig. 15) have been registered by hand.

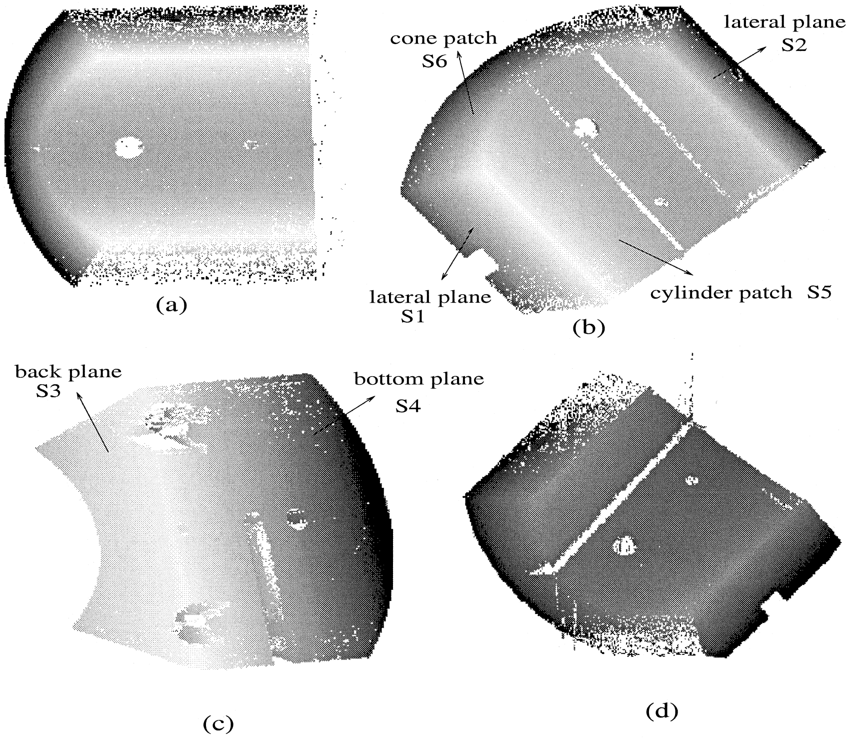


Fig. 15. Four views of the multi-quadric object 1.

Table 4
The surface's relative angle estimation with and without constraints

Angle	(S_1,S_2)	(S_1,S_3)	(S_1,S_4)	(S_2,S_3)	(S_2,S_4)	(S_3,S_4)
Without constraints	119.76	92.08	121.01	87.45	119.20	90.39
With constraints	120.00[a]	90.00[a]	120.00[a]	90.00[a]	120.00[a]	90.00[a]
Actual values	120	90	120	90	120	90

[a] Means that the estimated value has been constrained to be the true value.

Table 5
The cylinder characteristic estimates with and without constraints

Cylinder parameters	Angle (axis, S_3's normal)	Radius	Standard deviation of radius
Without constraints	2.34	37.81	0.63
With constraints	0.00[a]	59.65	0.08
Actual values	0	60	0

[a] Means that the estimated value has been constrained to be the true value.

One hundred estimations were carried out. At each trial 50% of the surface's points are selected randomly. Thus the algorithm starts with a different initialization each time. The results shown in this section are the average of these estimations. So by examining the mean of the estimations we can have a judgement on the algorithm convergence.

The results regarding the algorithm convergence are shown in Fig. 16. All of the constraint functions vanish and are highly satisfied.

The angles between the different fitted planes are presented in Table 4. It should be noticed that all the angles converge to the actual values. Tables 5 and 6 contain the estimated values of some attributes of the cylinder and the cone. The values show that each of the axis constraints are perfectly satisfied, the estimated radius and the cone half angle α improve when the constraints are introduced. We notice the good shape improvement, relative to the unconstrained least squares method, given by a reduction of bias of about 22 mm and 3°, respectively, in the radius and the half angle estimation. The standard deviation of the estimations have been reduced as well.

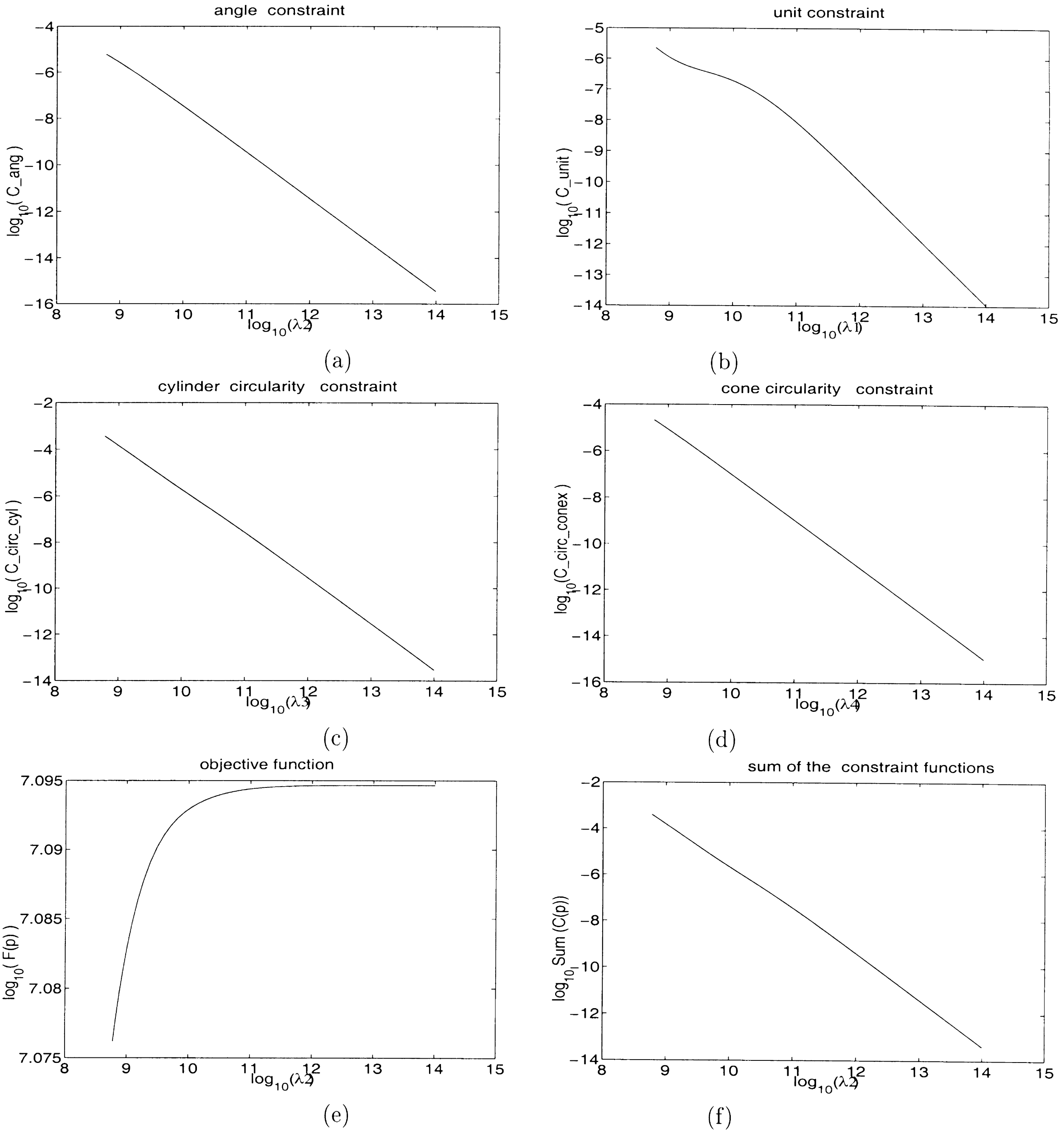


Fig. 16. Shape optimization of multi-quadric object 1. (a)–(d): Decrease of the different constraint functions with respect to the associated λ_k. (e),(f): variation of least squares error and the sum of all the constraint values during the optimization.

The radius estimation is within the hoped tolerances, a systematic error of about 0.5 mm is quite nice. However the cone half angle estimation involves a larger systematic error (about 1.8°). Two factors may contribute to this. The registration error may be too large since the registration was done by hand and the limited area of the cone patch which covers less then 30% of the whole cone. It is known that when a quadric patch does not contain enough information concerning the curvature, the estimation is very biased, even when robust techniques are applied, because it is not possible to predict the variation of the surface curvature.

5.7.2. Leaving some features unconstrained

We have also investigated whether leaving some of the features unconstrained affects the estimation since one can worry that the satisfaction of the other constraints may push the unconstrained surfaces away from their actual positions. To investigate this, we have left the angles between the pair of planes (S_1,S_2) and (S_1,S_3) unconstrained. The results are

Table 6
The cone characteristic estimates with and without constraints

Cone attributes	Angle (axis, S_4's normal)	α	Standard deviation of α
Without constraints	6.08	26.01	0.30
With constraints	0.00[a]	31.83	0.13
Actual values	0	30	0

[a] Means that the estimated value has been constrained to be the true value.

Table 7
Improvement of non-constrained angle estimates

Angle	(S_1,S_2)	(S_1,S_3)	(S_1,S_4)	(S_2,S_3)	(S_2,S_4)	(S_3,S_4)
Without constraints	119.76	92.08	121.48	87.45	119.20	90.39
With constraints	119.99	90.33	120.00[a]	90.00[a]	120.00[a]	90.00[a]
Actual values	120	90	120	90	120	90

[a] Means that the estimated value has been constrained to be the true value.

Table 8
Mean estimates of S_1 and S_3 normal and LS error in the two types of solutions

	$\vec{n}_i$	$\vec{n}_1$	$\vec{n}_3$	Angle ($\vec{n}_1$, $\vec{n}_3$) (degree)	LS error
1st case	0.5316 0.6733 0.5139	–	–	–	9.07
2nd case	–	0.5316 0.6733 0.5139	0.5316 0.6733 0.5139	0.00	9.06

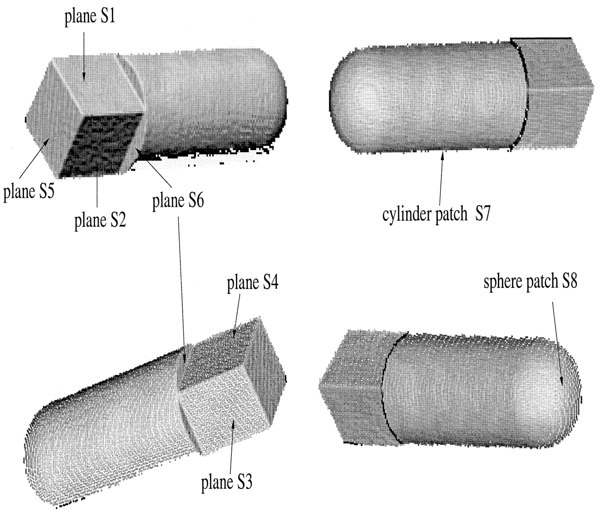


Fig. 17. Four views of the multi-quadric object 2.

shown in Table 7. We see that the estimated unconstrained angles are still close to the actual ones and the accuracy is improved compared to the non-constrained method.

5.8. Multi-quadric object 2

This object (Fig. 17) contains six plane surfaces $S_1, S_2, S_3, S_4, S_5, S_6$, a cylindrical surface S_7 and a spherical surface S_8. The surfaces S_1, S_2, S_3, S_4, S_5 form a square prism, the surface S_5 is a square plane surface.

The cylindrical patch is a whole cylinder and the spherical patch occupies a half sphere.

The surfaces of the object have the following relationships:

1. S_1, S_3 are parallel.
2. S_2, S_4 are parallel.
3. S_5, S_6 are parallel.
4. S_1, S_3 are orthogonal to S_2, S_4.
5. S_5, S_6 are orthogonal to S_1, S_3 and S_2, S_4.
6. S_1, S_3 and S_2, S_4 are separated by the same distance.
7. The cylinder axis is parallel to S_1, S_2, S_3 and S_4 and orthogonal to S_5, S_6.
8. The cylinder axis is located midway between S_1 and S_3 and midway between S_2 and S_4.
9. The cylindrical patch is circular.
10. The sphere centre lies on the cylinder axis.
11. The radius of the cylinder is equal to the radius of sphere.
12. The length diagonal of surface S_5 is equal to the cylinder diameter.

The constraints 1, 2 and 3 allow a single normal to be associated with each of the pair of planes (S_1,S_3), (S_2,S_4) and (S_5,S_6). Consequently the parameter vector of the object could be defined as:

$$\vec{p} = [\vec{n}_1^{\mathrm{T}}, d_1, d_3, \vec{n}_2^{\mathrm{T}}, d_2, d_4, \vec{n}_5^{\mathrm{T}}, d_5, d_6, \vec{p}_7^{\mathrm{T}}, \vec{p}_8^{\mathrm{T}}]^{\mathrm{T}}$$

where $\vec{n}_1$ is the normal associated to the pair of planes (S_1,S_3), d_1 is the parameter distance of S_1, d_3 is the parameter distance of $S_3, \vec{n}_2$ is the normal associated to the pair of planes (S_2,S_4), d_2 is the parameter distance of S_2, d_4 is the parameter distance of $S_4, \vec{n}_5$ is the normal associated to the pair of planes (S_5,S_6), d_5 is the parameter distance of S_5, d_6 is the parameter distance of $S_6, \vec{p}_7$ is the parameter vector associated to the cylindrical patch S_7 and $\vec{p}_8$ is the parameter vector associated to the spherical patch S_8.

The constraints 4 and 5 are expressed by:

$$C_{\mathrm{angl}}(\vec{p}) = \sum_{i=1}^{3} C_{\mathrm{angl}_i}(\vec{p}).$$

The 6th constraint is formulated by:

$$C_{\mathrm{dist}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(4,5,9,10)}\vec{p})^2 = 0.$$

The 7th constraint is imposed by associating the normal $\vec{n}_5$ to the unit vector of the cylinder axis (see paragraph circular cylinder in Section 2 (preliminaries)).

The constraints 8–12 are expressed, respectively, by:

$$C_{\text{axe\_pos}}(\vec{p}) = (-2\vec{p}^{\mathrm{T}}L_{(1,22,2)}\vec{p} + i^{\mathrm{T}}_{(4,-5)}\vec{p})^2 + (-2\vec{p}^{\mathrm{T}}L_{(6,22,2)}\vec{p} + i^{\mathrm{T}}_{(9,-10)}\vec{p})^2 = 0$$

$$C_{\text{circ}}(\vec{p}) = \sum_{k=1}^{6} C_{\text{circ}_k}(\vec{p}) = 0$$

$$C_{\text{sph}_{\text{center}}}(\vec{p}) = (\vec{p}^{\mathrm{T}}\mathrm{T}_{(11,12,22,23)}\vec{p})^2 + (\vec{p}^{\mathrm{T}}T_{(11,13,22,24)}\vec{p})^2 + (\vec{p}^{\mathrm{T}}T_{(12,13,23,24)}\vec{p})^2 = 0$$

$$C_{\text{equ}_{\text{radius}}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(25,30)}\vec{p} + \vec{p}^{\mathrm{T}}U_{(27,29,22,24)}\vec{p})^2 = 0$$

$$C_{\text{median}}(\vec{p}) = (\vec{p}^{\mathrm{T}}(I_{(4,1)} - 2U_{(22,24)})\vec{p} + 2\vec{i}^{\mathrm{T}}_{25}\vec{p})^2 = 0.$$

Finally the unit constraints related to the planes' normals and the unit constraint of the coefficient a of the sphere are grouped into the following unit constraint:

$$C_{\text{unit}}(\vec{p}) = (\vec{p}^{\mathrm{T}}U_{(1,3)}\vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}}U_{(6,8)}\vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}}U_{(11,13)}\vec{p} - 1)^2 + (\vec{p}^{\mathrm{T}}U_{(26,26)}\vec{p} - 1)^2.$$

The optimization function is then:

$$E(\vec{p}) = \vec{p}^{\mathrm{T}}H\vec{p} + \lambda_1 C_{\text{unit}}(\vec{p}) + \lambda_2 C_{\text{angl}}(\vec{p}) + \lambda_3 C_{\text{dist}}(\vec{p}) + \lambda_4 C_{\text{axe\_pos}}(\vec{p}) + \lambda_5 C_{\text{circ}}(\vec{p}) + \lambda_6 C_{\text{sph}_{\text{center}}}(\vec{p}) + \lambda_7 C_{\text{equ}_{\text{radius}}}(\vec{p}) + \lambda_8 C_{\text{median}}(\vec{p}).$$

The details concerning the formulation of all the above constraints are in Appendix B.

5.8.1. Experiments

The surfaces of the objects were recovered from four views shown in Fig. 17 and the registration of the range data was done by hand. Similarly to the previous object 100 trials were performed. At each of them 50% of the surfaces's points are selected randomly leading to a different initialisation each trial. In all the trials, the decrease of all the constraint errors and the high level of satisfaction of the constraints at the end of the optimization for a slight increase of the least squares error is essentially similar to that observed in the previous experiments and so similar graphs are not shown here.

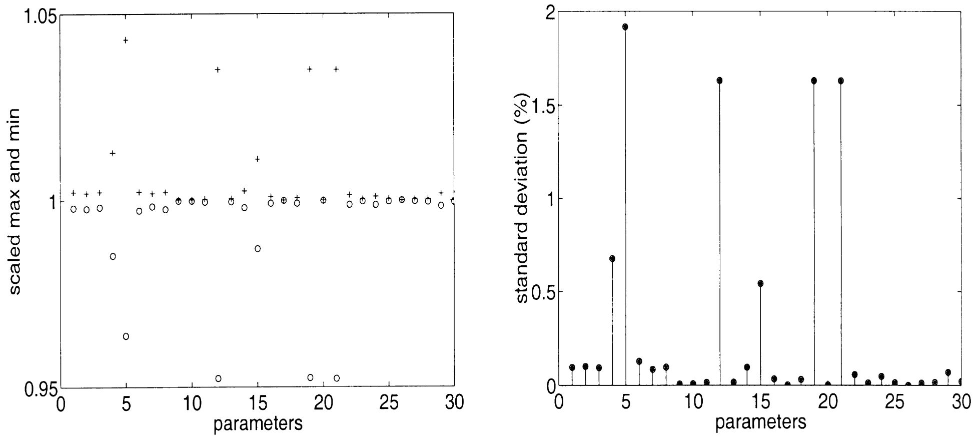
Through these different tests and trials we have been investigating:

1. How stable is the convergence of the algorithm?
2. How close is the estimation to the actual optimal value?
3. What are the effects of leaving some features unconstrained?
4. What is the effect of constraint invalidity?
5. What is the effect of constraint inconsistency?

Lastly, some results concerning the global shape improvement of the object model will be presented.

5.8.2. Stability of the convergence

The previous experiments performed each over 100 trials have shown that the mean of the estimated shapes obtained form these trials converges close to the actual solution which satisfies the constraints. The initial solution in each trial has a different value since the data points are selected randomly. This experiment aims to check the sensitivity of the algorithm with respect to the initial value, to test the stability of the convergence of the algorithm with respect to changes in the initial estimation. One way is to do so is to compute the difference between the maximum and the minimum value of each parameter in the set of different solutions. A second way is to examine statically the "closeness"



(a)

(b)

Fig. 18. (a) Maximum (+) and minimum (○) value for each parameter scaled by the absolute value of the mean. (b) Relative standard deviation of the parameters.

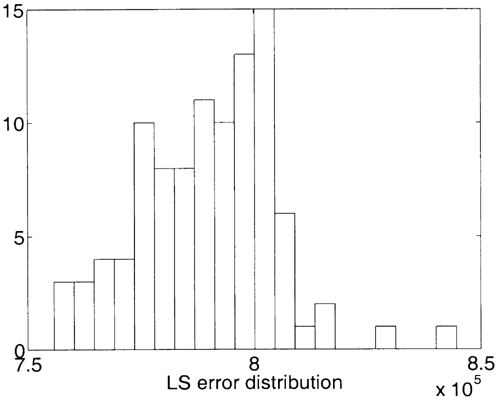


Fig. 19. Distribution of the least squares errors.

of the different estimates to the mean solution, known in the statistic terminology as the distribution of the solutions. This distribution could be obtained by computing the variance of each parameter from the solutions issued from the 100 trials. Fig. 18(a) shows the maximum and the minimum value (scaled by the absolute value of the mean) for each parameter. The extrema of the different parameters vary at a very low scale around the mean solution, in a range lower than 2%. The same is noticed in the standard deviations of the parameters illustrated in Fig. 18(b). This aspect is further confirmed in the distribution of the least squares errors of the different estimations shown in Fig. 19. The related relative variance is 1.94%.

5.8.3. Closeness to the actual optimal solution

By actual optimal solution we mean the estimate obtained from a process where the constraints are implicitly built into the least squares model. The solution provided in this case always completely satisfies the constraints. So one may ask how close is the estimate issued from our approach to this optimal solution. As we have mentioned previously, such an ideal and elegant formulation is difficult or impossible to achieve for many objects due to the complexity and to the nonlinearity of the geometric constraints. In fact one purpose and motivation of our approach is to overcome this problem. Nevertheless it is possible for some simple particular cases to combine the constraints with the least squares error.

So, in order to make a comparison with the optimal solution a sub-part of the multi-quadric object 2 was considered. It is composed of the two parallel planes S_1 and S_3. The objective is to estimate the planes' orientation taking into account the parallel constraints. For the first case, the parallel constraint is implicitly considered by associating one normal to both planes. The optimization function is then:

$$\vec{n}^{\mathrm{T}}H\vec{n} + \lambda(1 - \vec{n}^{\mathrm{T}}\vec{n})$$

where H is the appropriate data matrix. The second term of the function is the unit constraint. A closed form solution is provided by the eigenvalue method.

In the second case each plane was assigned a different normal vector. The equality of the two normals has to be satisfied through the optimization process. According to our approach the objective function is:

$$\vec{n}_1^{\mathrm{T}}H_1\vec{n}_1 + \vec{n}_3^{\mathrm{T}}H_3\vec{n}_3 + \lambda_1(1 - \vec{n}_1^{\mathrm{T}}\vec{n}_1)^2$$
$$+ \lambda_2(1 - \vec{n}_3^{\mathrm{T}}\vec{n}_3)^2 + \lambda_3(1 - \vec{n}_1^{\mathrm{T}}\vec{n}_3)^2.$$

One hundred tests were applied for each of the two cases. The average of the results are summarized in Table 8. The estimations are similar in the two cases. This shows that both solutions converge to the same value and almost equally minimize the least squares error. The LS of the second solution is slightly lower than the optimal solution one. This is because in the optimal case the constraint is perfectly satisfied so the least squares error has to absorb all the error. The same convergence of the two solutions is further confirmed from the distribution of the angle $(\vec{n}, \vec{n}_c)$,

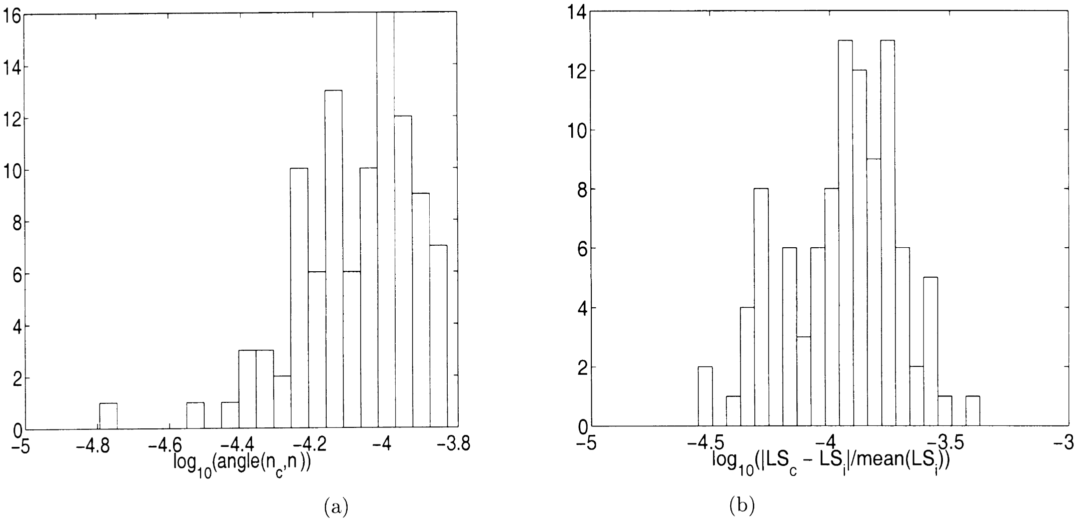


Fig. 20. (a) Distribution of the estimation difference. (b) Distribution of the LS residuals difference.

Table 9
Comparison of the estimation without median constraints with previous results

	Distance (S_1,S_3)	Distance (S_2,S_4)	Diagonal of S_5	Cylinder radius
Without constraints	–	–	–	14.64
With all constraints	21.17	21.17	29.94	14.97
Without median constraint	21.15	21.15	29.91	14.97
Actual values	21.28	21.28	30.02	15.01

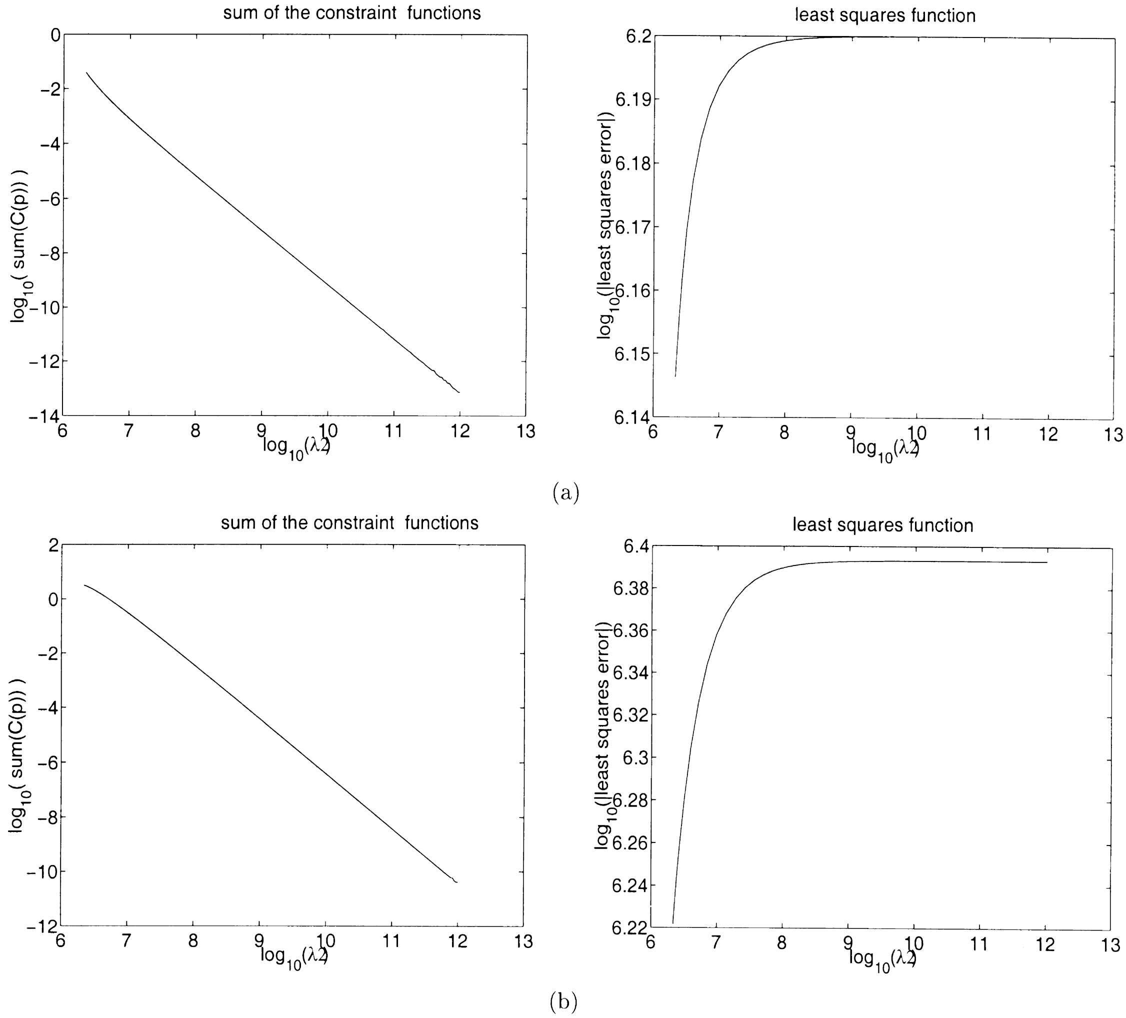
where $\vec{n}_c$ is the mean of $\vec{n}_1$ and $\vec{n}_3$, and the distribution of the difference between the LS error related to each of them, LS and LS$_i$ (Fig. 20). These distributions are issued from 100 trials.

5.8.4. Leaving some features unconstrained

Another series of tests has been performed without considering the median constraint (constraint 12). This is in order to check if this will affect the position of the four plane surfaces with respect to the cylinder axis and therefore the estimation of the edge of the square surface S_5. Results are shown in Table 9 with the previous results for comparison. It is noticed that the radius estimation is not affected but the incorporation of the additional constraints slightly reduces the diagonal length error.

5.8.5. Invalidity of the constraints

Suppose that one or more constraints do not reflect the actual relationships between features and therefore are invalid. What would be the behaviour of the algorithm? Will these “false constraints” be satisfied? What could be the resulting estimated model?



(a)

(b)

Fig. 21. (a) Constraint error function and least squares error function for valid constraints. (b) Constraint error and least squares error function for invalid constraints (3rd test).

To answer these questions, some angle constraints were set to incorrect values. Three tests was carried out, in the first the angle $(\vec{n}_1, \vec{n}_2)$ was set to $\pi/3$, in the second the angle $(\vec{n}_1, \vec{n}_5)$ was set to $\pi/3$ and in the third test both angles $(\vec{n}_1, \vec{n}_5)$ and $(\vec{n}_2, \vec{n}_5)$ were set to $\pi/3$ (the right values are $\pi/2$ for both angles).

In all these tests the behaviour and the convergence of the algorithm were qualitatively similar to those of the previous experiments. The algorithm converges, the least squares error stabilizes and all the constraints are satisfied at the end of the process although the least squares error is greater than the valid constraints case (Fig. 21). Table 10 summarizes the estimated model characteristics in each of the three tests.

An examination of Table 10 leads to the following observations:

1. In all of the three tests the cylinder and the sphere characteristics are not affected by the invalid constraints.
2. The normal $\vec{n}_1$ which is involved in each of the invalid constraints is affected in three tests.
3. The normal $\vec{n}_2$ is changed in the first and third test where it is involved in the invalid constraints whereas it is unchanged in the second test where it is not involved.
4. The normal $\vec{n}_5$ is kept unchanged in all the tests even in those where it is involved in the inconsistent constraints.

From these observations we can deduce that invalid constraints affect the object feature's locations by shifting the involved features toward positions where the invalid constraints are satisfied. Consequently, this will increase the least squares error (Fig. 21). The locations and the characteristics of the surfaces which are not involved in the invalid constraints are not affected (the sphere and the cylinder). However the normal $\vec{n}_5$ seems not to satisfy this rule since its orientation stays unchanged for all the cases where it is involved in an inconsistent constraint. This is explained by the fact that unlike $\vec{n}_1$ and $\vec{n}_2$, $\vec{n}_5$ is also involved in other constraints, in particular it is constrained to have the same orientation as the cylinder axis. The satisfaction of this constraint keeps it collinear to the cylinder axis and prevents its orientation from being affected. Thus the algorithm satisfies the invalid constraints in which $\vec{n}_5$ is involved by acting on the other normals involved in these constraints.

5.8.6. Inconsistency of the constraints

In this test we investigated what would be the behaviour of the algorithm when some constraints are inconsistent and have a conflict between them. For this purpose we introduced two additional inconsistent angle constraints by imposing the angles $(\vec{n}_1, \vec{n}_2)$ and $(\vec{n}_1, \vec{n}_5)$ to be $\pi/3$, which conflicts with the two other consistent constraints defining each pair of $(\vec{n}_1, \vec{n}_2)$ and $(\vec{n}_1, \vec{n}_5)$ as orthogonal vectors. The trial carried out with these inconsistent constraints revealed that the algorithm converges normally (Fig. 22) both the least squares and the constraint functions stabilizing at the end of the algorithm. From Fig. 22(a) we notice that the angle constraints are not satisfied. This is obvious because it is not possible to satisfy conflicting constraints simultaneously. The converging value of the constraint function (the sum of all the constraints (Fig. 22(b)) and the angle constraints error are practically equal at the end of the optimization process. This shows that the other consistent constraints are satisfied. This suggests that an inconsistent set of constraints can be detected by observing the convergence of the constraint error rather than its reduction to zero.

5.8.7. Global shape improvement

The different tables shown in this section compare the geometric characteristics of the object issued from an optimization with constraints and an optimization without and show the improvement of the object characteristics estimates when constraints are applied. The results presented in the tables are the average of the 100 estimations. The angles between each pair of surfaces (S_1,S_2), (S_1,S_5) and (S_2,S_5) were set as constraints and the constraints were nearly perfectly satisfied. From Table 11 we notice the satisfaction of the square property of the prism, illustrated by the equality of the two distances separating (S_1,S_3) and (S_2,S_4),

Table 10
The object characteristic estimates for invalid constraints and true constraints (last row)

	$\vec{n}_1$	$\vec{n}_2$	$\vec{n}_5$	R_{cyl}	R_{sph}	Axe_{cyl}	$Centre_{\mathrm{sph}}$
$(\vec{n}_1, \vec{n}_2) = \pi/3$	− 0.61	− 0.58	0.72			0.72	86.30
	− 0.47	0.52	− 0.02	14.97	14.97	− 0.02	− 87.38
	− 0.62	− 0.62	− 0.69			− 0.69	17.44
$(\vec{n}_1, \vec{n}_5) = \pi/3$	− 0.08	− 0.46	0.72			0.72	86.31
	− 0.60	0.72	− 0.02	14.97	14.97	− 0.02	− 87.41
	− 0.78	− 0.50	− 0.69			− 0.69	17.44
$(\vec{n}_1, \vec{n}_5) = \pi/3$	− 0.02	0.05	0.72	14.97	14.97	0.72	86.31
$(\vec{n}_2, \vec{n}_5) = \pi/3$	− 0.68	0.72	− 0.02			− 0.02	− 87.42
	− 0.72	− 0.68	− 0.69			− 0.69	17.44
True constraints	− 0.52	− 0.45	0.72	14.97	14.97	0.72	86.30
	− 0.67	0.73	− 0.02			− 0.02	− 87.38
	− 0.51	− 0.50	− 0.69			− 0.69	17.44

Table 11
Improvement of the prism characteristic estimates

	Distance (S_1,S_3)	Distance (S_2,S_4)	Diagonal of S_5
With constraints	21.17	21.17	29.95
Standard deviation/mean	0.03%	0.03%	0.03%
Actual values	21.28	21.28	30.02

Table 12
Improvement of the cylinder characteristic estimates

Cylinder parameters	Angle (axis, S_5's normal)	Radius (mm)	σ/mean (radius)
Without constraints	1.55	14.64	0.12%
With constraints	0.00[a]	14.97	0.03%
Actual values	0	15.01	0

[a] Means that the estimated value has been constrained to be the true value.

Table 13
Improvement of the sphere characteristic estimates

Sphere parameters	Distance (centre, cylinder axis)	Radius (mm)	σ/mean (radius)
Without constraints	1.36	16.02	0.11%
With constraints	0.00[a]	14.97	0.03%
Actual values	0	15.01	0

[a] Means that the estimated value has been constrained to be the true value.

their values which is close to the actual length of the edge of the square plane S_5 and closeness of the estimated value of the diagonal of S_5 to the actual value when the constraints are considered. The distances between these last surfaces for an optimization without constraints is not mentioned in this table since the estimated surfaces are not parallel.

The improvement of the quadric surfaces estimation is confirmed again for this object (Tables 12 and 13). The radius estimation error is less than 0.04 mm for both the cylinder and the sphere. The standard deviations of the cylinder and the sphere radius have been significantly reduced as well.

6. Conclusion

This work presents a framework for the reconstruction of object models incorporating geometric constraints. It can hold a large number of varied constraint types and incorporates them integrally without the need for linearization. The geometric constraints are formulated quadratic matrix functions which are continuous, differentiable and ensure a compact expression of the constraints and easy handling by the optimization process.

The proposed optimization algorithm belongs to sequential nonlinear programming. Theoretically, the characteris-

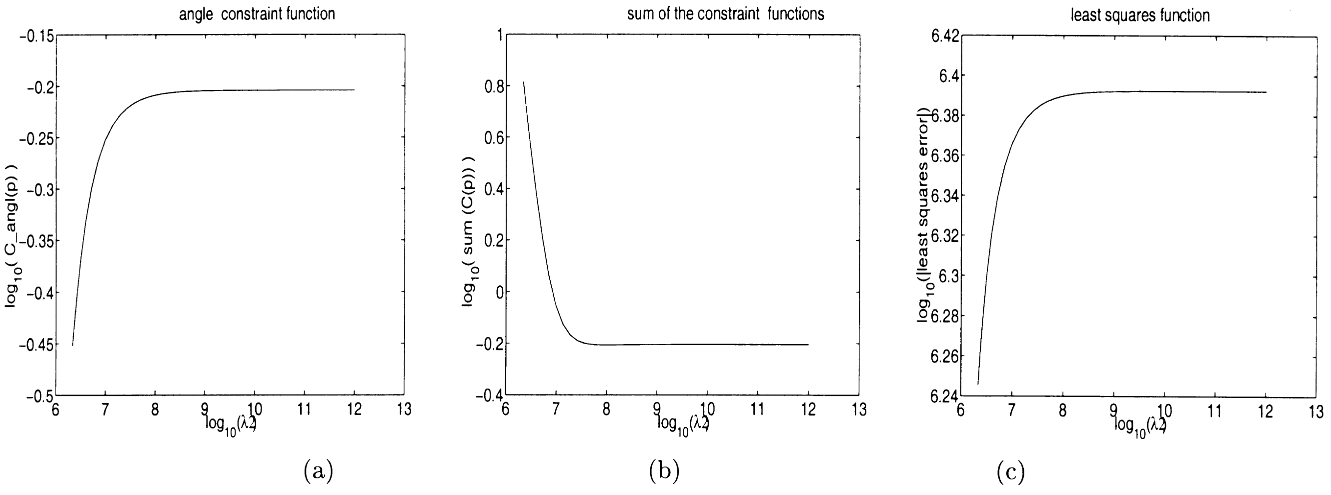


Fig. 22. Results for inconsistent constraints. (a) The sum of the angle constraints' error. (b) The sum of all the constraint functions. (c) The least squares error.

tics of the objective function and the constraint functions satisfy the requirements for an efficient application of the algorithm. The availability of a good initial solution obtained from the measurement data ensures the convergence of the algorithm towards the optimal solution. However, the last condition makes it inappropriate for constrained object design applications where a reasonable initial solution is not available. The practical difficulties of the algorithm manifested in the ill-conditioned Hessian matrix in the Levenberg–Marquardt algorithm is overcome by using an appropriate numerical technique.

The constraints can be integrated in a batch form at once or sequentially. In the sequential version the addition of a new constraint does not affect the satisfaction of the previously implemented constraints.

The experiments carried out on the different objects empirically confirm the convergence of the algorithm. The parameter optimization search does produce shape fitting that satisfies almost perfectly the constraints. They show in particular that the least squares error grows slightly as the constraints are applied and the weighting values increased, but this stabilizes above certain values of the λ_k while the constraint errors are still decreasing. Thus it is possible to satisfy the constraints up to the desired tolerance without seriously affecting the quality of the data fitting.

The above observations suggest that the proposed approach allows flexibility in the incorporation of the constraints, as well as in their satisfaction. The sequential version of the constraint implementation allows a human reverse engineer to supply them interactively whereas the batch form of constraint incorporation is suitable for being inferred by a knowledge-based system reasoning from general engineering principles. The stabilization of the increase of the least squares error while the constraint errors are still decreasing as λ_k increases offers the possibility that the user can control the degree of satisfaction of the constraints and to set the tolerances as high as necessary.

Regarding the slight increases of the LS error, we have to bear in mind that the increase of the least-squares residuals value may not reflect a bad estimation in the case when measurement errors are systematic, e.g. miscalibration and registration error. This last type of error is expected in our data since the registration process is performed by hand. We believe that the slight increase of the least squares error as a consequence of the constraints satisfaction is a result of the object being located more accurately. Future work could investigate a more robust form for the objective function involving the data noise statistics.

The different trials applied on the multi-quadric objects empirically confirm the stability of the convergence of the algorithm. The low values of the parameters' variances illustrates the stability of the solution provided by the optimization search process. On the level of the object shape, this aspect is reflected by the small values of the standard deviations of the object shape characteristics. The tests have shown as well that the proposed approach leads to an estimate which is close to the optimal solution (e.g. the solution given when the constraints could be combined with the least squares error). The experiments also show that applying the constraints to only some features does not seriously affect the estimation of the unconstrained surfaces. The estimation is still improved compared to the case of unconstrained optimization.

The examination of some constraint invalidity cases has shown that the constraints are always satisfied whether they are valid or not and the behaviour of the algorithm is typically the same. The satisfaction of invalid constraints leads to the relocation of the involved and less constrained features (having more degrees of freedom) toward positions where the inconsistency is removed. However this will result in a false object model. The trial performed with constraint inconsistencies case revealed the same behaviour regarding the convergence of the algorithm but the inconsistent constraints are not satisfied at the end of the optimization. This suggests that constraint validity and consistency checking have to be done before starting the optimization process.

Regarding the model estimation accuracy, the comparison of the object dimension estimates with those from unconstrained fitting confirms that the proposed approach improves the quality of the model construction to a high degree. For the second-quadric object the radius of the cylinder and the sphere have an estimation error in the range of 0.04 mm, the edge of the square prism has an estimation error around 0.1 mm. The radius of the cylinder patch estimated from the registered half cylinder has an estimation error around 0.01 mm. For a single a view it is less than 0.5 mm. The same range of error is obtained for the radius of the cylinder patch of the first multi-quadric object.

Results for the cone patches are reasonable for the cone object, the half angle estimation error is less than 0.5°, but less satisfactory for the first multi-quadric object. This is mainly due to the relatively small area of the conic patch. In fact, the comparison of the estimation error for the quadric surfaces shows that the larger the quadric patch, the smaller the estimation error. We intentionally chose to work with small patches because unconstrained fitting surface techniques fail to give reasonable estimates in this case (see the radius estimate in Table 5) even with robust algorithms due to the "poorness" of the information embodied in the patch.

Regarding the constraint representation, it is noticed that some constraints involve a large number of equations, in particular for the circularity constraint. One solution is to implicitly impose those constraints through the representation of the quadric equation $((X - X_0)^{\mathrm{T}}(I - \vec{n}\vec{n}^{\mathrm{T}})(X - X_0) - r^2 = 0)$ for the cylinder and $((X - X_0)^{\mathrm{T}}(\vec{n}\vec{n}^{\mathrm{T}} - \cos^2(\alpha))(X - X_0) = 0)$ for the cone, where $\vec{n}$ is the unit orientation vector of the cylinder or the cone axis, X_0 is a point on the cylinder axis in the cylinder case and is the apex for the cone case. The main problem encountered with this representation is the complexity of the related objective function and the difficulty of separating the data terms from the parameter terms. It will be also worthwhile investigating some topological constraints between surfaces which have a common intersection.

Although the experiments presented in this work were performed on single objects, the proposed approach can optimize multiple objects simultaneously. Generally industrial parts are designed to fit to each other, so geometric relationships between the parts may be considered and the resulting constraints can be incorporated as well in the optimization process.

Another area we are starting to investigate is how one might automatically identify inter-surface relationships that can have a constraint applied. In manufacturing objects, simple angular and spatial relationships are given by design. So, it should be straightforward to define simple Mahalanobis distance tests that hypothesize standard feature relationships, subject to the feature's statistical position distribution. With this analysis, a computer program could propose a variety of constraints that a human could either accept or reject, after which shape reconstruction could occur.

It is very likely that the consideration of the constraints tends to shift the object localization towards the actual position. The experiments carried out with the synthetic polyhedral objects provides evidence for this. It seems that the incorporation of the constraints compensate up to certain degree for the effect of the systematic errors and allows better estimation, although the authors have not yet a theoretical proof of this interpretation. This issue was partially justified in the work of Bolle et al. [44], but only for the intrinsic constraints, namely the circularity of the cylinder and perfect sphere. By considering a larger set of constraints, the proposed framework generalizes the concept of object localization considering the constraints and make a step toward a framework which unifies object localization and object modelling.

All the algorithm procedures have been implemented with C++. The computation time for the reconstruction on a 200 MHz Sun Ultrasparc workstation is typically few minutes or less (1–5 min), which is suitable for CAD work.

Acknowledgements

The work presented in this paper was funded by UK EPSRC grant GR/L25110.

Appendix A. Notation

$\vec{i}_r$ is a vector in which all the elements are zero except the rth element which is equal to 1.

$\vec{i}_{(r,s)}$ is a vector in which all the elements are zero except the rth and the sth elements which are equal to 1.

$\vec{i}_{(r,-s)}$ is a vector in which all the elements are zero except the rth and the sth elements which are equal to 1 and -1 respectively.

$\vec{i}_{(r,s,t,l)}$ is a vector in which all the elements are zero except for the rth, sth tth, and lth elements which are equal to 1, 1, -1 and -1, respectively.

$M_{(r,s)}$ is a diagonal matrix in which all the elements are zero except the rth and the sth elements which are equal to 1 and -1, respectively.

$U_{(r,s)}$ is a diagonal matrix defined by:

$$U_{(r,s)} = \begin{cases} U(i,i) = 1 & \text{if } r \le i \le s \\ U(i,i) = 0 & \text{otherwise} \end{cases}.$$

$I_{(r,s)}$ a symmetric matrix defined by:

$$I_{(r,s)} = \begin{cases} I(i,j) = I(j,i) = 1 & \text{for } r \le i \le s,\ \ r \le j \le s \\ I(i,j) = 0 & \text{otherwise} \end{cases}.$$

$U_{(r,s,p,t)}$ is a diagonal matrix defined by:

$$U_{(r,s,p,t)} = \begin{cases} U(i,i) = 1 & \text{if } r \le i \le s \\ U(i,i) = -1 & \text{if } p \le i \le t\,. \\ U(i,i) = 0 & \text{otherwise} \end{cases}$$

$L_{(r,s,p)}$ a symmetric matrix defined by:

$$L_{(r,s,p)} = \begin{cases} L(i,j) = L(j,i) = 1/2 & \text{for } r \le i \le r+p,\ \ s \le j \le s+p \\ L(i,j) = L(j,i) = 0 & \text{otherwise} \end{cases}.$$

$T_{(r,s,p,t)}$ a symmetric matrix defined by:

$$T_{(r,s,p,t)} = \begin{cases} T(r,t+5) = T(t+5,r) = T(s,p) = T(p,s) = 1/2 \\ T(r,t) = T(t,r) = T(s,p+5) = T(p+5,s) = -1/2. \end{cases}$$

Appendix B. Constraints definition

B.1. The half cylinder

Constraint 3 is represented by two conditions: axis vector $\vec{n}$ is orthogonal to S_2's normal $\vec{n}_2$, and one point of the axis satisfies S_2's equation. The first condition is guaranteed by constraint 2 since $\vec{n}_2$ is orthogonal to $\vec{n}_1$. For the second condition the point X_0 in Section 2.4 (the cylinder) has to satisfy the equation:

$$C_{\text{axe}}(\vec{p}) = (X_0^{\mathrm{T}}\vec{n}_2 + d_2)^2 = 0.$$

Using Eqs. (9) and (15) this equation can be written as

$$C_{\mathrm{axe}}(\vec{p}) = (-[u,v,w]^{\mathrm{T}}\vec{n}_2 + d_2)^2 = (\vec{i}_8^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}L_{(5,15,2)}\vec{p})^2 = 0.$$

The cylinder circularity constraint is implicitly defined by the Eq. (15). From these equations we extract the following constraints on the parameter vector $\vec{p}$:

$$C_{\mathrm{circ}_1}(\vec{p}) = (\vec{i}_9^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(1,1)}\vec{p} - 1)^2 = 0$$

$$C_{\mathrm{circ}_4}(\vec{p}) = (\vec{i}_{12}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(1,2,0)}\vec{p})^2 = 0$$

$$C_{\mathrm{circ}_2}(\vec{p}) = (\vec{i}_{10}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(2,2)}\vec{p} - 1)^2 = 0$$

$$C_{\mathrm{circ}_5}(\vec{p}) = (\vec{i}_{13}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(1,3,0)}\vec{p})^2 = 0$$

$$C_{\mathrm{circ}_3}(\vec{p}) = (\vec{i}_{11}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(3,3)}\vec{p} - 1)^2 = 0$$

$$C_{\mathrm{circ}_6}(\vec{p}) = (\vec{i}_{14}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(2,3,0)}\vec{p})^2 = 0.$$

We group these six constraints into a single one:

$$C_{\mathrm{circ}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_k}(\vec{p}) = 0.$$

B.2. Multi-quadric object 1

The first four angle constraints lead to six equations involving the surface normals:

$$\vec{n}_1^{\mathrm{T}}\vec{n}_2 = \cos(2\pi/3) = -0.5 \qquad \vec{n}_2^{\mathrm{T}}\vec{n}_3 = \cos(\pi/2) = 0$$

$$\vec{n}_1^{\mathrm{T}}\vec{n}_3 = \cos(\pi/2) = 0 \qquad \vec{n}_2^{\mathrm{T}}\vec{n}_4 = \cos(2\pi/3) = -0.5$$

$$\vec{n}_1^{\mathrm{T}}\vec{n}_4 = \cos(2\pi/3) = -0.5 \qquad \vec{n}_3^{\mathrm{T}}\vec{n}_4 = \cos(\pi/2) = 0.$$

A vector formulation of these equations as a function of $\vec{p}$ is:

$$C_{\mathrm{angl}_1}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(1,5,2)}\vec{p} + 0.5)^2 = 0$$

$$C_{\mathrm{angl}_4}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(5,9,2)}\vec{p})^2 = 0$$

$$C_{\mathrm{angl}_2}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(1,9,2)}\vec{p})^2 = 0$$

$$C_{\mathrm{angl}_5}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(5,13,2)}\vec{p} + 0.5)^2 = 0$$

$$C_{\mathrm{angl}_3}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(1,13,2)}\vec{p} + 0.5)^2 = 0$$

$$C_{\mathrm{angl}_6}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(9,13,2)}\vec{p})^2 = 0.$$

These equations are then grouped into

$$C_{\mathrm{angl}}(\vec{p}) = \sum_{i=1}^{6} C_{\mathrm{angl}_i}(\vec{p}) = 0.$$

The circularity of the cylinder and the cone are ensured using the set of Eqs. (15) and (19), respectively. This gives the following constraints on the parameter vector $\vec{p}$ for the cylinder:

$$C_{\mathrm{circ}_{\mathrm{cyl}_1}}(\vec{p}) = (\vec{i}_{17}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(9,9)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_4}}(\vec{p}) = (\vec{i}_{20}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(9,10,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_2}}(\vec{p}) = (\vec{i}_{18}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(10,10)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_5}}(\vec{p}) = (\vec{i}_{21}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(9,11,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_3}}(\vec{p}) = (\vec{i}_{18}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}U_{(11,11)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_6}}(\vec{p}) = (\vec{i}_{22}^{\mathrm{T}}\vec{p} + \vec{p}^{\mathrm{T}}L_{(10,11,0)}\vec{p})^2$$

and the following constraints for the cone:

$$C_{\mathrm{circ}_{\mathrm{cone}_1}}(\vec{p}) = (\vec{i}_{(27,-28)}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}M_{(13,14)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cone}_4}}(\vec{p}) = (\vec{i}_{30}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}L_{(13,14,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cone}_2}}(\vec{p}) = (\vec{i}_{(27,-29)}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}M_{(13,15)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cone}_5}}(\vec{p}) = (\vec{i}_{31}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}L_{(13,15,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cone}_3}}(\vec{p}) = (\vec{i}_{(28,-29)}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}M_{(14,15)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cone}_6}}(\vec{p}) = (\vec{i}_{32}^{\mathrm{T}}\vec{p} - \vec{p}^{\mathrm{T}}L_{(14,15,0)}\vec{p})^2.$$

The above sets are then grouped in two circular constraints, respectively:

$$C_{\mathrm{circ}_{\mathrm{cyl}}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_{\mathrm{cyl}_k}}(\vec{p}) = 0$$

$$C_{\mathrm{circ}_{\mathrm{cone}}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_{\mathrm{cone}_k}}(\vec{p}) = 0.$$

B.3. Multi-quadric object 2

The orthogonality constraints (4 and 5) between planes are formulated as:

$$\vec{n}_1^{\mathrm{T}}\vec{n}_2 = \vec{n}_1^{\mathrm{T}}\vec{n}_5 = \vec{n}_2^{\mathrm{T}}\vec{n}_5 = 0$$

which can be written in the following vector formulation.

$$C_{\mathrm{angl}_1}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(1,6,2)}\vec{p})^2 = 0$$

$$C_{\mathrm{angl}_2}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(1,11,2)}\vec{p})^2 = 0$$

$$C_{\mathrm{angl}_3}(\vec{p}) = (\vec{p}^{\mathrm{T}}L_{(6,11,2)}\vec{p})^2 = 0$$

and grouped into a single angle constraint function

$$C_{\mathrm{angl}}(\vec{p}) = \sum_{i=1}^{3} C_{\mathrm{angl}_i}(\vec{p}) = 0.$$

The 6th constraint can be expressed according to

$$d_1 + d_3 = d_2 + d_4$$

and then formulated by

$$C_{\mathrm{dist}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(4,5,9,10)}\vec{p})^2 = 0.$$

Since we assume that the cylinder axis is parallel to the planes S_1, S_2, S_3, S_4, the distance from the cylinder axis to one of these planes could be defined as the distance from one particular point X_0 of the axis and the given plane. The 8th constraint can be formulated by:

$$d(X_0, S_1) = d(X_0, S_3), \qquad d(X_0, S_2) = d(X_0, S_4).$$

Taking into account that S_1, S_3 have opposite orientation as well as S_2, S_4, these equations can be written as:

$$X_0^{\mathrm{T}}\vec{n}_1 + d_1 = -X_0^{\mathrm{T}}\vec{n}_1 + d_3, \qquad X_0^{\mathrm{T}}\vec{n}_2 + d_2 = -X_0^{\mathrm{T}}\vec{n}_2 + d_4$$

leading to:

$$2X_0^{\mathrm{T}}\vec{n}_1 + d_1 - d_3 = 0, \qquad 2X_0^{\mathrm{T}}\vec{n}_2 + d_2 - d_4 = 0.$$

By considering X_0 as defined in Section 2.4 (the cylinder) and by using the set of Eqs. (9) and (15) the last equations are written as:

$$-2[u, v, w]^{\mathrm{T}}\vec{n}_1 + d_1 - d_3 = 0,$$

$$-2[u, v, w]^{\mathrm{T}}\vec{n}_2 + d_2 - d_4 = 0$$

where u, v, w are the cross coefficients of the cylinder equation. A vector formulation of these equations is then given by:

$$C_{\mathrm{axe\_pos}_1}(\vec{p}) = (-2\vec{p}^{\mathrm{T}}L_{(1,22,2)}\vec{p} + i^{\mathrm{T}}_{(4,-5)}\vec{p})^2 = 0$$

$$C_{\mathrm{axe\_pos}_2}(\vec{p}) = (-2\vec{p}^{\mathrm{T}}L_{(6,22,2)}\vec{p} + i^{\mathrm{T}}_{(9,-10)}\vec{p})^2 = 0.$$

The cylinder axis position constraint is then:

$$C_{\mathrm{axe\_pos}}(\vec{p}) = C_{\mathrm{axe\_pos}_1}(\vec{p}) + C_{\mathrm{axe\_pos}_2}(\vec{p}) = 0.$$

The cylinder circularity constraint (9th constraint) is implicitly defined by the Eq. (15) and taking into account the constraint 7 which assumes that the cylinder axis is the same as normal $\vec{n}_5$ these equations are written as:

$$C_{\mathrm{circ}_{\mathrm{cyl}_1}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{16}\vec{p} + \vec{p}^{\mathrm{T}}U_{(11,11)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_4}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{19}\vec{p} + \vec{p}^{\mathrm{T}}L_{(11,12,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_2}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{17}\vec{p} + \vec{p}^{\mathrm{T}}U_{(12,12)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_5}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{20}\vec{p} + \vec{p}^{\mathrm{T}}L_{(11,13,0)}\vec{p})^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_3}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{18}\vec{p} + \vec{p}^{\mathrm{T}}U_{(13,13)}\vec{p} - 1)^2$$

$$C_{\mathrm{circ}_{\mathrm{cyl}_6}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{21}\vec{p} + \vec{p}^{\mathrm{T}}L_{(12,13,0)}\vec{p})^2$$

grouped then into a single constraint:

$$C_{\mathrm{circ}}(\vec{p}) = \sum_{k=1}^{6} C_{\mathrm{circ}_k}(\vec{p}) = 0.$$

The 10th constraint is satisfied if the centre of the sphere (21) satisfies the cylinder axis Eq. (11). We can show easily that by assuming the constraint 7, by using the set of Eqs. (9) and (15) and by requiring the coefficient a of the sphere to be unit, Eq. (11) leads to the following equations:

$$n_{x_5}(v_s - v_c) = n_{y_5}(u_s - u_c)$$

$$n_{x_5}(w_s - w_c) = n_{z_5}(u_s - u_c)$$

$$n_{y_5}(w_s - w_c) = n_{y_5}(v_s - v_c)$$

where u_s means the coefficient u related to the sphere equation, etc. Thus, these equations can be written using the vector form:

$$(\vec{p}^{\mathrm{T}}T_{(11,12,22,23)}\vec{p})^2 = 0$$

$$(\vec{p}^{\mathrm{T}}T_{(11,13,22,24)}\vec{p})^2 = 0$$

$$(\vec{p}^{\mathrm{T}}T_{(12,13,23,24)}\vec{p})^2 = 0$$

and the constraint 10 can then be stated as:

$$\begin{aligned} C_{\mathrm{sph}_{\mathrm{center}}}(\vec{p}) = (\vec{p}^{\mathrm{T}}T_{(11,12,22,23)}\vec{p})^2 + (\vec{p}^{\mathrm{T}}T_{(11,13,22,24)}\vec{p})^2 \\ + (\vec{p}^{\mathrm{T}}T_{(12,13,23,24)}\vec{p})^2 \\ = 0. \end{aligned}$$

The 11th constraint is imposed by equating the sphere radius Eq. (22) to the cylinder radius Eq. (13). Requiring again the coefficient a of the sphere to be unit and by using the set of Eqs. (9) and (15) this equality can be written using the vector form:

$$C_{\mathrm{equ}_{\mathrm{radius}}}(\vec{p}) = (\vec{i}^{\mathrm{T}}_{(25,30)}\vec{p} + \vec{p}^{\mathrm{T}}U_{(27,29,22,24)}\vec{p})^2 = 0.$$

The 12th constraint imposes a fixed position of the four plane surfaces S_1, S_2, S_3, S_4 with respect to the cylinder axis. It is formulated as:

$$\sqrt{2}(d_1 + d_3) = 2r_{\mathrm{cylinder}}.$$

By squaring this equation and by using the set of Eqs. (9) and (15) it can be written as:

$$(d_1 + d_3)^2 = 2(u_c^2 + v_c^2 + w_c^2 - d_c^2).$$

Thus this constraint can be put using the vector form:

$$C_{\mathrm{median}}(\vec{p}) = (\vec{p}^{\mathrm{T}}(I_{(4,1)} - 2U_{(22,24)})\vec{p} + 2\vec{i}^{\mathrm{T}}_{25}\vec{p})^2 = 0.$$

Appendix C. Levenberg–Marquardt algorithm

Here are the main steps of the Levenberg–Marquardt algorithm applied to a simple optimization function:

$$E(\vec{p}) = F(\vec{p}) + C(\vec{p})$$

```
$\alpha = \alpha_0$ % initialization
$E_{\text{decrease}}$ = big value
while $E_{\text{decrease}} > \epsilon$ % a threshold
  Do $G_E = \text{Grad}(E(\vec{p}) = \frac{\partial}{\partial \vec{p}}(E(\vec{p}))$
  Loop: $H_E = \text{Hessian}(E(\vec{p}) = \frac{\partial^2}{\partial^2 \vec{p}}(E(\vec{p}))$
  $H_E = H_E + \alpha(\text{diag}(H_E))$
    solve $H_E \delta\vec{p} = -G_E$
    $\vec{p}_{\text{updated}} = \vec{p} + \delta\vec{p}$
    $E_{\text{decrease}} = E(\vec{p}_{\text{updated}}) - E(\vec{p})$
    if $E_{\text{decrease}} > 0$

    increase $\alpha$
    go to Loop

    else

    $\vec{p} = \vec{p}_{updated}$
    decrease $\alpha$

    end if

end while
```

Here is a simple example of an optimization function and its derivatives:

$$E(\vec{p}) = F(\vec{p}) + C(\vec{p})$$

$$F(\vec{p}) = \ \vec{p}^{\mathrm{T}}\mathscr{H}\vec{p} \qquad \text{the least-squares function}$$

$$C(\vec{p}) = \lambda(\vec{p}^{\mathrm{T}}A\vec{p} - 1)^2 \qquad \text{the weighted constraint function}$$

$$G_E = 2\mathscr{H}\vec{p} + 4\lambda A\vec{p}(\vec{p}^{\mathrm{T}}A\vec{p} - 1)$$

$$H_E = 2\mathscr{H} + \lambda[4(\vec{p}^{\mathrm{T}}A\vec{p} - 1)A^{\mathrm{T}} + 8(A\vec{p})(A\vec{p})^{\mathrm{T}}].$$

From this example we can notice the usefulness of the matrix formulation: the optimization function is compact, its derivatives are easy to compute using elementary matrix algebra rules and all the data terms are encapsulated into $\mathscr{H}$ (which needs to be calculated only once).

Appendix D

Solving the linear system $H_E\delta\vec{p} = -G_E$ in the Levenberg–Marquardt algorithm has numerical perturbations due to the ill-conditioned matrix H_E for large values of λ_k. The key of the solution proposed to overcome this problem consist in splitting the system in two subsystems. The matrix associated with one of the subsystems will hold the matrix components which are sensitive to λ_k variations, the other matrix will hold the components which are not. Thus, both of the matrices will be well-conditioned. The two systems will be then solved consecutively and separately.

Let us set the coefficient α in Levenberg–Marquardt algorithm to zero without loose of generality. The system $H_E\delta\vec{p} = -G_E$ could be written more explicitly as:

$$(L + R^{\mathrm{T}}DR)\delta\vec{p} = -G_E \tag{D1}$$

where

$$L = 2\mathscr{H} + 4\sum_{k=1}^{M} \lambda_k C_k(\vec{p})A_k \tag{D2}$$

$$R^{\mathrm{T}} = \begin{bmatrix} \frac{\partial C_1}{\partial \vec{p}} & \frac{\partial C_2}{\partial \vec{p}} & \cdots & \frac{\partial C_M}{\partial \vec{p}} \end{bmatrix}$$

$$D = \begin{bmatrix} 2\lambda_1 & 0 & \cdots & 0 \\ 0 & 2\lambda_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 2\lambda_M \end{bmatrix}$$

$$G_E = 2\mathscr{H}\vec{p} + R^{\mathrm{T}}\nabla_C$$

$$\nabla_C = [2\lambda_1 C_1(\vec{p}), 2\lambda_2 C_2(\vec{p}), \ldots, 2\lambda_M C_M(\vec{p})]^{\mathrm{T}}.$$

As mentioned, the matrix L is well behaved since its condition number remains stable when the values of λ_k increase, whereas the condition number of $R^{\mathrm{T}}DR$ increases with λ_k.

Consider the matrix $S = D^{-1}(RR^{\mathrm{T}})^{-1}R$. By multiplying Eq. (D1) on both sides by S we get a system of M equations:

$$(SL + R)\vec{\delta}p = -SG_E \tag{D3}$$

when λ_k values increase and become large $\|S\|$ tends towards zero whereas $\|R\|$ remains stable since it is independent of λ_k so we get $\|SL\| \ll \ \|R\|$ and thus the system (D3) can be approximated by

$$R\delta\vec{p} = -SG_E. \tag{D4}$$

So now with this system of M equations and N (size of $\delta\vec{p}$) unknowns we can extract M components of $\delta\vec{p}$. The rank of R is equal to M so we can find an orthogonal matrix Q such:

$$QR^{\mathrm{T}} = \begin{bmatrix} U \\ [0] \end{bmatrix} \tag{D5}$$

where U is an (M,M) upper triangular non-singular matrix.

Since $QQ^{\mathrm{T}} = I$, (D4) can be written as

$$RQ^{\mathrm{T}}Q\delta\vec{p} = -SG_E. \tag{D6}$$

By splitting $Q\delta\vec{p}$ into $[\delta\vec{z}_1, \delta\vec{z}_2]$ where $\delta\vec{z}_1$ and $\delta\vec{z}_2$ have a size of, respectively, M and $N - M$ we get from (D6):

$$U^{\mathrm{T}}\delta\vec{z}_1 = -SG_E \tag{D7}$$

and then $\delta\vec{z}_1$ could be deduced from this equation. Now it remains to compute $\delta\vec{z}_2$.

Consider the matrix V whose columns are the basis of the null space of R. We have $RV = [0]$. By multiplying (D1) by

V^{T} we get:

$$V^{\mathrm{T}} L \delta \vec{p} = -V^{\mathrm{T}} G_E. \tag{D8}$$

Now since $RV = RQ^{\mathrm{T}}QV = [0]$ by using (D5) and splitting QV into $[J_1^{\mathrm{T}}, J_2^{\mathrm{T}}]^{\mathrm{T}}$, where J_1 and J_2 are, respectively, (M,M) and $(N - M,M)$ matrices we get

$$[U^{\mathrm{T}}, [0]^{\mathrm{T}}] \begin{bmatrix} J_1 \\ J_2 \end{bmatrix} = [0].$$

This implies that $J_1 = [0]$ since U is non singular and J_2 could be set to an arbitrary value say I. Then we can set $QV = [[0]^{\mathrm{T}}, I^{\mathrm{T}}]^{\mathrm{T}}$.

The system (D8) can be written:

$$V^{\mathrm{T}} Q^{\mathrm{T}} Q L Q^{\mathrm{T}} Q \delta \vec{p} = -V^{\mathrm{T}} G_E$$

$$(QV)^{\mathrm{T}} (QLQ^{\mathrm{T}}) Q \delta \vec{p} = -V^{\mathrm{T}} G_E$$

$$[[0]^{\mathrm{T}}, I^{\mathrm{T}}] QLQ^{\mathrm{T}} \begin{bmatrix} \delta \vec{z}_1 \\ \delta \vec{z}_2 \end{bmatrix} = -V^{\mathrm{T}} G_E.$$

If we denote the matrix QLQ^{T} by W such as:

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

we get:

$$[[0]^{\mathrm{T}}, I^{\mathrm{T}}] \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} \delta \vec{z}_1 \\ \delta \vec{z}_2 \end{bmatrix} = -V^{\mathrm{T}} G_E$$

from which we extract the system

$$W_{22} \delta \vec{z}_2 = -V^{\mathrm{T}} G_E - W_{21} \delta \vec{z}_1 \tag{D9}$$

and δz_2 can be then computed.

The computation of the term SG_E in (D7) is expensive. Practically it is faster to use a simplified expression. From (D2) we get

$$SG_E = D^{-1}(RR^{\mathrm{T}})^{-1} R (2\mathscr{H}\vec{p} + R^{\mathrm{T}} \nabla_C) \tag{D10}$$

$$= D^{-1}(2(RR^{\mathrm{T}})^{-1} R \mathscr{H}\vec{p} + \nabla_C)$$

$$= D^{-1}(2(RQ^{\mathrm{T}}QR^{\mathrm{T}})^{-1} RQ^{\mathrm{T}} Q\mathscr{H}\vec{p} + \nabla_C)$$

$$= D^{-1}(2[U^{-1}, [0]] Q\mathscr{H}\vec{p} + \nabla_C).$$

By splitting $Q\mathscr{H}\vec{p}$ into $[\vec{l}_1^{\mathrm{T}}, \vec{l}_2^{\mathrm{T}}]^{\mathrm{T}}$ where $\vec{l}_1$ and $\vec{l}_2$ have, respectively, sizes of M and $N - M$ we get

$$SG_E = D^{-1}(2U^{-1}\vec{l}_1 + \nabla_C) \tag{D11}$$

and $\delta \vec{z}_1$ can be then computed with

$$U^{\mathrm{T}} \delta \vec{z}_1 = -D^{-1}(2U^{-1}\vec{l}_1 + \nabla_C). \tag{D12}$$

Similarly the expression of $V^{\mathrm{T}} G_E$ in (D9) can be simplified:

$$V^{\mathrm{T}} G_E = V^{\mathrm{T}}(2\mathscr{H}\vec{p} + R^{\mathrm{T}} \nabla_C) \tag{D13}$$

$$= 2V^{\mathrm{T}} \mathscr{H}\vec{p}, \qquad \text{since } RV = [0]$$

$$= 2V^{\mathrm{T}} Q^{\mathrm{T}} Q\mathscr{H}\vec{p}$$

$$= 2[[0]^{\mathrm{T}}, I^{\mathrm{T}}] \begin{bmatrix} \vec{l}_1 \\ \vec{l}_2 \end{bmatrix}$$

$$= 2\vec{l}_2$$

and the computation of δz_2 is the performed with the following system

$$W_{22} \delta \vec{z}_2 = -2\vec{l}_2 - W_{21} \delta \vec{z}_1. \tag{D14}$$

Once $\delta \vec{z}_2$ and $\delta \vec{z}_1$ are computed the $\delta \vec{p}$ vector is deduced with

$$\delta \vec{p} = Q^{\mathrm{T}} \delta z. \tag{D15}$$

To recapitulate, the resolution of the equation $H_E \delta \vec{p} = -G_E$ in the Levenberg–Marquardt algorithm has to be performed through the following steps:

(1) Compute D, ∇_C, R.
(2) Compute Q and U from R using elementary geometric transformation (e.g. Householder transformation [63, p. 224]).
(3) Compute $Q\mathscr{H}\vec{p}$ and extract $\vec{l}_1^{\mathrm{T}}$ and $\vec{l}_2$.
(4) Compute $\delta \vec{z}_1$ from (D12).
(5) Compute $W = QLQ^{\mathrm{T}}$ and extract W_{22} and W_{21}.
(6) Compute $\delta \vec{z}_2$ from (D14).
(7) Compute $\delta \vec{p}$ from (D15).

References

[1] Anderl R, Mendegen R. Modelling with constraints: theoretical foundation and application. Comput Aided Design 1996;28(3):155–168.

[2] Feng CX, Kusiak A. Constraints-based design of parts. Comput Aided Design 1995;27(5):343–352.

[3] Hoffmann CM, Vermeer PJ. Geometric constraint solving in $R2$ and $R3$. In: Dingzhu D, Frank H, editors. Computing in Euclidean geometry, 2nd edition. World Scientific Publishing, 1995.

[4] Boring AH. The programming language aspect of ThingLab, a constrained oriented simulation laboratory. ACM TOPALS 1981;3(4):353–387.

[5] Heydon A, Nelson G. The Juno-2 constraint-based-drawings editor, SRC Research Report 131a, 1994.

[6] Light RA, Gossard DC. Modification of geometric models through variational geometry. Comput Aided Design 1982;14(4):209–214.

[7] Sutherland IE. Sketchpad: a man–machine graphical communication system. Proc Joint Comput Conf 1963;329–346.

[8] Anantha R, Kramer GA, Crawford RH. Assembly modelling by geometric constraint satisfaction. Comput Aided Design 1996;28(9):707–722.

[9] Buchberger B. Application of Grobner basis in non-linear computational geometry. In: Kapur D, Mundy J, editors. Geometric reasoning, Cambridge, MA: MIT Press, 1989.

[10] Kondo K. Algebraic method for manipulation of dimensional relationships in geometric models. Geom Aided Design 1992;24(3):141–147.

[11] Wu WT. Basic principles of mechanical theorem proving in geometry, Berlin: Springer, 1993.

[12] Arbab F, Wang B. Reasoning about geometric constraints. In: Yoishikawa H, Holden T, editors. Intelligent CAD II, Amsterdam: North Holland, 1990. pp. 93–107.

[13] Sunde G. Specification of shape by dimensions and other geometric constraints. In: Vozny MJ, editor. Geometric modelling for CAD applications, Amsterdam: North Holland, 1988. pp. 199–213.

[14] Wetkamp RC. Geometric constraint management with quanta. In: Brown DC, editor. Intelligent computer aided design, Amsterdam: North Holland, 1992. pp. 409–426.

[15] Bouma W, Fudos I, Hoffman C, Cai J, Paige R. Geometric constraint solver. Comput Aided Design 1995;27(6):487–501.

[16] Eggli L, Hsu CY, Bruderlin BD, Elbert G. Inferring 3D models from freehand sketches and constraints. Comput Aided Design 1997;29(2):101–112.

[17] Fudos I, Hoffman C. Constraints-based parametric conics for CAD. Comput Aided Design 1996;28(2):91–100.

[18] Gao XS, Chou SC. Solving geometric constraint systems. I. A global propagation approach. Comput Aided Design 1988;30(1):47–54.

[19] Hoffmann CM, Vermeer PJ. A spatial constraint problem. In: Proc. 2nd Workshop on Computational Kinematics. Sophia Antipolis, 1995. pp. 83–92.

[20] Hoffmann CM, Lomonosov A, Sitharan M. In: Smoka G, editor. Finding solvable subsets of constraint graphs, *Lecture Notes in Computer Science 1330*Berlin: Springer, 1997. pp. 463–477.

[21] Besl PJ, Jain RC. Three dimensional object recognition. ACM Computing Surveys 1985;17(1):75–145.

[22] Bolles RC, Cain RA. Recognizing and locating partially visible objects. Int J Robotics Res 1982;1(3):57–82.

[23] Bolles RC, Horaud P. 3DPO, a three dimensional part orientation system. Int J Robotics Res 1986;5(3):3–26.

[24] Forsyth D, Mundy JL, Zisserman A, Coelho C, Heller A, Rothwell C. Invariant descriptors for 3D object recognition and pose. IEEE Trans PAMI 1991;13(10):971–991.

[25] Grimson WE. The role of geometric constraints, London: MIT Press, 1990.

[26] Lowe DG. Fitting parameterized three dimensional models to images. IEEE Trans PAMI 1991;13(5):441–450.

[27] Gmur E, Bunke H. 3D Object recognition based on subgraph matching in polynomial time. In: Sanfeliu, Mohr, Pavildis, editors. Structural pattern analysis, Singapore: World Scientific, 1990. pp. 131–147.

[28] Shapiro LG, Haralick RM. Structural descriptions and inexact matching. IEEE Trans PAMI 1981;3(5):504–519.

[29] Herman I. The use of projective geometry in computer graphics, *Lecture Notes in Computer Science 564*. Berlin: Springer, 1992.

[30] Fitzgibbon AF, Eggert DW, Fisher RB. High-level CAD model acquisition from range images. Comput Aided Design 1997;29(4):321–330.

[31] Rockwood AP, Winget J. Three-dimensional object reconstruction from two-dimensional images. Comput Aided Design 1997; 29(4):279–286.

[32] Shin BS, Shin YG. Fast 3D solid model reconstruction from orthographic views. Comput Aided Design 1998;30(1):63–76.

[33] Yan QW, Chen CLP, Tang Z. Efficient algorithm for the reconstruction of 3D objects from orthographic projections. Comp. Aided Design 1994;26(9):699–717.

[34] Boyer KL, Mirza MJ, Ganguly G. The robust sequential estimator. IEEE Trans PAMI 1994;16(10):987–1001.

[35] Flynn PJ, Jain AK. Surface classification: hypothesizing and parameter estimation. In: Proc. IEEE Comp. Soc. CVPR. June 1988. pp. 261–267.

[36] Kumar S, Han S, Goldgof D, Boyer K. On recovering hyperquadrics from range data. IEEE Trans PAMI 1995;17(11):1079–1083.

[37] Chen Y, Medioni G. Object modeling by registration of multiple range images. Proc IEEE Int Conf Robotics Automation 1991;2:724–729.

[38] Shun HY, Ikeuchi K, Reddy R. Principal component analysis with missing data and its application to polyhedral object modelling. IEEE Trans PAMI 1995;17(9):855–867.

[39] Soucy M, Laurendo D. Surface modelling from dynamic integration of multiple range views, Proc. 11th Int. Conf. Pattern Recognition. 1992. pp. 449–452.

[40] Vemuri BC, Aggrawal JK. 3D model construction from multiple views using range and intensity data. Proc. CVPR. 1986. pp. 435–437.

[41] Varady T, Martin RR, Cox J. Reverse engineering of geometric models, an introduction. Comput Aided-Design 1997;29(24):255–268.

[42] Porrill J. Optimal combination and constraints for geometrical sensor data. Int J Robotics Res 1988;7(6):66–78.

[43] De Geeter J, Brussel HV, De Schutter J, Decreton M. Smoothly constrained Kalman filter. IEEE Trans PAMI 1997;19(10):1171–1177.

[44] Bolle RM, Cooper DB. On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data. IEEE Trans PAMI 1986;8(5):619–638.

[45] Werghi N, Fidher RB, Robertson C, Ashbrook A. Improving model shape acquisition by incorporating geometric constraints. Proc. BMVC, Essex, September 1997. pp. 530–539.

[46] Bell RJ. An elementary treatise on coordinate geometry, London: McMillan and Co, 1910.

[47] Requicha AAG. Representation of tolerances in solid modelling: issues and alternative approaches. In: Boyse JW, Pickett MS, editors. Solid modelling by computers: from theory to applications, New York: Plenum Press, 1984. pp. 3–22.

[48] Goldberg DE. Genetic algorithms in search, optimization and machine learning, Reading, MA: Addison–Wesley, 1989.

[49] Michalewicz Z. Genetic algorithms + data structures = evolution programs, Berlin: Springer, 1996.

[50] Taura T, Nagasaka I, Yamagishi A. Application of evolutionary programming to shape design. Comput Aided Design 1988;30(1):29–35.

[51] Jian D, Vijayan S. Manufacturing feature determination and extraction. Part I: optimal volume segmentation. Comput Aided-Design 1997;29(6):427–440.

[52] Robertson C, Fisher RB, Corne D, Werghi N, Ashbrook AP. Investigating evolutionary optimisation of constrained functions to capture descriptions from range data. Proc. 3rd On-line World Conference on Soft Computing (WSC3), 1998.

[53] Wallace DR, Jakiela MJ, Flowers WC. Design search under probabilistic specifications using genetic algorithms. Comput Aided Design 1996;28(5):405–421.

[54] Fletcher R. Practical methods of optimization, New York: Wiley, 1987.

[55] Gill PE, Murray W, Wright MH. Practical optimization, New York: Academic Press, 1981.

[56] Lawrence KL, Muthukrishna SN, Nambiar RV. Refinement of 3D meshes at surface intersections. Comput Aided Design 1995;27(8):637–645.

[57] Ong CJ, Wong YS, Loh HT, Hong XG. An optimization approach for biarc curve-fitting of B-spline curves. Comput Aided Design 1996;28(12):951–959.

[58] Wang X, Cheng F, Brian B. Energy and B-spline interproximation. Comput Aided Design 1997;29(7):405–421.

[59] Rockafellar RT. Convex analysis, Princeton, NJ: Princeton University Press, 1970.

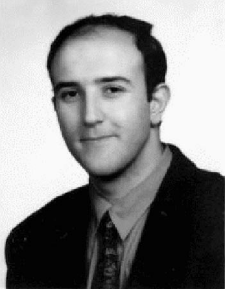
[60] Fiacco AV, McCormick GP. Nonlinear programming: sequential unconstrained minimization techniques, New York: Wiley, 1968.

[61] Broyden CG, Attia NF. Penalty functions, Newton's method and quadratic programming. J Optim Theory Appl 1988;58(3):377–385.

[62] Hoover A, Jean-Baptiste G, Jiang X, Flynn PJ, Bunke H, Goldgof D, Bowyer K, Eggert D, Fitzgibbon A, Fisher R. An experimental

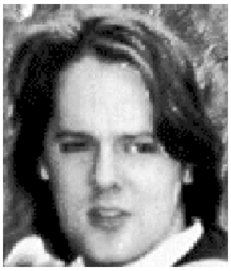
comparison of range segmentation algorithms. IEEE Trans PAMI 1996;18(7):673–689.

[63] Golub GH, Van Loan CF. Matrix computations, 3. Baltimore, MD: Johns Hopkins University Press, 1996.



Naoufel Werghi is a Research Associate in the Department of Artificial Intelligence at the University of Edinburgh. He received the Ph.D. in Computer Vision from the University of Strasbourg, France (1996), an MS in Signal processing and automation from the University of Rouen, France (1993) and Principal Engineering degree in Electrical Engineering from the National Engineering School of Monastir, Tunisia (1992). His research interests include object modelling, object recognition and localization and image processing.

Robert B. Fisher received a BS with honors (Mathematics) from California Institute of Technology (1974) and an MS (Computer Science) from Stanford University (1978). He received his Ph.D. from University of Edinburgh (awarded 1987), investigating computer vision in the Department of Artificial Intelligence. Dr. Fisher is a Reader at the Department of Artificial Intelligence, University of Edinburgh. His research covers topics in high level computer vision, and he directs a research project investigating three dimensional model-based vision, automatic model acquisition and robot grasping. He teaches general and industrial vision courses for undergraduate, MSc and Ph.D. level students.



Craig Robertson received his BSc degree in Mathematics from Coventry University in 1991 and an MSc degree in Statistics and Operational Research from the University of Birmingham in 1992. He has recently completed a Ph.D. degree in Computer Science which included work at both Newcastle and Reading Universities. His research interests include computer vision as well as neural, genetic and emergent algorithms. He is currently employed as a Reseach Associate in the Computer Vision Laboratory at the University of Edinburgh Artificial Intelligence Department. He is a member of the British Machine Vision Association.

Anthony Ashbrook received his BSc Hons degree in Electrical and Electronic Engineering at the University of Aston, Birmingham in 1992. He then pursued his Ph.D. in the field of Computer Vision in the Electronic Systems Group at the University of Sheffield. Here he studied the application of statistical methods for representing and classifying shape. He is now working in the Vision Group within the Department of Artificial Intelligence at the University of Edinburgh, where he is developing techniques for automatically modelling articulated objects from examples.

Finding Surface Correspondence for Object Recognition and Registration using Pairwise Geometric Histograms

A. P. Ashbrook, R. B. Fisher, C. Robertson and N. Werghi

Department of Artificial Intelligence
The University of Edinburgh
5, Forrest Hill, Edinburgh, EH1 2QL
Telephone: +44 131 650 4504
Fax: +44 131 650 6899
anthonya@dai.ed.ac.uk

Abstract. Pairwise geometric histograms have been demonstrated as an effective descriptor of arbitrary 2-dimensional shape which enable robust and efficient object recognition in complex scenes. In this paper we describe how the approach can be extended to allow the representation and classification of arbitrary $2\frac{1}{2}$- and 3-dimensional surface shape. This novel representation can be used in important vision tasks such as the recognition of objects with complex free-form surfaces and the registration of surfaces for building 3-dimensional models from multiple views. We apply this new representation to both of these tasks and present some promising results.

1 Introduction

Finding a correspondence between two or more surfaces is a frequently encountered problem in many computer vision tasks. When surface based descriptions are used for object recognition, the hypothesis that a particular object is in a scene is confirmed by finding a good correspondence between scene and model surfaces [6]. When constructing geometric models of objects by merging multiple range images taken from different viewpoints, the surfaces described by each range image require registration into a common coordinate frame [3, 1]. This can be done by finding the correspondence between portions of the object's surface which is common to two or more views.

In this paper we present a novel representation for arbitrary $2\frac{1}{2}$- and 3-dimensional surface data which enables correspondences to be found reliably and efficiently. The representation is based on pairwise geometric histograms which have previously been demonstrated as a representation for 2-dimensional shape data for object recognition applications [4].

The approach that we are proposing determines whether two surfaces have a correspondence as follows:

1. Each of the surfaces is approximated by a triangular mesh. The details of this approximation and the algorithms we have employed for this are presented in Section 3.1.
2. Each triangular mesh facet is represented by a pairwise geometric histogram which records the relationship between this facet and the surrounding facets within some specified neighbourhood. This representation is discussed in Section 3.2.
3. Correspondences between individual facets are found by matching their respective geometric histograms. These *local* correspondences provide hypotheses for the correspondence between the two surfaces. The metric employed for matching geometric histograms is described in Section 4.
4. The *global* surface correspondence is found by finding consistent local hypotheses using a probabilistic Hough transform. This is discussed in Section 5.

2 Background

A number of approaches to the problem surface registration have been developed from the "iterated closest point" (ICP) algorithm proposed by Besl and McKay [2]. These algorithms have been popular for registering multiple views of an object for model construction and for refining pose in object recognition tasks. The central idea behind this algorithm is that by forming correspondences between points on one surface and their nearest neighbours on another and then minimising the distances between them, the registration of the two surfaces is improved. If this process is iterated the registration of the surfaces often converges. The approach is computationally expensive because of its use of raw surface point data and because of the iterative nature of the algorithm. A more serious problem is that the algorithm is not guaranteed to converge, sometimes getting caught in local minima, and typically requires good initial alignment of the surfaces to get a reasonable solution. One of the advantages of the ICP approach is that, because it uses all of the surface data available, when it does converge the registration can be very accurate. The algorithm is also suitable for arbitrary classes of surface.

Other researchers have used interest points on the surface instead of all of the surface data and formed correspondences by matching geometric descriptors of those points. Thirion [13] proposes the use of extremal points on 3-dimensional surfaces which can be characterised by a number of properties such as their curvature. Interest points with similar properties are treated as potential correspondents and the transformation that aligns the surfaces is determined from triplets of corresponding pairs. Recently, Johnson and Hebert [9] have proposed a novel interest point descriptor which allows point correspondences to be formed between surfaces. In their approach the interest points are defined by the vertices of a polygonal mesh fitted to the surface. At each vertex the geometric relationship with all of the other mesh vertices are recorded in a 2-dimensional *spin-image* which is invariant to rigid transformations of the surface. Interest point correspondences are found by identifying points with similar spin-images.

Local surface features such as edges and surface patches have also been used to determine the correspondence between two surfaces [5]. Initially all features on the first surface are considered as potential correspondents of features of the same class on the second surface. The number of potential correspondences is then quickly reduced using approaches based on geometric constraints such as the interpretation tree. Each pair of matched features provides a constraint on the transformation that aligns the surfaces and these are used to determine the best global alignment. The motivation for using features is to reduce the amount of data to be processed whilst maintaining valuable information needed to perform matching and constrain the alignment transformation. The disadvantage is that a particular choice of features can limit the scope of the algorithm to particular classes of surfaces.

3 A Novel Surface Shape Representation

3.1 Surface Reconstruction and Approximation

Initially a given surface S, acquired using a range sensor, is described by a set of points samples $P = \{p_1, \ldots, p_N\}$. The points may represent a single view of the surface or a number of different views, for example from different viewpoints around an object. The point set is then used to construct a triangular mesh approximation $\hat{S}$ to the original surface, where $\hat{S} = \{t_1, \ldots, t_M\}$ and t_i is a triangular facet of the mesh.

It is important to clarify at this stage that the only requirement of the mesh is that it is a good approximation of the surface shape. No assumptions are made about the actual distribution of facets over the surface as this is unlikely to be repeatable. To minimise the amount of memory and computation needed to solve the correspondence problem, the mesh should also contain the smallest number of facets needed to give a good approximation of the surface.

A number of algorithms have been proposed for reconstructing a triangular faceted mesh from a set of points. In the work presented here an initial, regular mesh was constructed from the sampled point data using a reconstruction algorithm by Hoppe *et al* [8]. The resulting regular mesh was then refined to minimise the number of facets whilst maintaining most of the surface shape using a surface simplification algorithm by Garland and Heckbert [7].

There are a number of advantages in using a triangular mesh to approximate the surface to be represented instead of more complex features such as quadric patches, the most obvious being efficiency. Constructing a mesh is also significantly more straightforward than segmenting a surface into more complex features. A second important issue is scope. Any surface can be approximated by a triangular mesh but selecting a fixed set of features can impose limitations on the types of surfaces that can be described. Another important issue is that of stability. If surface patches are assigned to different classes based on their shape then borderline cases can result in sudden changes in the representation because of slightly different viewing conditions or noise.

The disadvantage of using a triangular mesh is that it requires many facets to describe surfaces with high curvature to a high degree of accuracy. By statistically modelling the shape error introduced by the triangular shape approximation, it is still possible to obtain a good shape representation when only a relatively small number of facets are used.

3.2 Histogram Construction

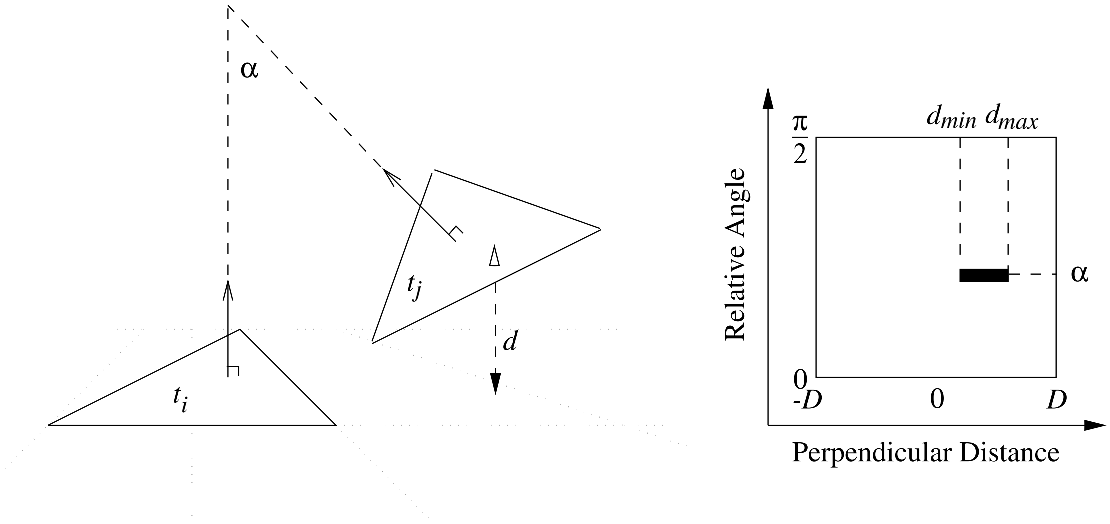
A pairwise geometric histogram h_i is constructed for each triangular facet t_i in a given mesh which describes its pairwise relationship with each of the other surrounding facets within a predefined distance. This distance controls degree to which the representation is a local description of shape. The histogram is defined such that it encodes the surrounding shape geometry in a manner which is invariant to rigid transformations of the surface data and which is stable in the presence of surface clutter and missing surface data.

Figure 1(a) shows the measurements used to characterise the relationship between facet t_i and one of its neighbouring facets t_j. These measurements are the relative angle, α, between the facet normals and the range of perpendicular distances, d, from the plane in which facet t_i lies to all points on facet t_j. These measurements are accumulated in a 2-dimensional frequency histogram, weighted by the product of the areas of the two facets as shown in Figure 1(b). The weight of the entry is spread along the perpendicular distance axis in proportion to the area of the facet t_j at each distance. To compensate for the difference between the measurements taken from the mesh and the true measurements for the original surface, the entry is blurred into the histogram. For the work presented here a Gaussian blurring function has been used, but we intend to investigate more appropriate error models in the future. Certainly the scale of the blurring function relates to the coarseness of the mesh. The complete pairwise geometric histogram for facet t_i is constructed by accumulating these entries for each of the neighbouring facets.

For clarity, an example of a pairwise geometric histogram is presented in Figure 2(a). This has been constructed for the highlighted facet on the hemispherical mesh presented in Figure 2(b). Note that the representation only depends upon the surface shape and not on the placement of facets over the surface. This independence on the placement of the facets is important because recovering exactly the same mesh for the same surface under different viewing conditions is very unlikely, particularly if there is some surface occlusion.

4 Generating Correspondence Hypotheses

Given two surface meshes, $\hat{S}^A$ and $\hat{S}^B$, the geometric histogram representation allows correspondences between all facets, t_i^A and t_j^B, from each of the meshes to be determined. A match for facet t_i^A is determined by finding the best match between its respective pairwise geometric histogram and all of the histograms



(a) (b)

Fig. 1. (a) The geometric measurements used to characterise the relationship between two facets t_i and t_j. (b) The entry made into the pairwise geometric histogram to represent this relationship.

representing the facets in surface $\hat{S}^B$. These *local* correspondences are treated as hypotheses for the correspondence between the two surfaces S^A and S^B.

The similarity, D_{ij}, between two pairwise geometric histograms h_i and h_j is defined using the Bhattacharyya metric. This is given by the expression:

$$D_{ij} = \sum_{\alpha,d} \sqrt{h_i(\alpha,d)}\sqrt{h_j(\alpha,d)} \tag{1}$$

The Bhattacharyya metric is appropriate when the error on the data can be described using a Poisson distribution. This is a reasonable assumption for measured frequency distributions such as a geometric histogram [12]. A derivation of this metric is presented in Appendix A.

5 Hypothesis Verification

Each pair of matched mesh facets provides evidence that the surfaces to which they belong have the same shape, at least locally, and can therefore be registered. The transformation that aligns the paired facets also provides a constraint on the transformation that aligns the complete surfaces. The problem then is to determine whether there is enough evidence to support these hypotheses and, if so, to determine the transformation that aligns the surface data.

We have used an approach taken by other researchers in which N-tuples of matched features, in our case paired mesh facets, are used to estimate the

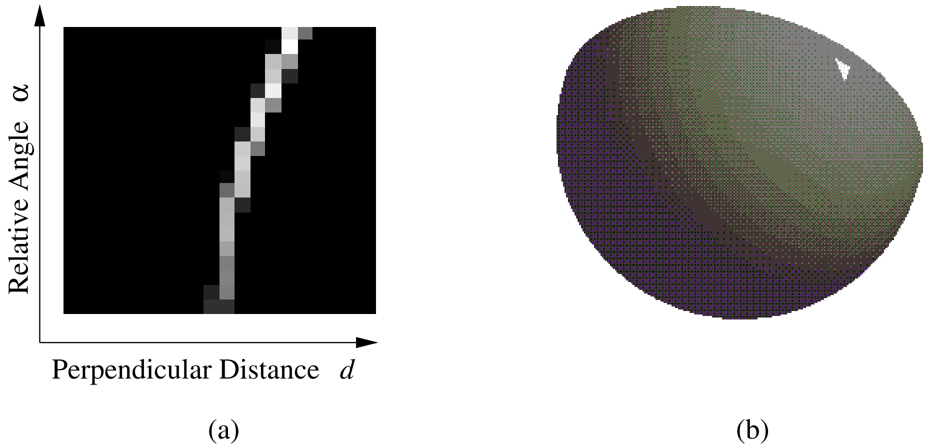


Fig. 2. (a) The geometric histogram that characterises the relationship between highlighted facet and the other facets in the mesh in (b).

alignment transformation. These estimates are then accumulated in a Hough transform resulting in a peak where there is consistency. As an improvement to this scheme we have adopted a probabilistic approach in which the error on the estimated transformation is integrated into the Hough accumulator [11]. This error is determined by statistically modelling the error between the facets and the true surface and propagating this error through the transformation estimator.

Initially 2-tuples of paired facets are used to estimate the rotation component of the alignment transformation and votes are placed in a 3-dimensional Hough transform. The number of 2-tuples can be very large so only a proportion of the largest paired facets are used. If a significant peak is found in this space then 3-tuples of paired facets are used to estimate the translation component of the alignment transformation. Again, only a proportion of the largest facets are used to allow fast operation. If a significant peak is found in the translation space then the hypothesis that the surfaces can be registered is accepted.

6 Experiments

Two applications of the proposed surface representation are presented here. The first application is the registration of two different views of an object with a complex surface. The second application is the identification and localisation of known objects in a scene. All of the data were acquired using a laser stripe range scanner with an accuracy of approximately 0.1mm. The pairwise geometric histogram parameters selected for both of these experiments are presented in Table 1.

Quantisation of Relative Angle Axis	20 bins
Quantisation of Perpendicular Distance Axis	20 bins
Maximum Perpendicular Distance	± 100mm
Maximum Relative Angle	$\frac{\pi}{2}$ radians

Table 1. The pairwise geometric histogram parameters used in the experiments presented here.

6.1 Registration of Free-form Surfaces

In this experiment the objective is to find the correspondence between two surfaces constructed from different views of an object. The surface meshes, presented in Figure 3, describe the surface of a farm animal model and consist of 1000 facets each. It should be noted that the model has quite complex, free-form surfaces which are difficult to describe using features such as quadric patches or edges.

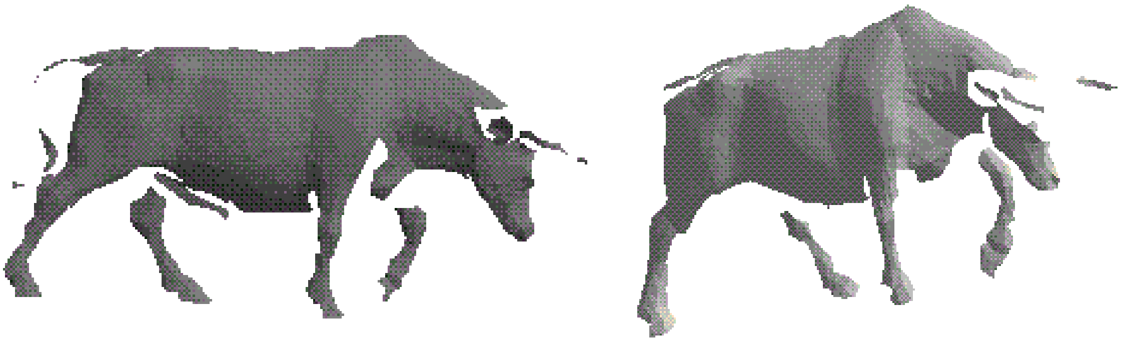


Fig. 3. The triangular meshes for two different views of the surface of a farm animal model.

Figure 4(a) presents the two surfaces in their registered positions. Certainly, from a qualitative point of view, the registration seems to have been successful. This is emphasised by the inter-meshing of the two surfaces on the rear leg of the model shown in close-up in Figure 4(b). The fact that this inter-meshing is not visible over all of the surface suggests that there is some registration error, however.

Only the largest 5% of the facets were matched and used to determine the alignment transformation. The entire registration process took approximately 4 minutes 24 seconds on a 200MHz Sun Ultra. A breakdown of these times is presented in Table 2.

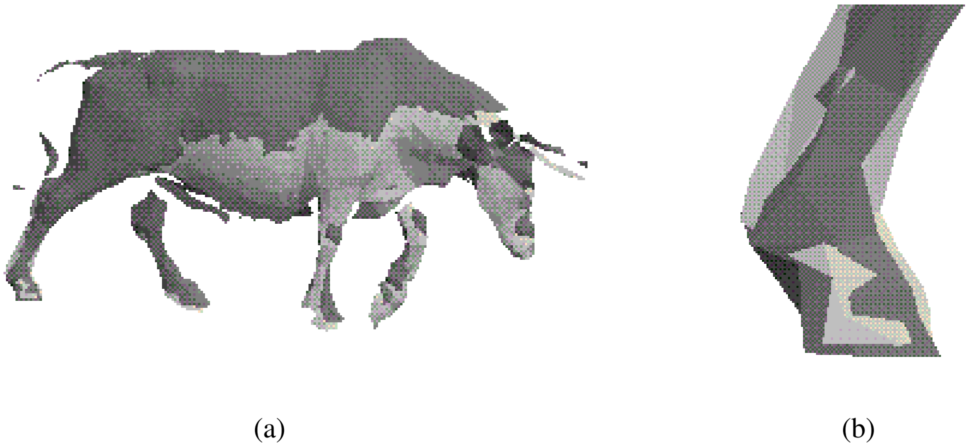


Fig. 4. (a) The two meshed surfaces in their registered positions. (b) A close-up of the rear leg of the model. The light and dark shades of grey represent the two different surfaces.

Triangular Mesh Construction	110 seconds
Geometric Histogram Construction	212 seconds
Geometric Histogram Matching	6 seconds
Resolving Hypotheses	126 seconds

Table 2. A breakdown of the time to complete the registration for each of the main algorithm stages.

6.2 Object Recognition and Pose Estimation

The objective of this experiment is to identify known objects in a scene and estimate the pose of those objects. The object models, presented in Figure 5, have been constructed from multiple views to produce a complete 3-dimensional description of all of the surfaces. Each model is represented by 1000 facets.

Figure 6 presents a scene containing two of the known models. The scene has been captured with a single range image and represented by 1000 facets.

The classification of each of the scene facets is presented in Figure 7. In each of the three images the scene facets which best match a facet from the respective model have been drawn. It can be seen that most of the facets have been classified as belonging to the correct models. Most of the incorrectly classified facets lie very close to surface discontinuities where the recovery of the surface normal is very poor. This is largely due to the mesh construction algorithm which has problems preserving discontinuities in the range data. There are also some problems with the classification of the underside of the cylinder model. This is likely to be because this surface is almost parallel to the viewing direction which makes recovery of the surface normal prone to error.

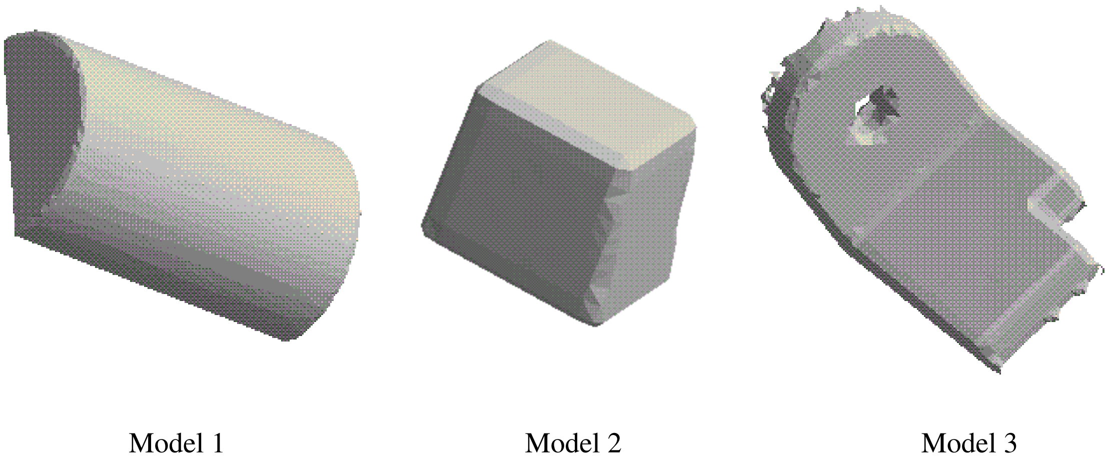


Fig. 5. The three model objects used in the recognition experiment.

Figure 8 presents the results of the recognition of pose estimation process. The original scene data is shown in the darker shade and the recognised models are shown in their estimated positions in the lighter shade. The algorithm has both determined the objects present in the scene and formed a reasonable estimation of their positions.

All of the facets were matched and then the largest 5% from each class were used to determine the model poses. The entire object recognition process took approximately 14 minutes 3 seconds on a 200MHz Sun Ultra. A breakdown of these times is presented in Table 3.

Triangular Mesh Construction	54 seconds
Geometric Histogram Construction	96 seconds
Geometric Histogram Matching	329 seconds
Resolving Hypotheses	364 seconds

Table 3. A breakdown of the time to complete the recognition process for each of the main algorithm stages.

7 Conclusions

The problem of finding a correspondence between two or more surfaces has been investigated by a number of researchers and several solutions have been proposed. The most reliable approaches are based on finding point-feature or

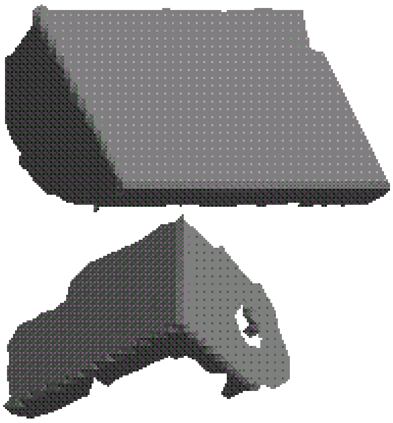


Fig. 6. The scene data used in the recognition experiment.

surface-feature correspondences between the surfaces being registered and using these to estimate the transformation that aligns the complete surfaces.

In this paper we have proposed a novel representation for surface data which enables local surface correspondences to be determined. This representation is invariant to rigid transformations of the surface data and, because of its statistical nature, allows errors in the approximation of the surfaces by triangular meshes to be modelled.

Having established local correspondences we have shown that the transformation that aligns complete surfaces can be determined using a Hough voting scheme. The advantage of using Hough voting is that it is possible to model transformation errors present in the local correspondences by adopting a probabilistic Hough transform.

To demonstrate the effectiveness of the new representation and the algorithm that determines the alignment transformation, we have presented two experiments. In the first experiment two surfaces of a complex curved surfaced object taken from different viewpoints are successfully registered. In the second experiment, known objects are successfully identified and located in a scene.

Acknowledgements

The work presented in this paper was funded by a UK EPSRC grant GR/H86905. The author would also like to thank Dr. Peter Rockett for the derivation of the Bhattacharyya distance presented in Appendix A.

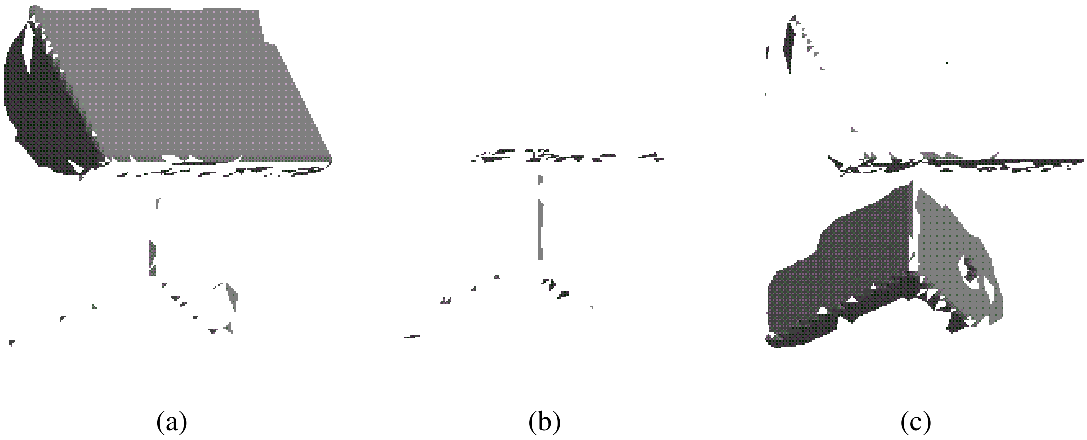


Fig. 7. (a),(b) & (c) present the scene facets which best match facets in Models 1, 2 & 3 respectively.

A Derivation of the Similarity Metric

In this section the derivation of a statistical metric for comparing binned measurements is presented. Given a random variable X, a statistical measure of the distance D between the endpoints $X = x$ and $X = x + \delta x$ of a short line is obtained by normalising by the standard deviation σ.

$$D = \frac{\delta x}{\sigma} \tag{2}$$

In general then, the statistical distance between any two points $X = s$ and $X = m$ can be determined by the definite integral:

$$D = \int_{s}^{m} \frac{dx}{\sigma} \tag{3}$$

For N independent measurements the statistical distance is given by a sum of squared components:

$$D^2 = \sum_i (\int_{s_i}^{m_i} \frac{dx_i}{\sigma_i})^2 \tag{4}$$

It is well known that binned data conforms to a Poisson distribution and that the variance of a Poisson variable is equal to its mean. A statistical distance metric for binned data is then obtained by substitution of $\sigma_i = \sqrt{x_i}$.

$$D^2 = \sum_i (\int_{s_i}^{m_i} \frac{dx_i}{\sqrt{x_i}})^2 \tag{5}$$

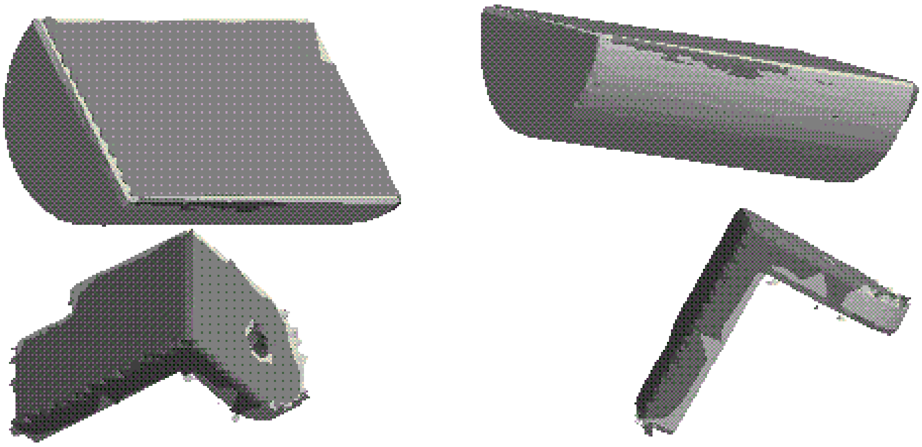


Fig. 8. The identification and localisation of the two objects in the scene. The scene data is presented in the darker shade and the models in the lighter shade. The second image presents the scene from a different view-point.

$$= 4 \sum_i (\sqrt{s_i} - \sqrt{m_i})^2 \tag{6}$$

Removing the constant factor in this expression gives the statistical metric proposed by Matusita [10] which is known as the Matusita distance.

$$D_{matusita} = \sum_i (\sqrt{s_i} - \sqrt{m_i})^2 \tag{7}$$

Expanding this expression gives:

$$D_{matusita} = \sum_i s_i + \sum_i m_i - \sum_i \sqrt{s_i}\sqrt{m_i} \tag{8}$$

If both m and s are normalised, or when using this metric to compare a single *scene* histogram with a set of normalised *model* histograms, this is simply:

$$D_{matusita} = const - \sum_i \sqrt{s_i}\sqrt{m_i} \tag{9}$$

Removing the constant results in the Bhattacharyya distance.

$$D_{bhattacharyya} = \sum_i \sqrt{s_i}\sqrt{m_i} \tag{10}$$

References

1. Bergevin, R., Laurendeau, D. and Poussart, D., “Registering Range Views of Multipart Objects”, CVIU, 61(1), pp1-16, 1995.
2. Besl, P. J. and McKay, N. D., ”A method for registration of 3-D shapes”, IEEE PAMI, 14(2), pp 239-256, 1992.
3. Eggert, D., Fitzgibbon, A. W. and Fisher, R. B., “Simultaneous registration of multiple range views for use in reverse engineering”, Proc. ICPR96, pp243-247, Vienna, 1996.
4. Evans, A. C., Thacker, N. A. and Mayhew, J. E. W., “The Use of Geometric Histograms for Model-Based Object Recognition”, Proc. BMVC93, pp429, 1993.
5. Faugeras, O. D. and Hebert, M., “A 3-D Recognition and Positioning Algorithm using Geometric Matching between Primitive Surfaces”, Proc. 8th IJCAI, pp-996-1002, 1983.
6. Fisher, R. B., “From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis”, John Wiley & Sons, 1989.
7. Garland, M. and Heckbert, P. S., “Surface Simplification using Quadric Error Metrics”, SIGGRAPH97, pp209-216, 1997.
8. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., “Surface Reconstruction from Unorganised Points”, Computer Graphics, 26(2), pp71-78, 1992.
9. Johnson, A. E. and Hebert, M., “Recognizing Objects by Matching Oriented Points”, Proc. CVPR97, pp684-689, 1997.
10. Matusita, K., “Decision Rules Based on Distance for Problems of Fit, Two Samples and Estimation”, Ann. Mathematical Statistics, Vol. 26, pp. 631-641, 1955.
11. Stephens, R. S., “A Probabilistic Approach to the Hough Transform”, Proc. BMVC90, pp55-59, 1990.
12. Thacker, N. A., Aherne, F. J. and Rockett, P. I., “The Bhattacharyya Metric as an Absolute Similarity Measure for Frequency Coded Data”, STIPR97, 1st International Workshop on Statistical Techniques in Pattern Recognition, Prague, Czech Republic, 1997.
13. Thirion, J., “New Feature Points based on Geometric Invariants for 3D Image Registration”, IJCV, 18(2), pp121-137, 1996.

A Functional-Based Segmentation of Human Body Scans in Arbitrary Postures

Naoufel Werghi, *Member, IEEE*, Yijun Xiao, and Jan Paul Siebert, *Member, IEEE*

***Abstract*—This paper presents a general framework that aims to address the task of segmenting three-dimensional (3-D) scan data representing the human form into subsets which correspond to functional human body parts. Such a task is challenging due to the articulated and deformable nature of the human body. A salient feature of this framework is that it is able to cope with various body postures and is in addition robust to noise, holes, irregular sampling and rigid transformations. Although whole human body scanners are now capable of routinely capturing the shape of the whole body in machine readable format, they have not yet realized their potential to provide automatic extraction of key body measurements. Automated production of anthropometric databases is a prerequisite to satisfying the needs of certain industrial sectors (e.g., the clothing industry). This implies that in order to extract specific measurements of interest, whole body 3-D scan data must be segmented by machine into subsets corresponding to functional human body parts. However, previously reported attempts at automating the segmentation process suffer from various limitations, such as being restricted to a standard specific posture and being vulnerable to scan data artifacts. Our human body segmentation algorithm advances the state of the art to overcome the above limitations and we present experimental results obtained using both real and synthetic data that confirm the validity, effectiveness, and robustness of our approach.**

***Index Terms*—Human body shape analysis, Reeb graph, scattered 3-D range data segmentation, 3-D surface topology, 3-D whole-body scanners.**

I. Introduction

THE PAST FEW years have witnessed the emergence of three-dimensional (3-D) imaging technology that enables full scanning of the human body (HB) surface with reasonable measurement accuracy as well as at an acceptable computational cost. This advance facilitates the exploitation of the HB form in various areas such as anthropometrical research [1]–[3], clothing design [4]–[6], and virtual human animation [7], [8]. Although the raw data generated by the HB scanner requires substantial main memory and storage resources, it contains little semantic information. To achieve effective and efficient use of body scan data, it is often necessary to partition the whole scan data set into subsets corresponding to the principal body parts. This segmentation provides the basis for a high-level representation of the scan data and is a prerequisite for further semantic analysis. For example, in medical applications, the segmentation process provides an Atlas for extracting data belonging to limbs that can then be used to support further analysis such as fitting generic limb models. These models can then be used to automate specific clinical protocols, such as spinal curvature assessment. Applications dealing with the estimation of HB motion from range image sequences [9], can exploit scan data segmentation when initializing the parameters of a HB tracking algorithm. HB scan segmentation is also useful for online garment shopping applications [5], [6] as it can contribute toward providing accurate body measurements and sizing.

Many attempts to devise a framework for the segmentation of objects that are human-like in shape have been reported in the literature [10]–[13]. Most of this previous work is based on contour-based segmentation techniques whereby points of discontinuity in the range data are first detected and then dynamically grouped into contours using various techniques, such as energy-minimization, when processing deformable curves [14]. However, automatic segmentation of real HB data is a more challenging problem, firstly because the body shape is both articulated and deformable and secondly because the scan data is by nature nonuniformly sampled and may exhibit gaps and may be corrupted by noise. It was therefore necessary to explore new techniques in order to formulate approaches that would be better able to cope with these challenges. In his pioneering work, Nurre [15] approximated the body structure by a stick-template representing the head, the two arms, the two legs and the torso. His goal was to segment the body into six segments corresponding to these parts. This approach combines a global shape description, namely moments analysis, and local criteria of proximity which are derived from a priori knowledge of the relative positions of the body parts in the standard posture (standing body with arms held at the sides). The range data is organized into slices of data points. The horizontal slices are stacked vertically and the data points are assigned to the different body parts according to the slice's topology and its position in the body. While this work achieved considerable progress toward the automatic decomposition of HB scan data, it has been criticized for imposing the requirement to limit body poses to a strict standard posture and for its lack of robustness against noise, gaps in the data, and variations in shape and posture of the HB. There have been many subsequent attempts to improve Nurre's approach. For example, Decker *et al.* [16] improved the localization of the key landmarks of the HB by applying differential operations on slice shape attributes. Although a degree of improvement resulted, this

Manuscript received June 21, 2004; revised February 9, 2005. This work was supported by the U.K. EPSRC under Grant RG 533801 and by the U.K. Imaging Faraday Partnership. This paper was recommended by Associate Editor D Goldgof.

N. Werghi is with the College of Information Technology, Dubai University College, United Arab Emirates (e-mail: nwerghi@duc.ac.ae).

Y. Xiao and J. P. Siebert are with the Department of Computing Sciences, University of Glasgow, Glasgow, U.K. (e-mail: yjxiao@dcs.gla.ac.uk; psiebert@dcs.gla.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2005.854503

approach could not remedy the limitations of Nurre's approach. Recently, Wang *et al.* [17] proposed a new approach based on a Fuzzy logic framework, however, this again was restricted to standing postures. Their segmentation technique involved local curvature analysis of the slices and operates on mesh data that has undergone several preprocessing stages. The overall performance of this approach remains identical to that of Nurre.

From the above it is evident that the approaches developed so far are restricted by their underlying assumptions, and none of these has been able to overcome the standard posture restriction. Furthermore, most of these approaches suffer from instability when applied within real applications that must cope with noisy and corrupted 3-D HB scan data. In addition, to date no evidence citing the repeatability of these previous algorithms has been reported in the literature. By definition, HB scan data segmentation must be of practical utility, it must be robust to variation in the body surface shape stemming from biological factors such as age, genetics, etc. It must also cope with changes of body posture as well as with the diversity of the scan data sources. While *ad hoc* techniques might work for special cases, they cannot address the above stringent requirements. In a recent publication [18], the authors presented an approach that successfully addressed some of the previously discussed issues. However this approach can only deal with moderate variations around the standard posture.

In this paper we propose a general topological analysis framework that offers a systematic way to segment HB body data. The salient feature of this framework is that it can cope with body shape variations, posture changes, rigid transformations and diverse sources of scan data. Furthermore, our approach does not require any pre-processing stages, operates on 3-D point-cloud data, and does not rely on local feature analysis, which would be vulnerable to deficiencies in the scan data.

The novel aspects of this paper are 1) the extension of the Reeb graph concept to sets of scattered data points, which represents an extension to the work of Biasotti *et al.* [19] which explores the use of Reeb graph applied to polygonal meshes; 2) a simple and efficient technique for computing the geodesic distance map of HB shape (and generally of a three-dimensional shape) with respect to a source point; 3) a robust technique, for constructing the discrete Reeb graph (DRG), which can cope effectively with data deficiencies; and 4) a new functional surface segmentation method based on the concept of the DRG applied to the HB surface.

The remainder of this paper is organized as follows. Section II describes the theoretical foundations of our approach, namely Morse theory, the Reeb graph and geodesic distance. Section III describes the implementation of the segmentation approach and the mechanisms involved. Section IV validates our proposed framework via experimental results. Finally, in Section V, we provide a summary of the main findings of this paper and make suggestions for future research.

II. Morse Theory and the Reeb Graph

Morse theory can be thought of as generating the classical theory of critical points (maxima, minima and saddle points) of smooth functions on a smooth manifold. Specifically Morse theory states that for a generic function defined on a closed compact manifold, the nature of its critical points determines the topology of the manifold. Morse functions are generic functions whose critical points are nondegenerate (Hessian matrix of the function at the critical point is nonsingular). For a Morse function, the critical points determine the homology groups of the manifold, which in turn fully describe its topology. Moreover the way the manifold is embedded can be coded using a Reeb graph, as proposed by Reeb [20] to represent the evolution and arrangement of level-set curves on a manifold. A Reeb graph describes the configuration of and relationship between critical points and provides a way to understand the intrinsic topological structure of a shape. Morse theory and the Reeb graph have been used in many applications such as shape matching [21], shape coding [22]–[24], surface compression [25], volume visualization [26], terrain analysis [27] and 3-D skeletonization [28], [29]. The last publication cited is particularly close to our work. We will describe and compare in detail the related approaches in Section III-F.

A Reeb graph can be defined as follows.

Definition 1: Let $\boldsymbol{f}$ be a real-valued function on a compact manifold M. The Reeb graph of $\boldsymbol{f}$ is the quotient space of the graph of $\boldsymbol{f}$ in M by the equivalence relation "$\sim$" defined by $(X_1, \boldsymbol{f}(X_1)) \sim (X_2, \boldsymbol{f}(X_2))$ if $\boldsymbol{f}(X_1) = \boldsymbol{f}(X_2)$ and X_1 and X_2 are in the same connected component of $\boldsymbol{f}^{-1}((\boldsymbol{f}(X_1))$.

Roughly speaking, the two pairs $(X_1, \boldsymbol{f}(X_1))$ and $(X_2, \boldsymbol{f}(X_2))$ are represented as the same element in the Reeb graph if the values of $\boldsymbol{f}$ are the same and if they belong to the same connected component of the inverse image of $\boldsymbol{f}(X_1)$ or $\boldsymbol{f}(X_2)$. Actually one element in the Reeb graph of a compact manifold represents all points having the same value under a real function. The Reeb graph is a representation of the evolution and arrangement of these groups of points, also called level-sets. Fig. 1(a) illustrates an example of a Reeb graph for a torus. The function $\boldsymbol{f}$ is the "height" function which here simply returns the value of the coordinate z of a point X. In the corresponding Reeb graph, a point in a branch represents isovalued and connected points on the manifold. From bottom to top, the level-sets on the torus expand, split, merge and then become smaller. The Reeb graph gives an intuitive description of the evolution of level-sets, where diamond points denote the level-sets passing through saddle points on the torus. By applying the Reeb graph to a HB, we get a tree-like representation as illustrated in Fig. 1(b). Extremal points lie on the head top, hand tips and the bottom of the feet. Saddle points are located at the armpits and groin. Moreover, the branches in the Reeb graph reflect the body parts of the human figure, i.e., arms, legs, torso, and head. Therefore, if we succeed in retrieving the level-sets in these branches, we can partition the input point-cloud data into sub-sets approximately corresponding to the body parts of the human shape.

A. The Morse Function

For standard postures such the one shown in Fig. 1(b), where the human figure stands in the measuring platform with arms held at the sides and legs separated, the simple height function $h(X) = z$ that returns the z coordinate of a point X, is an optimal choice because the orientations of the body parts in such

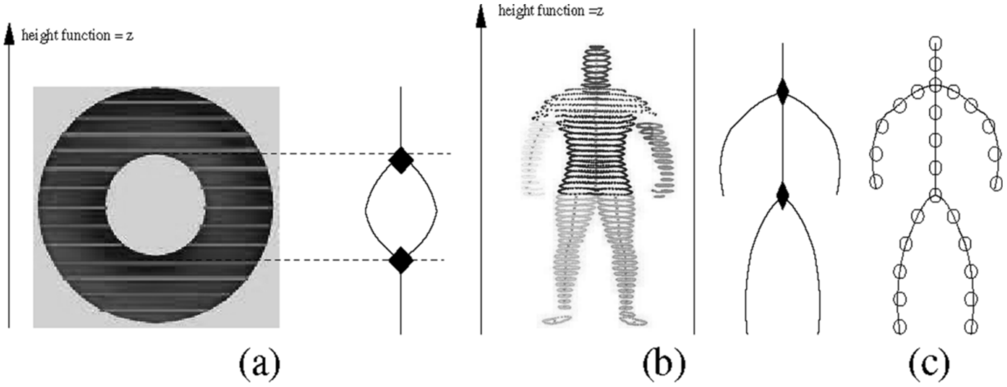


Fig. 1. (a) Reeb graph of a torus. (b) Reeb graph of a HB shape. (c) Discrete Reeb graph.

postures are orthogonal to the cross-sectional planes inferred by the height function. For nonstandard postures where the human figure is not constrained to be in a standing position, both the arms and legs as well as the whole body can have arbitrary orientations. The only constraint is that the body limbs must be separated from each other everywhere except at their joints. For such postures, the simple height function cannot guarantee a Reeb graph faithful to the HB anatomy. This limitation emanates from the fact that the height function is not invariant under rigid transformation or under deformation inferred by whole body transformation or body part movement. Therefore, to be capable of handling nonstandard postures, it is imperative that the Morse function be invariant under these transformations, i.e., a function that keeps the same value as long as the topology of the surface is preserved. The curvature function could be a candidate as it is invariant under rotation and only very slightly affected by body movements. However, it is very sensitive to noise and data deficiencies that would lead to highly unstable Reeb graph structures. Therefore, it is not appropriate for our application which must handle scattered noisy scan data that might be corrupted by many deficiencies such as holes, gaps and nonuniform sampling. To find a function that overcomes these problems, we employed the geodesic distance [31] defined as the length of the shortest path connecting two points. The geodesic distance is invariant to the rotations and transformations produced by body movements and is resistant to data corruption and perturbation. Thus the geodesic distance metric underpins a stable Reeb graph with respect to these aspects. The function defined as

$$\rho_S(X) = \mathrm{gd}(X, S) \tag{1}$$

where $\mathrm{gd}(X, S)$ returns the geodesic distance from a point X to a source point S, is a reasonable candidate for a Morse function. However the location of the source point might affect, to a certain extent, the structure of the Reeb graph as will be shown later.

Another candidate for a Morse function is the function defined as

$$\sigma(X) = \int_{Y \in \mathrm{surface}} \mathrm{gd}(X, Y) d_{\mathrm{surface}}. \tag{2}$$

This function represents the sum of the geodesic distances between the point X and all the points on the body surface. In addition to being resistant to geometric transformations and deformations produced by body movements, this function is not related to any source point and therefore guarantees a stable Reeb graph. Intuitively, the σ function presents low values for points located at the center of the body (torso area), for which the distance to other points is relatively small, and high values for points located at the body extremities. On the other hand, this function is computationally expensive when compared to $\rho_S(x)$ as it will be described in Section III-A.

B. The DRG

Classical Morse theory is concerned with only nondegenerate critical points of smooth functions (Morse functions) on smooth manifolds. The notion of the Reeb graph in its standard form is defined with respect to smooth and continuous surfaces. There have been several successful attempts to extend the Reeb graph to discrete surfaces e.g., [19], but the surface data is required to be organized into polygonal meshes. In practice, our data format does not comply with this requirement, as it consists of an unorganized cloud of 3-D data points which have been corrupted by noise, gaps and nonuniform sampling. We present a Reeb-graph extraction technique that is compatible with this type of data. For clarification of terminology, hereafter we refer to Reeb graph extracted from such data as the DRG to distinguish it from the classical Reeb graph. The concept of the DRG is described in the following definitions.:

Definition 2 (Connectivity of Point Sets): Two point sets $P = \{p_i\}, i = 1 \cdots m$ and $Q = \{q_j\}, j = 1 \ldots n$ are defined as connected if $\exists p_i \in P$ and $\exists q_j \in Q$ such that $|p_i - q_j| \leq \tau$.

$|p_i - q_j|$ denotes the distance between points p_i and q_j and τ is a given threshold. The above definition also holds for the connectedness between two points for the particular case where the sets P and Q contain a single point each.

Definition 3 (Connective Point Set): A point set C is connective if $\forall$ subset $\Omega \subset C$ and $\Omega \neq \emptyset, \Omega$ and $\bar{\Omega}$ are connected. Here $\bar{\Omega}$ denotes the complement of Ω in C. Definition 3 defines a "tight" point set in which all the points are connected.

Definition 4 (Level-Set Curve): A level-set curve is an iso-valued connective point set, that is a group of points, that share the same Morse function value, and which forms a connective point set.

Dentition 5: A Discrete Reeb graph is a non-oriented two-dimensional (2-D) graph, where a node represents a level-set curve and where an edge represents a connection (in the sense of Definition 2) between two adjacent level-set curves.

Based on the above definitions, the construction of the DRG involves the following tasks.

Step 1: Establishing Level-Sets: Level-sets are groups $\mathcal{G}_i, i = 1 \ldots M$ of iso-valued data points defined as $\mathcal{G}_i = \{X, f(X) \in [v_i, v_{i+1}]\}$, where f is the Morse function and $v_i, i = 1 \ldots M + 1$ is a set of discrete equidistant values.

Step 2: Establishing Level-Set Curves: Each level-set is decomposed into a group of level-set curves, using the criteria outlined in Definition 3, and where the threshold τ is set to d which stands for the minimum distance between a pair of points.

Step 3: Building the Connectivity Between Nodes: Two nodes in two adjacent level sets ($\mathcal{G}_i$ and $\mathcal{G}_{i+1}$) are linked if their corresponding level-set curves are connected, according to Definition 2. However, the related threshold τ is set to βd, where β is a parameter used to tune the precision of the connection.

By following the above steps, we can construct progressively a graph containing all the nodes and their associated links. Fig. 1(c) depicts a DRG of the HB shape. This graph has the appearance of a discrete version of the graph in Fig. 1(b), where the continuous branches are replaced with successively linked nodes. The DRG extends the concept of Reeb graph to discrete surfaces, and thus permits topological analysis of scan data.

III. The Segmentation

The segmentation process involves three tasks, namely: 1) computing the Morse function; 2) extracting the level-sets; 3) decomposing these level-sets into connected level-set curves; and 4) extracting the different branches. Task 3 in essence comprises DRG construction. The implementation of these tasks within the segmentation process and the complexity of the overall algorithm depend on the adopted Morse function. When the simple height function is used (i.e., when dealing with standard postures), the four tasks are carried out within a single stage in one pass algorithm. When the ρ_S function is used, tasks 1 and 2 are performed simultaneously. Alternatively, when the σ function is employed the four tasks are executed consecutively. The following sections will shed light on these aspects.

A. Computation of the Morse Function ρ_s and σ

Both Morse functions ρ_S and σ involve the computation of geodesic distances. In the literature, Dijkstra's algorithm [30] has been the most popular tool for computing geodesic distances between a group of points and a source point. In addition, it provides the path from any point to the source point. However, this algorithm implies a significant computational cost. Indeed, it requires that the group of points be organized in a graph, where a node is associated to each point and edge represents a connection between pair of points, according to a proximity criterion. The construction of such graph infers a computational complexity that can go up $O(N^2)$, where N is the number of points. Dijkstra's algorithm itself, in its optimal implementation, infers a complexity of $O(E + N\text{Log}(N))$, where E is the number of edges in the graph. For these reasons and because finding the paths to a source point is not required in our application, we preferred not to use this algorithm. Instead, we developed an efficient algorithm based on a wavefront propagation technique. It is based on the following principle: Given a wave centered on a manifold, then all the points on the wavefront have the same geodesic distance to the wave center and thus form a level-set. Our wavefront propagation algorithm operates on a binary voxel grid since it is easy to define a neighborhood in voxel space and to then traverse connected voxels. Due to these well-behaved properties, wavefront-propagation on a voxel grid can have a very simple mathematical form as follows:

$$\begin{cases} W_0 = \{v_s\} \\ W_{i+1} = (W_i \oplus e - (W_i \oplus e) \bigcap S_i) \bigcap \bar{S}_i \end{cases}$$

where W_i is the wavefront generated on the ith iteration of the algorithm; v_s is the source voxel; $S_i = \sum_{j=0}^{i} W_j$. $\bar{S}_i$ is the complement set of S_i. $\oplus$ denotes the morphological dilation operator and $\mathbf{e}$ is a $3 \times 3 \times 3$ structuring element composed of 27 1-valued voxels. The wavefront starting position is located at the source voxel associated with the source point and the wavefront then iteratively spreads on the voxelised surface from this location. In each iteration, the wavefront is the level-set containing voxels of the same geodesic distance to the source point. The attractive aspect of this technique is that it simultaneously extracts the level-sets while it computes the ρ_S function. It is easy to prove that the computational complexity in each iteration is $O(n_i)$, where n_i is the number of voxels in W_i. Therefore, the complexity of the whole algorithm is $O(N)$, where N is the number of all 1-valued voxels. This linear complexity allows efficient calculation of geodesic distances and level-sets. Therefore the computation of the function ρ_S is carried out by simply applying the above algorithm after selecting a source point. Fig. 2(a) illustrates wavefront-propagation applied on a simple ellipsoid surface. Row 1 in Fig. 2 depicts grey level mapping of the ρ_s function, related to a posture instance, and corresponding to different locations of the source point, namely the head (column f), the torso(columns g, h, i, and j), the knee (column k), and the hand (column l). The grey level varies from white (which corresponds to the minimum value, at the source point) to the black (largest value). Row 2 depicts the corresponding level-sets. We can observe that while the level-set orientations follow the body limbs in all cases, they do present some dissimilarities, particularly at the junction areas.

The computation of the σ function is more complex as it involves for each data point the sum of geodesic distances from that point to all the points in the body surface. The discrete approximation of σ in (2) is

$$\sigma(X) = \sum_{i}^{N} \text{gd}(X_i, X) \tag{3}$$

Using (1), the above equation can be rewritten as $\sigma(X) = \sum_i^N \rho_X(X_i)$. This indicates that the computation of σ at a given point requires computing the function ρ for all the 1-valued voxels. Making thus the complexity of the whole algorithm that calculates the σ function for all the points to be $O(N^2)$.

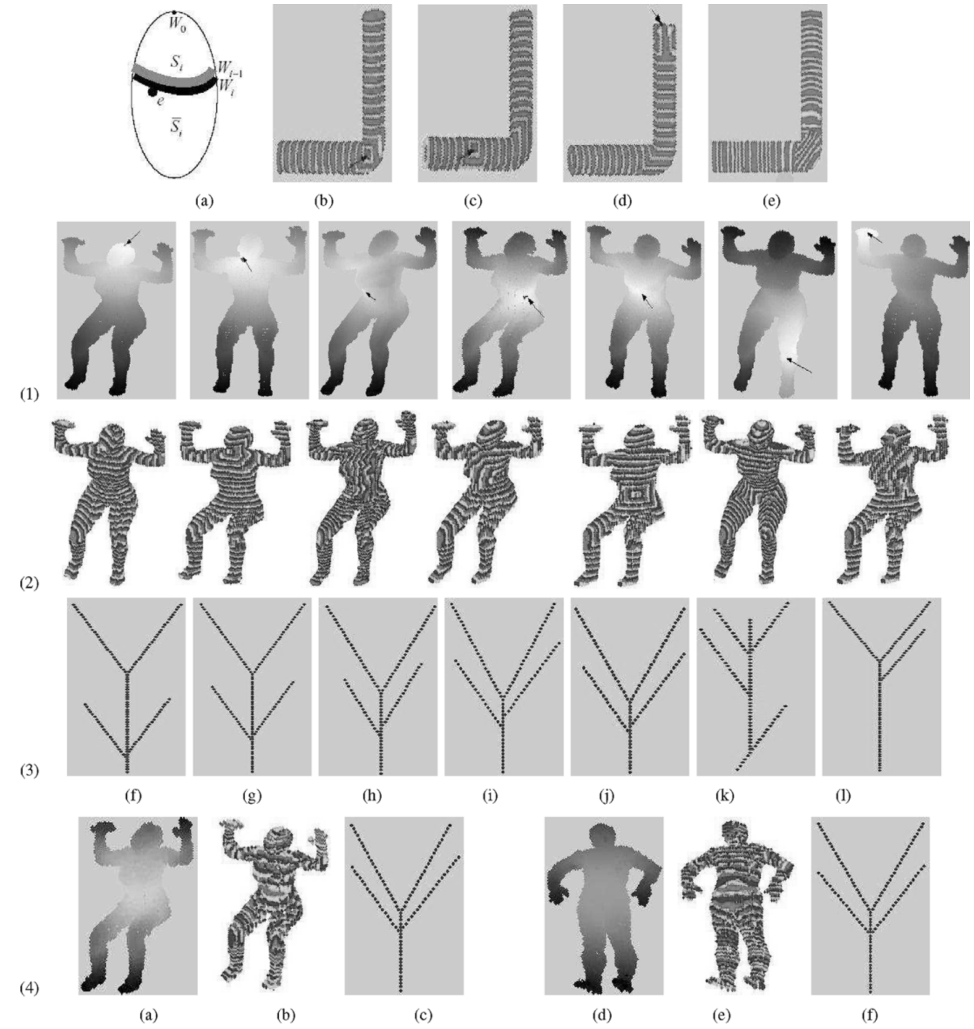


Fig. 2. (a): Wavefront propagation on a simple elliptic surface. (b), (c), (d) Level-sets of ρ_S function, related to an L-Shaped object, and corresponding to source points located at the junction, the middle and the extremity of a branch respectively. (e) Level-sets of σ function. Row 1: grey level mapping of the ρ_S function for different source point locations (indicated by the arrows). Rows 2 and 3: corresponding level-sets' patterns and DRGs, respectively. Row 4: grey level mappings of the σ function, the corresponding levels-sets' patterns and DRGs related to two different postures.

B. Source Point Location and Its Effect on the Level-Sets and the DRG

While the ρ_S exhibits nice properties in terms of efficiency and ease of implementation, an effective deployment of this function for constructing the DRG, and for performing the separation between branches, depends to some extent on the location of the source point however. Ideally, the use of geodesic distance would permits us to extract level-sets that maintain a consistent orientation relative to the HB pose. Locally, this would result in the recovery of "slices" parallel to the principal axes of human limbs. Unfortunately, this desirable behavior is compromised at areas comprising junctions (and also within the neighborhood of the source point) and the patterns resulting from such corruption are dependent on the location of the source point itself. Fig. 2(b), (c), and (d) illustrates this effect on a simple L-shaped object, showing the different level-sets that are produced by the ρ_S function corresponding to three different source points located at respectively: the junction area, the middle, and the extremity of one branch. By observing their corresponding patterns at junction areas clear dissimilarities become apparent. The disparity between the level-set's behaviors in the three cases makes their decomposition into subsets, and the correspondence of these subsets to the their associated branches, unlikely to result in an identical separation between these branches. For the purpose of comparison, we show in Fig. 2(e) level-sets of the σ function for the same object. We notice that the level-sets, seemingly behave like those related to the junction-located source point case. Indeed, both of them exhibit a degree of symmetry at the junction area.

Regarding the reconstruction of the DRG, ideally, we would like to have a DRG structure that is similar to the "standard" HB DRG depicted in Fig. 1(c). This structure reflects the anatomy of the HB shape, and thereby facilitates the identification of the branches. However, the stability of this structure is guaranteed only for source points selected at the central area of the body, that is the torso-head area. Fig. 2 (row 3) illustrates this aspect. It shows a group of DRGs obtained for different source point locations. We can observe that for cases where the source point is located within the torso-head area a DRG structure is generated which is close to the standard form displayed in Fig. 1(c). On the other hand, in those cases where the source point is lo-

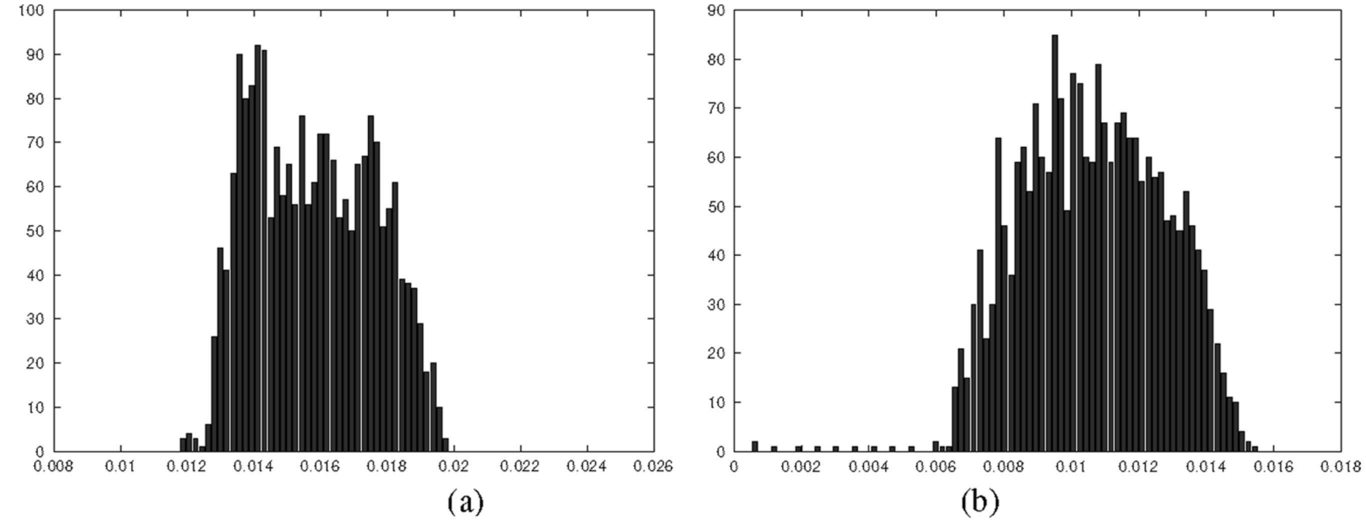


Fig. 3. Histogram of the distances between the most closest pairs of scan data points. (a) Cyberware scanner. (b) Wick & Wilson scanner.

cated at the hand and the knee, the recovered DRG appears to have become corrupted. These observations suggest that in order to obtain a standard DRG the source point must be selected at the neighborhood of the medial axis of the body (i.e., the torso axis). While we do not have a theoretical basis for this hypothesis, we believe that it is linked to the fact that the satisfaction of that condition infers a symmetric distribution of the ρ_S function with respect to the medial axis of the body. As the grey level mappings show, in Fig. 2(row1), cases exhibiting a symmetric distribution of the ρ_S function induce a standard form of DRG, as opposed to cases where the symmetry of the ρ_S function is severely violated (row 1, columns k and l, for source points located at the knee and the hand).

In contrast to the ρ_S function, the σ function which by definition has no dependance on a source point, exhibits a centered and a symmetric distribution (Fig. 2, row 4, a and d) from which it is possible to infer a stable DRG as shown in (Fig. 2, row 4, c and f). Therefore, when a ρ_S function is employed, a suitable source point can then be obtained via manual selection or automatically using simple heuristic methods. One plausible method is the following: 1) compute the ρ_S function for an arbitrary source point; 2) search for the point at which ρ_S is closest to the average value; and 3) repeat 1) and 2) until the location of the source point converges. The point of convergence will be then located nearby the geodesic center of mass, in the torso-head area, and can therefore be selected as a suitable source point.

C. Threshold Setting

Setting the appropriate range of the threshold d involved in step 2 and step 3 of the DRG construction should be performed with care since large values might introduce "short-circuit" edges in the DRG while small values can render the graph excessively sparse. In either case, the topological integrity of the HB shape might not be preserved. In our application, however, since the global geometry of the HB shape is known, some constraints can be used in the second case to eliminate false segments (some criteria related to this aspect are proposed in Section III-D1). Therefore we have to care only about the maximum value allowed for the threshold. For this purpose we used a practical approach which consists of first estimating the resolution ϵ of the scan data. This was conducted as follows. We determined the set of closest pairs of data points (i.e., nearest neighbor tuples) over a large area of the scanned data. Then we set the resolution to the weighted average deduced from the related distance histogram (the term distance here refers to the distance between the pairs of points). Fig. 3 shows two distance histograms corresponding to two portions of scan surface obtained from two HB scanners, namely a Cyberware scanner and a Wicks & Wilson scanner (these scanners are discussed in Section IV). The corresponding resolutions are $\epsilon = 1.6$ mm and $\epsilon = 1.1$ mm, respectively. The expression of the threshold can then be defined as $d = K\epsilon$, with the minimum value of K being set to 2. This value would normally lead to the most precise segmentation, however at the expense of increased computational time. A larger value of K, might reduce the accuracy of the segmentation, though our experiments (Section IV) showed that a reasonable topology-preserving segmentation can be obtained with up to $K = 7$, as long as the the separation between the body parts is larger than the threshold d.

D. DRG Construction

The scan data is first organized into a voxel grid. The size of voxel is $d \times d \times d$ where d is the threshold outlined in Step 2. When the height function is adopted (i.e., when dealing with standard postures), a group of iso-valued data points comprising a "slice" of points is obtained by intersecting a plane of a certain height with the body surface. When dealing with nonstandard postures, the ρ_S or σ is computed for each point in the voxel grid. For the ρ_S function, the level-sets are implicitly extracted in this stage as described in Section III-A, whereas for the σ function they are extracted subsequently. Each level-set of data points is then decomposed into level-set curves and the DRG is constructed according to Steps 2 and 3 described in Section II-B.

1) DRG Analysis and Branch Extraction: In this stage, the DRG is analyzed to detect critical nodes and to extract the branches corresponding to body parts. For the sake of clarity we shall describe the methodology for the case of a standard posture where the height function is employed. The general

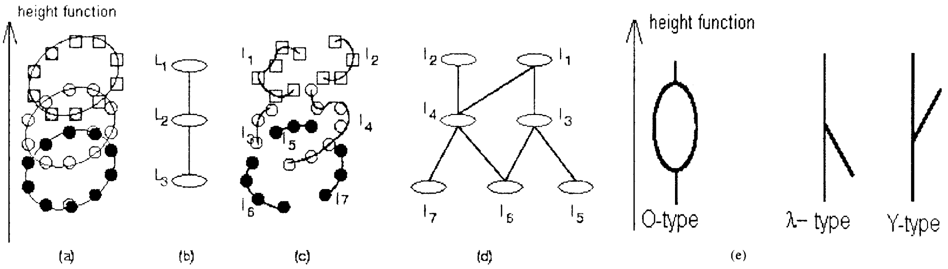


Fig. 4. (a) Three slices representing a portion of clean data. Each slice contains a single level-set curve (group of linked points), thus each slice is mapped into a single node in the DRG (b). (c) Data corruption causes each slice to decompose into several level-set curves. This results in a disorganized graph (d). In this example, the original nodes L_1, L_2, and L_3 have degenerated respectively into $(l_1, l_2), (l_3, l_4)$ and (l_5, l_6, l_7). (e): Three primary patterns in DRG.

principles of the methodology remain valid for the other Morse functions and we shall point out any dissimilarities where these arise.

While it appears to be straightforward to detect the critical nodes in a DRG related to clean and well organized scan data [such as that depicted in Fig. 1(c)], this task is not trivial for real scan data. The data deficiencies (noise, holes and gaps) produce topological disturbances that lead to "false" critical nodes. This results in a corrupted tree structure as illustrated in Fig. 4. For simplicity, the example shown in this figure covers only a portion of three adjacent slices. In "clean" data, each slice consists of an organized set of connected points 4(a). Thus, each slice represents a single level-set curve. This will result in a DRG composed of three nodes 4(b). For real data, a slice might be composed of several level-set curves because of data corruption, leading thus to several nodes per slice. The example in Fig. 4(c) shows three level-set curves for slice 1 and two level-set curves for each of slice 2 and slice 3. The three initial nodes L_1, L_2, L_3 have degenerated into seven nodes $l_1, l_2, \ldots, l_7$. Setting the connections amongst the nodes as the final step of the algorithm (as described in step 3, Section II–B) leads to a disorganized graph [4(d)] that results, at the scale of the whole scan, in the DRG degenerating into a chaotic graph. The challenge therefore is to be able to recover the topological structure of the measured HB from such a corrupted graph. This problem is tackled as follows: Firstly, the nodes in the DRG are arranged level-by-level, and a link can only exist between two nodes in adjacent levels. This property cannot be damaged by the decomposition of nodes. Secondly, we have identified three primary topological configurations that appear in the DRG and have termed these as O-type, λ-type and Y-type patterns respectively [Fig. 4(e)]. For example, the group of nodes $(l_7, l_4, l_2, l_1), (l_6, l_4, l_1, l_3)$, and (l_6, l_5, l_3, l_1) represent a Y-type, an O-type and a λ-type, respectively. O-type patterns comprise two saddle nodes connected by two branches and this pattern reflects data anomalies (wholes, gaps) because the topology of the human form cannot produce such a topological configuration. Indeed, this argument also applies to Y-type patterns since a standard posture cannot generate a Y-type configuration. Therefore, O-t and Y-type occurrences in the DRG originate only as a result of deficient data. The λ-type and Y-type patterns are topologically identical however, since each comprises one saddle node and one leaf branch. Therefore, in order to distinguish between them, we introduce topographic information, namely, the direction of the height function, to allow a down/up (λ-type/Y-type) leaf branch categorization. Given the preceding, if O-types and Y-type patterns occur in the DRG, we know that they are caused by data corruption. In order to distinguish genuine λ-type patterns from false ones that have been produced by data deficiencies, we assume that the size of a leaf branch associated to a true λ-type pattern must not be smaller than the size of the smallest body part. This assumption is reasonable since the scan data is unlikely to contain gaps or holes at the scale of the body limbs, as such cases can be easily prevented by some form of quality control during the data acquisition phase.

Based on these considerations, the following criteria are used to identify "true" λ-type patterns which comprises "true" branches and saddle nodes: 1) a 'true' branch is downward and must satisfy: $|h_{\max}(\text{branch}) - h_{\min}(\text{branch})| > h_{\lim}$ where $h_{\max}/h_{\min}(\text{branch})$ denotes the maximum/minimum value of a branch in the height direction, and $h_{\lim}$ is a threshold that represents the minimum length allowed for a branch and 2) a "true" saddle node has at least two branches.

The strategy followed to reject false critical nodes and identify 'true' branches for nonstandard postures remains the same except that the categories of patterns in the DRG will be reduced to only two types, namely the O-type (caused by data noise) and either the λ-type or the Y-type, depending on which orientation the level sets are constructed. We choose to construct the level-sets in the orientation of the increasing geodesic distances, thus allowing only O-type and Y-type patterns in the DRG. Therefore, the branch orientation in criterion 1 is amended to upward and the Morse function is set to the ρ_S function or σ function depending on which one is adopted. Criterion 2 remains unchanged, however.

E. The Segmentation Algorithm

The very constrained configuration of the HB in standard-postures and the simplicity of the height function in terms of computation permit the simultaneously extraction of level-sets with the construction and analysis of the DRG to extract the branches. The segmentation algorithm, named here Algorithm 1 (shown below), contains only a one-pass search from the bottom to the top of the scan data. In this pass, the critical nodes representing the bottom of the feet, groin, hand tips, armpits and head top are detected, and the 'true' branches between these critical nodes are extracted. The identification of branches corresponding to the body parts then becomes very simple, whereby the branches between the groin and bottom of the feet correspond to legs and the branches between armpits and hand tips correspond to the arms. The reminder of the data corresponds to torso and head. For nonstandard postures, the Morse function and the level-sets are first computed (in one stage for the ρ_S function and in two stages for the σ function) and thereafter, Algorithm 1 is applied (after setting the 'true' orientation of the branches to up) to extract the critical nodes and the true branches.

Algorithm 1: Notation

Slice: A level-set

Node: A level-set curve in a slice.

Class: A group of connected nodes.

Class(Node): The Class containing the Node.

Branch: A 'true' branch connected to a 'true' saddle node.

f: The Morse function (height function for standard postures) and ρ_S function or σ function for nonstandard postures

Code:

```
For each slice
    Group data points into level-set curves
    For each level-set curve
        Assign a node to this level-set curve
        If it is the 1-st slice
            $\mathrm{Class(node)} := \{\mathrm{this\ node}\}$
        Else
            Extract the group of nodes $(l_1, l_2, \ldots, l_m)$, connected to node, from the previous slice
            If $m = 0$
                $\mathrm{Class(node)} := \{\mathrm{this\ node}\}$
            Else
                From $\{\mathrm{Class}(l_i), i = 1, 2, \ldots, m\}$ select the classes verifying:
                $\{C_j : |f_{\max}(C_j) - f_{\min}(C_j)| > h_{\lim},\ j = 1, 2, \ldots, n, (n \leq m)\}$
                If $n \geq 2$
                    For each $C_j$
                        $\mathrm{Branch}_k = C_j, \quad k := k + 1$
                    End For
                End If
                $\mathrm{Class(node)} = \{\mathrm{Class(node)}\} \bigcup \{\mathrm{Class}(l_1)\} \bigcup \cdots \bigcup \{\mathrm{Class}(l_m)\}\}$
            End If
        End If
    End for
End for
```

F. Summary of the Approach and Comparison With 3-D Skeletonization

TABLE I

STAGES AND TASKS INVOLVED IN THE DIFFERENT VERSIONS OF THE SEGMENTATION PROCESS. THE FIRST COLUMN INDICATES THE EMPLOYED MORSE FUNCTION. THE SECOND COLUMN LISTS THE NUMBER OF INFERRED STAGES. THE THIRD COLUMN INDICATES THE TASKS INVOLVED IN EACH STAGE

Morse function	Stages	Tasks
height	1	Level-sets extraction, DRG Construction and branch extraction
ρ_S	1	Computing ρ_S and level-sets extraction
	2	DRG construction
	3	Branch extraction
σ	1	Computing σ
	2	level-sets extraction
	3	DRG construction
	4	Branch extraction

The HB scan segmentation approach involves four tasks and the implementation depends on the adopted Morse function. Table I summarizes the different versions. Of the different tasks, the computation of the Morse function is the most time consuming. The height function implies the simplest implementation, because it does not infer any computation. This permits us to perform all of the tasks in a single stage. Conversely, the σ function requires the sequential execution of the four tasks, and represents the most costly implementation $(O(N^2))$. The computational time related to this function is of the order of several hours on a Pentium IV, 1.7 Ghz computer. The ρ_S function, exhibits a reasonably tolerable computational cost. It exhibits an overall complexity of O(N). Despite the fact that the σ infers a stable DRG, we prefer the ρ_S function, because of the considerable disparity in terms of computational cost. Furthermore, our experiments showed that the two versions are similar in performance. The presented algorithm operates on a voxel grid and the number of voxels containing data points is small compared with the number of raw data points in the scan. However, it is straightforward to recover segmented raw HB scan data points from the segmented voxel data structure.

The approach of Verroust and Lazarus [29], mentioned previously in this paper, is the closest to ours in terms of theoretical background. Although both approaches involve similar concepts, namely, geodesic distance and level-sets, there are several fundamental differences, namely the objective of our task and the complexity and robustness of our respective implementations. In the following section, we shall detail these aspects, while emphasizing the characteristics of our approach.

Firstly, regarding the task undertaken, the two approaches target different objectives, namely skeleton construction and functional-based segmentation. Regarding implementation, the skeletonization process in [29] involves five stages: 1) establishing a neighborhood graph, where nodes represent data points, and an edge between a pair of nodes represents a connection between the corresponding pair of points, according to the m-nearest points rule; 2) computing a geodesic graph out of the neighborhood graph—this graph is actually a tree composed of geodesic paths joining the data points to a source point. The Dijkstra algorithm is used to compute the geodesic paths as well as the geodesic distances between the data points and the source point; 3) extracting level-sets of isovalued points using the geodesic graph; 4) Partitioning each level-set into subsets corresponding to the different branches of the surface; and 5) computing the centroids of connected subsets in each level-set and joining them, via the geodesic graph, to construct the skeletal curves. The approach was implemented on powerful Indy machines. In this process the extraction of the level-sets goes through the construction of a neighborhood graph and the use of Dijkstra's algorithm. This results in a high computational complexity. Indeed, although the neighborhood graph algorithm is optimized, its complexity is estimated to be $O(nk^2)$, where n is the number of data points and k is the average of the number of points contained in the neighborhood of each point. The value of k is not small given the potentially very large number of points in the raw scan data. Dijkstra's algorithm has a complexity of $O(e + n\log(n))$ where e is the number of edges in the neighborhood graph. Proceeding as in [29], to determine the level-sets in our application, will induce an intolerable computational cost. Based on the preceding implementation, we estimate that to process one HB containing 13 000 points would require approximately 20 h to execute on a Pentium IV, 1.7-GHz computer. In the contrary, the ρ_S version of our approach where the extraction of the level-sets infers a complexity of O(N), N being the number of 1-valued voxels, takes around 70 s to achieve the segmentation of the whole scan. Yet the most important feature of our work with respect to [29] is its robustness. Indeed, it is not clear how the approach in [29] can cope with an irregular sampling distribution of data points, data corruption and deficiencies. Since in order to obtain a valid skeleton, faithful to the body anatomy, each obtained subset (stage 4 of the skeletonization process, mentioned above) must correspond to a branch and thereby: 1) contain a single connected component and 2) form a closed curve. The non satisfaction of condition 1, results in false branches in the skeleton. The authors in [29] showed an instance of this case related to a body scan, presenting false branches, at the level of the feet, caused by the irregular distribution of the data points and the presence of gaps. The violation of condition 2, causes the centroid to be shifted (stage 5) from its actual location at the branch axis implying thus a distorted curve. The resulting skeleton will not then reflect correctly the actual body template. Cases such as those depicted in Fig. 9(b) and (c) for instance, cannot be accommodated by the approach in [29]. Actually, the authors in [29] recognized that the validity of their approach is conditional upon having data points that are regularly distributed on the object surface. Our approach, on the contrary, copes rather effectively with different types of data corruption due to the mechanisms implemented in the DRG analysis that eliminate false branches while preserving correct connectivity. Regarding segmentation accuracy, it appears that operating on raw data points and the use of Dijkstra's algorithm as in [29] improves segmentation accuracy because this approach is able to extend the computation of the ρ_S to the interior of the edges of the geodesic graph by interpolating the endpoint values of the edge. This in turn permits a finer mapping of the ρ_S function. While this result appears to be attractive, as it allows the possibility of further refining the level-sets and improving the accuracy of the segmentation, it is also very likely that this benefit will be cancelled by the corruption of the level-sets at the junction areas of the body (see Section III-B). It must also be noted that this limitation applies equally to our approach. Nevertheless the established thresholds, namely $d = K\epsilon$, and βd, involved respectively in setting connections between pair of points and level-set curves, permit us to control to a limited extent the accuracy of the segmentation in our approach.

IV. Experiments

A series of experiments were conducted using real and synthetic data to test the validity of our approach. The performance and the robustness of the corresponding algorithms were assessed with respect to: 1) variation of the HB shape; 2) variation of scan source; 3) severe corruption of the scan data; and 4) variation of the HB posture.

Real-world HB scan data was collected from different sources, namely the Cyberware website [32], the CAESAR project website [3] and the HB scanner located within EdVEC, Edinburgh Virtual Reality Centre [34]. The scans of the first two sources were acquired by Cyberware whole body scanners WB4 [32]. This scanner, uses laser-based technology in which a laser beam is projected on the body. The beam profile is captured by different cameras around the body and 3-D points are then extracted from each camera view to be combined into a single 3-D point cloud representing the body surface. The acquisition time of this scanner is approximately 17 s. The set of collected scans contains 25 scans of different individuals including males and females (only 4 are shown in this paper). Each scan contains of the order of 13 000 data points. The third source is a Wicks & Wilson HB scanner [33], which is based on a Moire fringing technology where fringe patterns, projected onto the body surface, are captured by four cameras, with a moving mirror system providing another four viewing positions. The sets of 3-D points extracted from each camera (triangulating with the fringe projector) are combined together to form the whole scan. The scanning time of this device is around 8 s, yet the person is required to stand very still during the scanning. Furthermore, the space inside the scanner is very limited and allows little freedom of body movement. For these reasons, it was difficult to perform scans for nonstandard postures. However, we did manage to obtain a few nonstandard postures and collected a set of ten scans (seven are shown in the paper) related to three different male individuals. The number of points in each scan is approximately 11 000 points. Synthetic HB scans were obtained using POSER software [35]. The data sets were generated by randomly sampling the surface of four different models (two men and two women, two with clothes and two without). 16 simulated scans were then extracted representing human figures in different non standard postures. First, we conducted preliminary tests in order to: 1) compare the performance of the algorithm versions related to the ρ_S and σ Morse functions and 2) assess the appropriateness of the threshold selection with respect to $d = K\epsilon$ (described in Section III-C) and βd used to establish connections between adjacent level-set curves. Fig. 5 shows trials made with two postures, using the ρ_S version (a) and the σ version (b) and we can observe that the segmentation results are not significantly different. Threshold setting $d = K\epsilon$ was investigated out as follows: We segmented a HB scan (in standard posture) several times, each with a different threshold $d = K\epsilon$, with K varying from 2 to 7. These tests were carried out with the algorithm version that utilizes a height function. Fig. 6 (Rows and 1 and

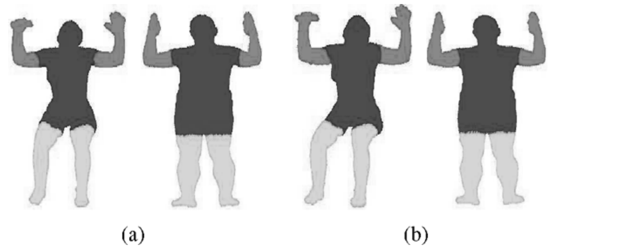


Fig. 5. Cyberware scan and a Wick & Wilson scan segmented using respectively a ρ_S-based version (a) and a σ-based version (b) of the approach.

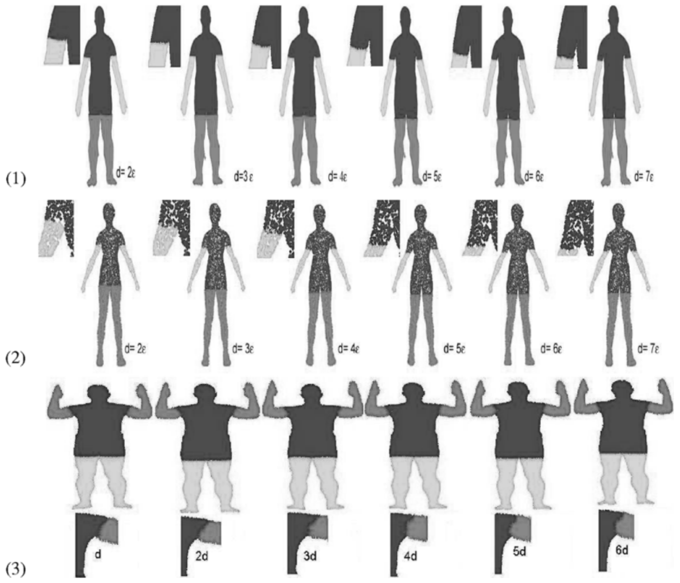


Fig. 6. Rows 1 and 2: A Wick & Wilson HB scan and Cyberware scan segmented using a sequence of six thresholds ranging from $d = 2\epsilon$ to $d = 7\epsilon$, where ϵ is the scanner resolution estimated in Section III-C. Row 3: A Wick & Wilson HB scan segmentation results corresponding to six thresholds $\beta d, \beta = 1, \ldots, 6, d = 2\epsilon$.

2) shows that the overall segmentation appears to be similar throughout the trial. However, by examining the zoomed areas around the armpit we can observe a decrease in the segmentation accuracy as the threshold becomes larger (a decrease in segmentation accuracy is clearly visible for $K \geq 4$). This behavior is caused by "short-circuit" edges between nodes of the arm and nodes of the torso. These short-cuts occur when the distance separating the arm and the torso becomes relatively small compared to the threshold d. Threshold setting βd was investigated as follows. We set the threshold d to the best precision (2ϵ) and we segmented a nonstandard posture six times, each with a different value of β varying from 1 to 6. The segmentation was carried out with the ρ_S-based version of the approach. The results are depicted in Fig. 6 (row 3). The observation of the zoomed area around the right armpit, across the different trials reveals a slight improvement in the accuracy, as β increases until it reaches $3 - D$. Beyond that value it appears to stabilize.

Next, we applied the segmentation algorithm to a collection of scans obtained from different sources, including real and synthetic data. Because of space constraints we show principally the results related to nonstandard postures. Exhaustive results corresponding to standard postures have been published in [18].

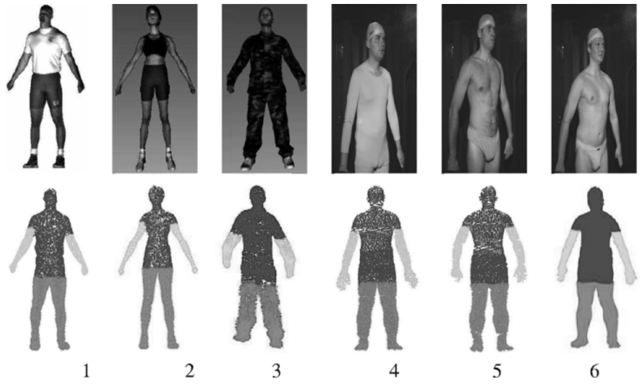


Fig. 7. Segmentation of HB scans acquired with the Cyberware scanner (1, 2, 3) and Wicks & Wilson scanner (4, 5, 6).

Fig. 7 depicts images of the scanned persons, in a standard posture taken during the scanning process and the corresponding segmented scans. These scans were acquired with the Cybeware scanner (1, 2, 3) and the Wicks & Wilson scanner (4, 5, 6). We can observe that the scans are segmented with reasonable accuracy. The scan 3 in Fig. 7 corresponds to that of a fully dressed person. The corresponding surface presents many irregularities caused by wrinkles, in comparison to other scans which present smooth surfaces. Despite the surface irregularity, the segmentation result is reasonable and confirms that the algorithm can cope with ill-conditioned (non smooth) surfaces.

For nonstandard postures, validation experiments were performed using the second version of the algorithm, that involves the Morse function ρ_S. A first series of tests were performed on HB scans, related to people in a seated posture, and collected from the CAESAR website [3]. The Results are depicted in Fig. 8(a). We can see that all the data sets are clearly segmented into five subsets corresponding to the arms, legs and the torso, despite the sitting posture of the subjects and the variety of body shapes. A second series of tests was conducted over several scans acquired from Wick & Wilson scanner [33], the results are depicted in Fig. 8(b), which also shows images of the scanned persons. The results reveal reasonable segmentations that produce correctly separated body parts.

Because of the current shortage of real scans of nonstandard postures we had to resort to computer-simulated data sets in order to examine additional postures. A first set of artificial postures was obtained as follows: We conformed a parameterized HB template to a Cyberware HB scan and then animated the conformed model by varying the template parameters. Some instances from this set are shown after segmentation in Fig. 8(c). The results illustrate correct segmentation for all the samples. The second set was obtained by generating 16 synthetic scan models in different postures using Poser software [35]. These data were obtained by randomly sampling the animated mesh models, thus generating scattered data point sets of similar properties to real scan data. The scans were extracted from four different models comprising a man and a woman, in both dressed and undressed states. The dressed models included the hair. Fig. 8(d) and (e) shows the 16 scans and the segmentation results. We can see that all the data sets are clearly decomposed into the five principal parts for all the different postures, illustrating the robustness of the algorithm with respect to rigid transformation and deformation. Fig. 8(e)

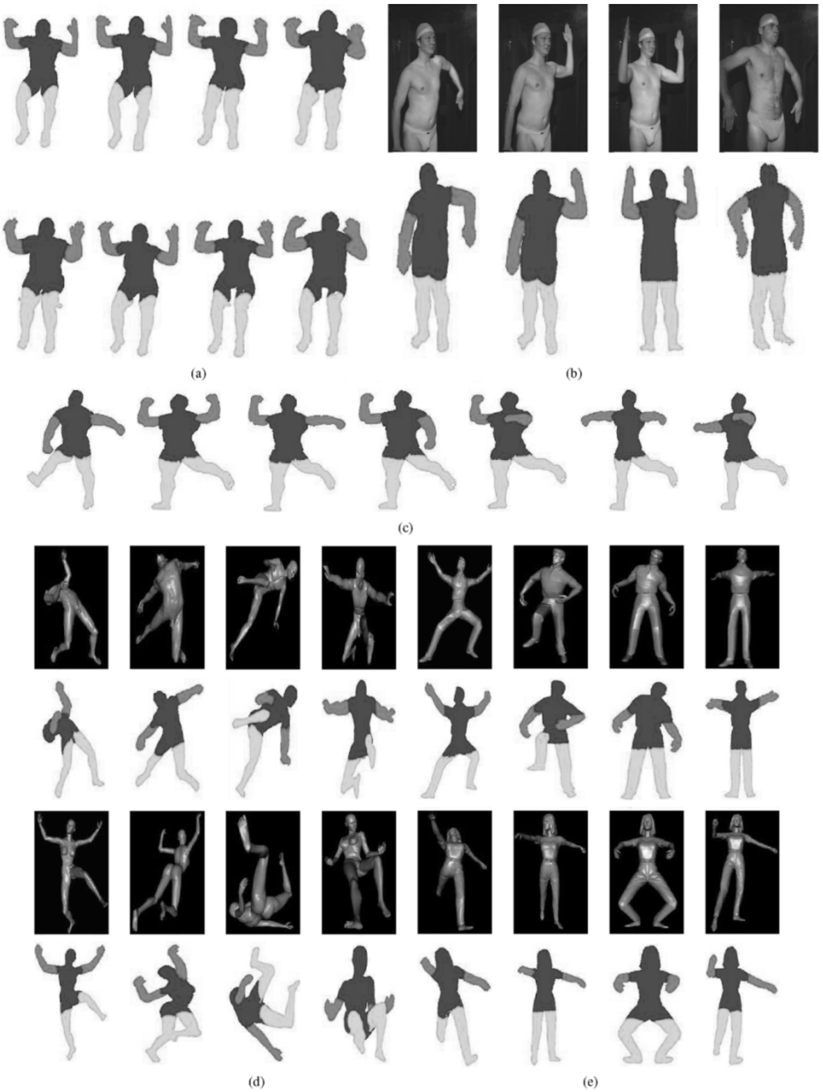


Fig. 8. (a) Segmented real scans corresponding to setting postures acquired withe the Cyberware scanner. (b) Segmented Wick & Wilson scans related to nonstandard postures. (c) Segmentation results of an animated Cyberware scan. (d) and (e) Segmentation results of simulated scans corresponding respectively to scans of naked and dressed characters.

in particular, illustrates the robustness of the algorithm to false branches. Indeed, if we examine for instance, the head of the dressed woman, we can observe wrinkles and waves characterizing the hair surface. These features are likely to cause false critical nodes that consequently generate false branches. However, our algorithm did not fail when applied to this data set. Also we can observe that fingers are not segmented separately, even though some of them appear to be partially detached from the hand and therefore disposed to generate true branches in the DRG. This is explained by the fact that the first criterion established in Section III-D1 states that a true branch must have a length above a given threshold, which is set here to the size of smallest body part, namely the arm. Therefore branches inferred by body parts smaller than the arm, such as the fingers and the head will be discarded by the algorithm.

A. Robustness With Respect to Data Deficiencies

It is also worth mentioning that these results were obtained with poorly scanned data as illustrated in Fig. 9(a), which shows a zoomed area around the groin. The nonregular sampling of the data and the presence of gaps and holes are clearly visible. To further test the robustness of the segmentation algorithm, we corrupted the data by creating large holes in a group of scans representing a variety of postures. The results presented in Fig. 9(b) and (c) reveal a consistent segmentation over all of the corrupted scans: all of these scans are properly decomposed into the five body parts, despite the presence of large gaps. These results confirm that the algorithm is capable of discarding effectively the O-type critical nodes described in Section III-D1. In another experiment we corrupted the scan data by adding Gaussian noise of different amplitudes to the data points. A representative set of the associated segmented scans are depicted in Fig. 10. These validation trials illustrate, therefore, the robustness of our approach against noise disturbance and severe data deficiencies.

V. CONCLUSIONS

This paper presents a topological framework for the segmentation of HB scans. The framework extends the Reeb graph con-

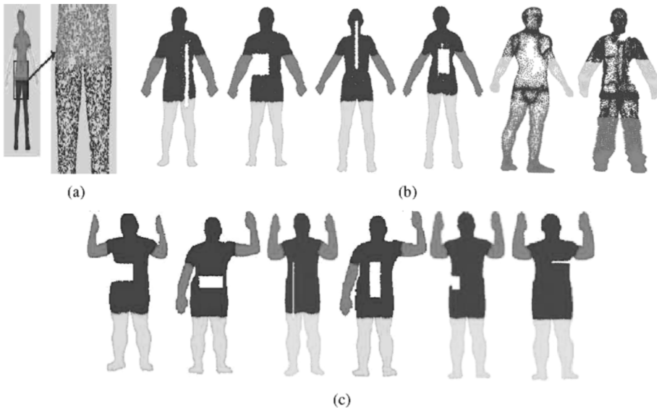


Fig. 9. (a) Zoomed image illustrating the distribution of the scan data. (b), (c) Segmented HB scans corrupted by large holes.

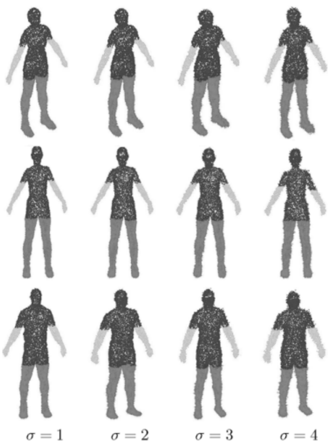


Fig. 10. Segmented HB scans corrupted with a Gaussian noise of different amplitudes.

cept to unorganized clouds of data points by defining connectivity notions. Compared to work previously reported in th eliterature describing on HB segmentation, our framework is differentiated by the following features: 1) it handles directly the raw scan data without the need of any preprocessing or pre-formatting stages; 2) it involves only topology-based techniques; 3) it can cope with arbitrary postures; 4) it offers different configurations in order to accommodate different categories of HB posture; and 5) no post-processing stage is required. With respect to other 3-D skeletonization frameworks, our approach is distinguished by the following aspects: 1) intrinsic robustness, that allows it to cope with data deficiencies and severe corruptions; 2) an efficient implementation; and 3) it provides a solution to the problem of locating the source point, by using a source-point independent Morse function, although at the expense of additional computational cost.

The scan sets used in the experiments were collected from different sources and cover a rich variety of HB shapes and profiles including some severely damaged scans. The results confirm the robustness of the approach with respect to the diversity of scan sources, diversity of the body shapes, rigid transformations, irregular distribution, and corruption of the scan data.

From a quantitative point of view, the experiments confirm the reliability and the repeatability of the two algorithms. In the 152 segmentation trials (58 are shown in this paper), the algorithms always segment the scan into five connected and compact parts that respect the topology of the HB. Cases of severe under-segmentation, over-segmentation, or cases where a segmented part contains disjoint subparts (e.g., from the arm and the leg), never occurred.

While the results show overall reasonable segmentations, it would appear to be difficult to achieve what would be considered a "perfect" segmentation to a human observer because of the ill conditioned behavior of the level-sets at the junctions areas. In fact, we believe that to achieve human segmentation performance would require techniques that go beyond our middle-level processing approach. Nevertheless, the segmentation provided by our approach could be used as a starting point for finer processing and analysis. One might also ask to what extent the joint areas detected by our approach can be used to locate actual useful body landmarks, which are used for instance in garment sizing. While our approach is suitable for delimiting the locale of such landmarks, detecting their precise location may be literally beyond the grasp of purely scan-based approaches, since these landmarks often correspond to bony points below the skin, which are usually detected and located by palpation. There is some hope that a 2-D/3-D approach to landmark detection based on combining both 2-D image features and 3-D surface features used in conjunction with explicit statistical knowledge, as encapsulated in an active appearance model, may provide a route to a more complete analysis of the human form. To this end, we propose that the techniques presented here could provide a useful means of constraining the search space by providing an initial annotation of the body.

Being based on topological analysis, our framework is intrinsically not qualified to handle postures where limbs are joined together, for example closed legs, or arms touching the torso. Dealing with such cases requires that the contours of discontinuities between joined parts of the body be detected and labeled, using local surface analysis and differential geometry techniques or explicit model fitting. The work developed in [12], for example, could perhaps afford an appropriate framework for handling the above cases.

It is quite plausible that our framework can be extended to deal with other variety of 3-D shapes. The approach should integrate mechanisms to accommodate with change of scale and shape. Some directions that can be explored are: 1) using multiple source points for the Morse function ρ_S, analyzing the resulting DRGs, and establishing criteria to reject those degenerate graphs; and 2) a fast implementation of the σ function as it infers a stable DRG; in addition it can be adapted for a multiscale segmentation approach, in the absence of any prior knowledge about the object size.

ACKNOWLEDGMENT

The authors would like to thank Cyberware and the Caeser project, from which we obtained HB scans, and the Edinburgh Virtual Environment Centre for providing access to their HB scanning facilities. Finally, they would like to thank the UK Engineering and Physical Sciences Research Council who supported this work and also the reviewers for their constructive feedback.

REFERENCES

[1] P. R. M. Jones and M. Rioux, "Three dimensional surface anthropometry: Applications to human body," *Opt. Lasers Eng.*, vol. 28, no. 2, pp. 89–17, 1997.

[2] E. Paquet, K. M. Robinette, and M. Rioux, "Management of three-dimensional and anthropometric databases: Alexandria and Cleopatra," *J. Electron. Imag.*, vol. 9, pp. 421–431, 2000.

[3] The Civilian American and European Surface Anthropometry Resource Project Website (2005). [Online]. Available: http://www.sae.org/technicalcommittees/caesar.htm

[4] R. P. Pargas, N. J. Staples, and J. S. Davis, "Automatic measurement extraction for apparel from a three-dimensional body scan," *Opt. Lasers Eng.*, vol. 28, no. 2, pp. 157–172, 1997.

[5] D. Protopsaltou *et al.*, "A body and garment creation method for an internet-based virtual fitting room," in *Advances in Modeling, Animation, and Rendering*, J. Vince and R. Earnshaw, Eds. Berlin, Germany: Springer-Verlag., 2003, pp. 105–122.

[6] F. Cordier, H. Seo, and N. Magnenat-Thalmann, "Made-to-measure technologies for an online clothing store," *Comput. Graph. Applicat.*, pp. 38–48, Jan.–Feb. 2003.

[7] J. Starck and A. Hilton, "Human shape estimation in a multi-camera studio," in *Proc. British Machine Vision Conf.*, Manchester, U.K., 2001, pp. 573–583.

[8] J. Starck, G. Collins, R. Smith, A. Hilton, and J. Illingworth, "Animated statues," *J. Mach. Vis. Applicat.*, pp. 248–259, 2003.

[9] M. Lin, "Tracking articulated objects in real-time image sequence," in *Proc. Int. Conf. Computer Vision*, Corfu, Greece, 1999, pp. 648–653.

[10] D. L. Borges and R. B. Fisher, "Segmentation of 3-D articulated objects by dynamic grouping of discontinuities," in *Proc. British Machine Vision Conf.*, Surrey, U.K., 1993, pp. 279–287.

[11] D. Dion Jr., D. Laurendeau, and R. Bergevin, "Generalized cylinders extraction in a range image," in *Proc. IEEE. Int. Conf. Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, ON, Canada, 1997, pp. 296–299.

[12] F. Ferrie, J. Lagarde, and P. Whaite, "Darboux frames, snakes and superquadrics: Geometry from the bottom up," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 8, pp. 771–784, 1993.

[13] E. Trucco, "Inferring convex subparts from slice data," *Patt. Recognit. Lett.*, vol. 1, no. 12, pp. 707–715, 1991.

[14] M. Kass and D. Terzopoulos, "SNAKES: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–332, 1988.

[15] J. H. Nurre, "Locating landmarks on human body scan data," in *Proc. IEEE Int. Conf. 3-D Digital Imaging and Modeling*, Ottawa, ON, Canada, 1997, pp. 289–295.

[16] L. Dekker, I. Douros, B. F. Buxton, and P. Treleaven, "Building symbolic information for 3-D human body modeling from range data," in *Proc. IEEE Int. Conf. 3-D Digital Imaging and Modeling*, Ottawa, ON, Canada, 1999, pp. 388–397.

[17] C. L. Wang, T. K. Chang, and M. Yuen, "From laser-scanned to feature human model: A system based on fuzzy logic concepet," *Comput.-Aided Des.*, vol. 35, no. 3, pp. 241–253, 2003.

[18] Y. Xiao, P. Siebert, and N. Werghi, "A discrete reeb graph for the segmemtation of human body scans," in *Proc. IEEE Int. Conf. 3-D Digital Imaging and Modeling*, Alberta, Canada, 2003, pp. 378–385.

[19] S. Biasotti, B. Falcidieno, and M. Spagnuolo, "Extended reeb graphs for surface understanding and classification," in *Proc. Int. Conf. Discrete Geometry for Computer Imagery*, Uppsala, Sweden, Dec. 2000, pp. 185–197.

[20] G. Reeb, "Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numrique," *Comptes Rendus Acad. Sciences Paris*, vol. 222, pp. 847–849, 1946.

[21] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology matching for fully automatic similarity estimation of 3-D shape," in *Proc. SIGGRAPH 2001*, New York, 2001, pp. 203–212.

[22] C. Tai, Y. Shinagawa, and T. Kunii, "A Reeb graph-based representation for nonsequential construction of topologically valid shapes," *Comput. & Graph.*, vol. 22, no. 2, pp. 255–268, 1998.

[23] J. C. Hart, "Morse theory for implicit surface modeling," in *Mathematical Visulaization*, H. C. Edge and K. Plothier, Eds. Heidelberg, Germany: Springer-Verlag, 1998, pp. 257–268.

[24] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface coding based on morse theory," *IEEE Trans. Comput. Graph. Applicat.*, vol. 11, no. 5, pp. 66–78, 1991.

[25] S. Biasotti, M. Mortara, and M. Spagnuolo, "Surface compression and reconstruction using Reeb graphs and shape analysis," in *Proc. Spring Conf. Computer Graphics*, Bratislava, Slovakia, 2000, pp. 175–184.

[26] I. Fujishiro, T. Azuma, and Y. Takeshima, "Automating transfer function design for comprehensible volume rendering based on 3-D field topology analysis," in *Proc. IEEE Conf. Visualization*, San Francisco, CA, 1999, pp. 467–470.

[27] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore, "Contour trees and small seed sets for isosurface traversal," in *Proc. Symp. Computational Geometry*, Nice, France, 1997, pp. 212–220.

[28] M. Mortara and G. Patané, "Affine-invariant skeleton of 3-D-shapes," in *Proc. Int. Conf. Shape Modeling and Applications*, Banff, AB, Canada, 2002, pp. 245–252.

[29] A. Verroust and F. Lazarus, "Extracting skeletal curves from 3-D scattered data," *Vis. Comput.*, vol. 16, no. 1, pp. 15–25, 2000.

[30] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.

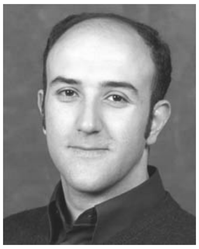
[31] J. S. B. Mitchell, D. M. Mount, and C. H. Paradimitriou, "The discrete geodesic problem," *SIAM J. Comput.*, vol. 16, no. 4, pp. 647–668, 1987.

[32] Cyberware Website (2005). [Online]. Available: http:www.cyberware.com

[33] Wicks and Wilson Website (2005). [Online]. Available: http://www.wwl.co.uk/wwl2/index.html

[34] Edinburgh Virtual Reality Center Website (2005). [Online]. Available: http://www.edvec.ed.ac.uk

[35] Poser Website (2005). [Online]. Available: http://www.curiouslabs.com



Naoufel Werghi (M'97) received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France, in 1996, and the M.Sc. degree in instrumentation and control for computer vision from the University of Rouen, Rouen, France, in 1993.

He has been a Research Fellow at the Division of Informatics, University of Edinburgh, Edinburgh, U.K., and a Lecturer at the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He has also been a Visiting Professor at the Department of Computer and Electrical Engineering, University of Louisville, Louisville, KY. Currently, he is an Assistant Professor at the College of Information Technology, Dubai University College, UAE. His research interests are in the areas of computer vision and computer graphics, in particular 2-D/3-D image processing, 3-D shape analysis and modeling.

Yijun Xiao is currently a Research Assistant in the Department of Computing Science, University of Glasgow, Glasgow, U.K. His research interests are in the areas of computer vision, graphics, and image analysis, with particular focus on 3-D. He has authored or co-authored 14 international journal/conference papers.

Dr. Xiao is a member of the British Computing Society.

Jan Paul Siebert (M'01) received the B.Sc. and Ph.D. degrees from the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, U.K., in 1979 and 1985, respectively.

He is currently a Senior Research Fellow in the Department of Computing Science, University of Glasgow, and the Group Leader for the Computer Vision and Graphics Laboratory. From 1991 to 1997, he was with the Turing Institute, Glasgow, developing photogrammetry-based 3-D imaging systems for clinical applications, and served as Chief Executive since 1994. Prior to this, he was a Scientist at BBN Laboratories, Edinburgh, U.K., from 1988 to 1991. His research interests include 3-D imaging systems and tools for human and animal surface anatomy assessment, and also robot vision systems based on biologically motivated principles. He has co-authored more than 60 international journal and conference papers in these areas.

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 643–651
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A robust approach for constructing a graph representation of articulated and tubular-like objects from 3D scattered data

Naoufel Werghi

College of Information Technology, Dubai University College, Dubai, P.O. Box 14143, United Arab Emirates

Received 5 April 2005; received in revised form 4 September 2005
Available online 29 November 2005

Communicated by S. Dickinson

Abstract

This paper describes an approach for constructing a graph representation of 3D objects and more particularly of articulated and tubular-like objects. For objects without cavities, this representation is a tree structure that encodes the object template while being invariant to global and local rigid transformation. The approach described in this paper has some interesting aspects: (1) It operates on raw 3D scattered data points, without any pre-processing stage. (2) It has low computational cost. (3) It is robust against irregular data point distribution and data deficiencies. This graph representation can be used in various applications such as object coding, recognition, and segmentation.

Keywords: Graph-based 3D shape representation; Articulated and tubular-like objects; Reeb-graph; Geodesic distance; Graph visualization

1. Introduction

3D object shape abstraction and encoding has been receiving an increasing attention in the recent years. It is fuelled by the advances in 3D shape imaging technologies and the proliferation of 3D object model databases, where 3D shape representation plays a fundamental role in model retrieval and shape matching. In the literature (Tangelder and Veltkamp, 2004) shape representation can be broadly categorized in three categories, namely, feature based representations, graph based representations, and other representations. Feature based representations encompass only pure geometry information of the object. In contrast, graph based representations, which use a graph showing how shape components are linked together, embed in addition to some geometric information, topological and structural meanings that are quite suitable for high level processing. Graph based representations include three families, namely, model graph, skeletons, and Reeb-graph. Model graph representations are especially suitable for man-made objects (i.e. CAD/CAM models) and are generally difficult to apply for models of natural shape. Skeletons can be applied to wider shapes including animal and human shapes. Skeleton constructions have been approached using the medial axis model (Chuang et al., 2000; Näf et al., 1996; Siddiqi et al., 2002; Bouix et al., 2005) and the distance transform (Gavani and Silver, 1999; Sanniti di Baja and Svensson, 2002; Svensson and Sanniti di Baja, 2002).

Reeb-graph, introduced by Reeb (1946), is a particular skeleton determined using a continuous scalar function defined on an object surface. The main characteristics of a Reeb-graph are (1) one-dimensional graph structure and (2) invariance to both global and local geometric transformations. These characteristics make it suitable for articulated objects. Reeb-graph has been used in many applications such as shape coding (Tai et al., 1998), shape matching (Hilaga et al., 2001), surface compression (Biasotti et al., 2002), and human-body scan segmentation (Xiao et al., 2003a,b, 2004). In this paper we propose a method for constructing and visualizing a Reeb-graph of a 3D object. Compared to previous methods, our method is

E-mail address: nwerghi@duc.ac.ae

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.10.002

characterized by the following features: (1) It operates on raw 3D data, i.e. cloud of scattered data points (in contrast to methods that require mesh-model data). (2) It is robust against data deficiencies such as irregular distribution reflected by gaps and holes. (3) It has low computational costs. The approach targets objects having tubular-like shapes or a blending of generalized cylinder shapes and assumes that the surface of the object is topologically continuous.

The rest of the paper is organized as follows: Section 2 gives an overview of the approach. Sections 3–6 describe the different stages of the approach. Experimental results are discussed in Section 7. Finally, in Section 8, conclusions are drawn and further research work is suggested.

2. Overview of the approach

The approach operates on a set of 3D scattered data points representing the object shape. It involves four main stages. These are depicted in Fig. 1.

2.1. Computation of the level-sets

In this step, a scalar function map is computed over the set of the data points of the object surface (b) and level-sets representing isovalued points (points having the same scalar function value) are extracted (c).

2.2. Construction of a connectivity graph

Here, each level-set is decomposed into subsets of connected data points according to a given connectivity criteria. We call these subsets level-set curves. The level-set curves are then mapped into a connectivity graph (d), where a node represents a level-set curve and an edge represents a connection between two adjacent level-set curves, i.e. level-set curves belonging to two adjacent level-sets. The nodes in that graph are arranged by ascending horizontal levels which represent the values of the scalar function at the level-sets determined in stage 1. Thus, nodes corresponding to level-set curves which are part of the same level-set will be placed at a same level.

2.3. Extraction of joints and branches

In this stage, the connectivity graph is analyzed to locate the joint nodes and to segment the connectivity graph into groups that correspond to the object branches (e). This stage outputs a particular Reeb-graph, namely, a tree structure (f) in which nodes represent the branches and edges represent the joint nodes. The tree structure reflects the assumption that the object does not contain holes.

2.4. Visualization

The Reeb-graph of the object is automatically visualized (g) using the tree-structure obtained in the previous stage. This graph encodes the object branches and parts while describing the evolution of the scaler function across them.

3. Computation of the level-sets

Given a set of data points V and a scalar function: $F : X \mapsto \mathscr{R}$ where $X \in \mathscr{R}^3$ is a data point, level-sets in dis-

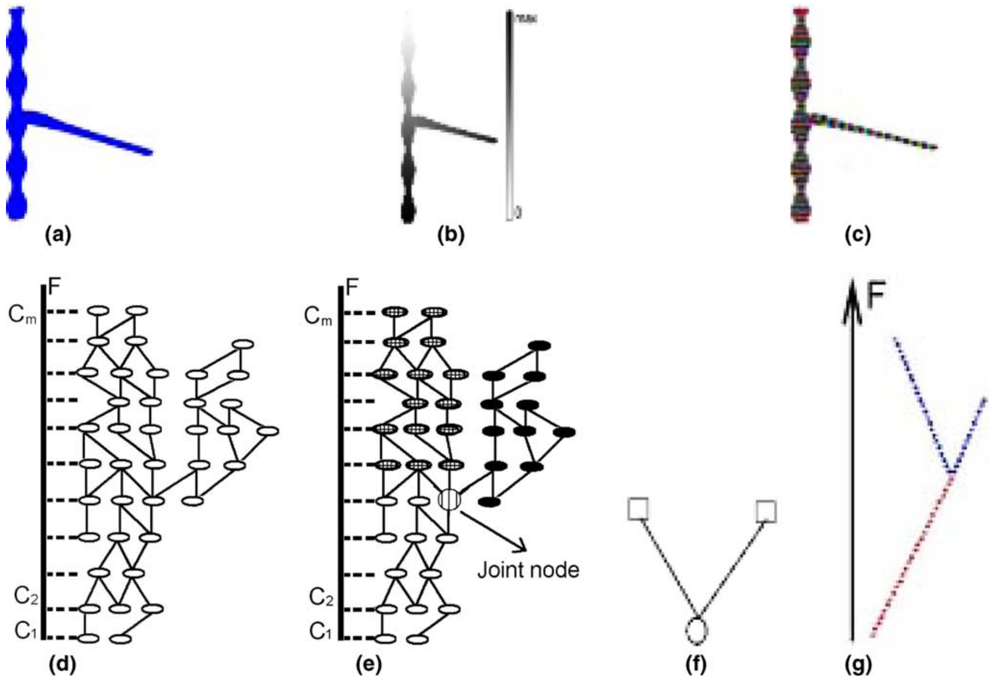


Fig. 1. (a) A 3D object. (b) Grey level mapping of the scalar function on the object surface. (c) Level-sets of the scalar function. (d) The connectivity graph. (e) Determination of the joint nodes and the branches. (f) The tree structure. (g) Visualization of the Reeb-graph.

crete space are formally defined by $\{X \in V, F(X) = C_k\}$ where $C_k, k = 1 : m$ is a set of discrete values ranging from the minimum value to the maximum value of the function F in the domain V. To ensure a stable representation, the scalar function should be invariant with respect to rigid transformations. The curvature function satisfies this condition, however, it is highly sensitive to noise and data corruption. A function employing the geodesic distance (Mitchell et al., 1987) is more appropriate as such distance is quite resistant to data corruption in addition to be invariant to rigid transformations. We utilize the function defined by $F(X) = gd(X, S)$ which returns the geodesic distance from a point X to a source point S. In the literature, the Dijkstra algorithm (Cormen et al., 1990) has been the most popular tool for computing geodesic distances between a group of points and source point. However it has a high computational cost. So we rather developed an efficient algorithm, tailored for our application. The algorithm deploys a wavefront propagation technique, which is based on the following principle: Given a centred wave on a manifold, all the points on the wavefront have the same geodesic distance to the wave centre and form thus a level-set. The wavefront propagation algorithm operates on a binary voxel grid since it is easy to define a neighborhood in voxel space and to traverse connected voxels. Due to these well-behaved properties, wavefront-propagation on a voxel grid can have a very simple mathematical form as follows:

$$\begin{cases} W_0 = \{v_s\}, \\ W_{i+1} = (W_i \oplus e - (W_i \oplus e) \cap S_i) \cap \bar{S}_i, \end{cases}$$

where W_i is the wavefront generated on the ith iteration of the algorithm; v_s is the source voxel; S_i is the set of all 1-valued voxels visited at the iteration i and located at the same geodesic distance from the source voxel v_s. $\bar{S}_i$ is the complement set of S_i. $\oplus$ denotes the morphological dilation operator and $\boldsymbol{e}$ is a $3 \times 3 \times 3$ structuring element composed of 27 1-valued voxels. At the beginning, the wavefront is the source voxel associated to the source point, then the wavefront iteratively spreads on the voxelised surface. In each iteration, the wavefront is the level-set containing voxels with the same geodesic distance. The attractive aspect of this technique is that it simultaneously extracts the level-sets while computing the scalar function F. It is easy to prove that the computational complexity in each iteration is $O(n_i)$, where n_i is the number of voxels in W_i. Therefore the complexity of the whole algorithm is $O(N)$, where N is the number of all 1-valued voxels.

The source point S can be selected manually or determined automatically following these steps: (1) Choose a source point at random from the set of data points. (2) Compute the geodesic distance map. (3) Choose the point corresponding to the maximum geodesic distance value. For articulated and tubular-structured objects, such point will be located at the extremity of the object parts. This presents the advantage of maximizing the range of the geodesic distance.

4. Construction of a connectivity graph

For a perfect data, a level-set would be a compact set of connected points. For real data characterized by a non-uniform distribution and gaps, the level-set is rather fragmented into sets of connected points. These sets, which we call level-set curves, are conceptualized by the following definitions.

Definition 1 (*connectivity of point sets*). Two point sets $P = \{p_i\}$, $i = 1, \ldots, m$ and $Q = \{q_j\}$, $j = 1, \ldots, n$ are defined as connected if $\exists p_i \in P$ and $\exists q_j \in Q$ such that $|p_i - q_j| \leqslant \tau$. Where $|p_i - q_j|$ denotes the distance between points p_i and q_j and τ is a given threshold. The above definition also holds for the connectedness between two points for the particular case where the sets P and Q contain a single point each.

Definition 2 (*connective point set*). A point set C is connective if $\forall$ subset $\Omega \subset C$ and $\Omega \neq \emptyset$, Ω and $\overline{\Omega}$ are connected. Here $\overline{\Omega}$ denotes the complement of Ω in C. Definition 2 defines a 'tight' point set in which all the points are connected.

Definition 3 (*Level-set curve*). A level-set curve is an iso-valued connective point set, that is a group of points, that have the same scalar function value, and which forms a connective point set.

At the implementation level, the threshold τ used in Definition 1 is set to the resolution of the 3D data points. The resolution is estimated as follows: we determine the distribution of the distance values of the most closed pairs (i.e. nearest neighbor tuples) of data points over a large set of data. This will permit to construct the 3D density histogram. The median value or the weighted average are reasonable estimates of the resolution, but more elaborated techniques can be used however (Scott and Sain, 2004).

The connectivity graph is an oriented graph where a node represents a level-set curve and where an edge represents a connection between two adjacent level-set curves, (i.e. two level-set curves belonging to two adjacent level-sets).

The connection between the two level-set curves is established according to Definition 1. Ideally, when we assume a clean data, the connectivity graph will be reduced to a tree structure. But practically, the deficiencies of the data produce topological disturbance that causes false joint nodes. This results in a corrupted graph structure. Fig. 2 shows an example illustrating this aspect in more details. For simplicity, the example represents a cylinder-like shape and covers only a portion of three adjacent level-sets. If we assume an ideal data, each level-set consists of an organized set of connected points Fig. 2(a). So each one will represent a single level-set curve. This will result in a graph composed of three nodes (Fig. 2(b)). For real data, a

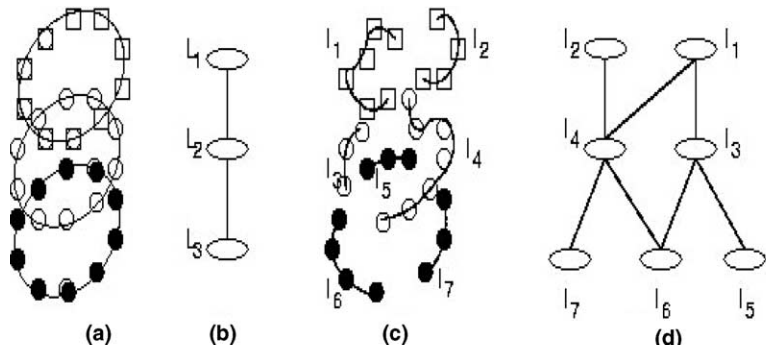


Fig. 2. (a) Three level-sets representing a portion of clean data. Each level-set contains a single level-set curve (group of linked points), resulting thus in a connectivity graph containing three nodes (b). (c) Data corruption causes each slice to decompose into several level-set curves. This results in a disorganized graph (d). In this example the original nodes L_1, L_2 and L_3 have degenerated respectively into (l_1, l_2), (l_3, l_4) and (l_5, l_6, l_7).

level-set might be composed of several level-set curves because of data corruption, leading to several nodes per level-set. The example in Fig. 2(c) shows three level-set curves for level-set 3 and two level-set curves for each of level-set 1 and level-set 2. The three initial nodes L_1, L_2, L_3 have degenerated into seven nodes $l_1, l_2, \ldots, l_7$. Setting afterwards the connections among the nodes (leads to a disorganized graph Fig. 2(d)). Therefore, at the scale of the object the connectivity degenerates into chaotic graph representing false joint nodes and false branches. The challenge therefore, is to be able to determine the correct joints and branches from such degenerated graph to obtain a representation faithful to the topological structure of the object. This will be described in the next section.

5. Extraction of joint nodes and branches

The strategy adopted in this stage is based on the following analysis: in the connectivity graph we identified three primary topological patterns. These patterns are called O-type, λ-type and Y-type. For example, The group of nodes (l_7, l_4, l_2, l_1), (l_6, l_4, l_1, l_3), and (l_6, l_5, l_3, l_1) represent a λ-type, an O-type, and a Y-type, respectively. O-type comprises two joint nodes connected by two branches. This pattern reflects data corruption (gaps, missing data) because we assumed that the object does not contain cavities and therefore this pattern is simply ignored. The λ-type and Y-type are topologically identical as each represents three branches that meet at the joint node. We can reduce the number of patterns to a single one by considering a topographic constraint when constructing the connectivity graph. Indeed, by arranging the nodes in ascending levels, where these levels represent the values of the geodesic function at the different level-sets, and by placing at each level the nodes associated to level-set curves which belong to a same level-set, only Y-type patterns can figure in the connectivity graph. To distinguish genuine Y-types from false ones which are inferred by data deficiencies, we impose the length of a branch (evaluated in terms of inferred number of levels) associated to a true Y-type to be above a minimum value.

Based on these considerations, the following criteria are used to identify a 'true' Y-type which consists of 'true' branches and a joint node: (1) A 'true' branch is upward and must satisfy: $|L_{\max}(\text{branch}) - L_{\min}(\text{branch})| > L_{\lim}$ where $L_{\max}/L_{\min}(\text{branch})$ denotes the maximum/minimum levels inferred by the branch in the connectivity graph. $L_{\lim}$ is a threshold that represents the minimum length allowed for a branch. (2) A true joint node has at least two branches.

The setting of $L_{\lim}$ value is very crucial. In effect a low value might generate false branches whereas a high value causes actual branches to be lost. Here we note that a genuine choice of $L_{\lim}$ should stand on a prior knowledge of the relative sizes of the object's segments. For example when dealing with animal shapes, we know that the minimum size of a functional part cannot be less one tenth of the whole size. A reasonable value of $L_{\lim}$ can then be set based on that proportion.

Based on the details described above, the detection of the true joint nodes is accomplished using the algorithm given below. This algorithm browses the connectivity graph level by level starting from the highest level that corresponds to the level-set associated with the maximum value of the geodesic distance function. In a second phase the true branches are extracted based on the locations of extreme nodes and joint nodes. Having located the joint nodes and branches, a tree structure is constructed afterwards where nodes represent branches and edges represent joint nodes linking the branches.

```
Notation:

Branch: A group of connected nodes.
Branch(Node): The Class containing the Node.
For each level in the connectivity graph
    For each node at that level
        Extract from the upper level the group of nodes
        $(l_1, l_2, \ldots, l_m)$ connected to that node
        If $m = 0$
            Add this node to the list of extreme nodes
            Branch(node) := {this node}
        Else
```

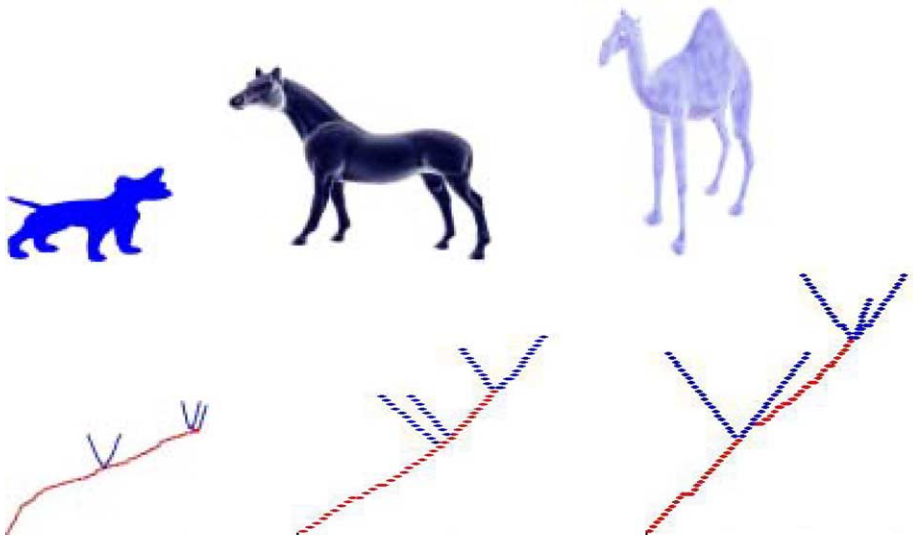


Fig. 3. Instances of objects and their graph representations.

```
            From the set: $\{Branch(l_i), i = 1, \ldots, m\}$ select the
            Branches verifying: $\{B_j : |L_{\max}(Branch(l_j)) - L_{\min}(Branch(l_j))| > L_{\lim}, j = 1, \ldots, n\}$ $(n \leqslant m)$
                If $n \geqslant 2$
                    add the nodes $l_j, j = 1, \ldots, n$ to the list of
                    Joint nodes
                End If
            $Branch(node) = \{Branch(node)\} \cup \{Branch(l_1)\} \cup \cdots \cup \{Branch(l_m)\}\}$
        End If
    End for
End for
```

6. Visualization

In this stage, the topological structure embedded in the Reeb-graph of the object is visualized. The tree structure outputted by the previous stage is browsed in a depth first fashion. At each visited node, the associated branch is mapped into a 2D curve where the x-coordinate and the y-coordinate represent a level-set curve and its corresponding level in the connectivity graph. We have to mention here that the orientation of the 2D curve reflects only the evolution of the geodesic distance at the associate branch and does not contain geometrical information as it is the case in skeletons.

7. Experiments

We applied our approach to a variety of objects acquired from different sources. Fig. 3 shows results obtained with animal shapes. These models were acquired from Princeton Benchmark.[1] We can observe that the resulting graphs reflect correctly the topology of the models. The graphs of the dog and the camel present each a main branch and five ramifications that correspond to the limbs and the tail. The graph of the horse shows only four ramifications as the corresponding model does not include a tail. One might ask if using a smaller values of $L_{\lim}$ (Section 5) would permit to detect finer details of the shape (e.g. the hears of the dog or the horse). This is possible under ideal conditions (i.e. dense and clean data), however practically because of data irregularities, lowering the $L_{\lim}$ might cause undesired noise branches in the graph. To ease these effects, one solution would be to clean-up the data and increase the resolution in a pre-processing stage.

Fig. 4 shows three models of blood vessels, composed of three, five and eight branches respectively. These models where obtained from the CVMT Lab.[2] The second row in Fig. 4 depicts the corresponding graphs. We can see that all the trees exhibit a correct representation in terms of structure and number of branches. We conducted other trials on the third blood vessel model to assess the stability of the representation with respect to rigid transformations and change of source point location. The results are depicted in Fig. 5. In the first trial (a) we applied random rotation of the model and kept the same source point. The corresponding tree remains unchanged. This illustrates the invariance of the geodesic distance function to rigid transformations. In the two other trials (b) and (c) we rotated the model and changed the location of the source points. While the resulting trees look having different configuration, they do preserve the same number of branches and nodes, reflecting thus the stability of the representation with respect to the topological structure of the model. In a second series of experiments we tested the robustness of the approach with respect to data deficiencies. It is worth to

[1] http://shape.cs.princeton.edu.

[2] Computer Vision and Media Technology Lab, Aalborh University, Denmark. www.cvmt.dk.

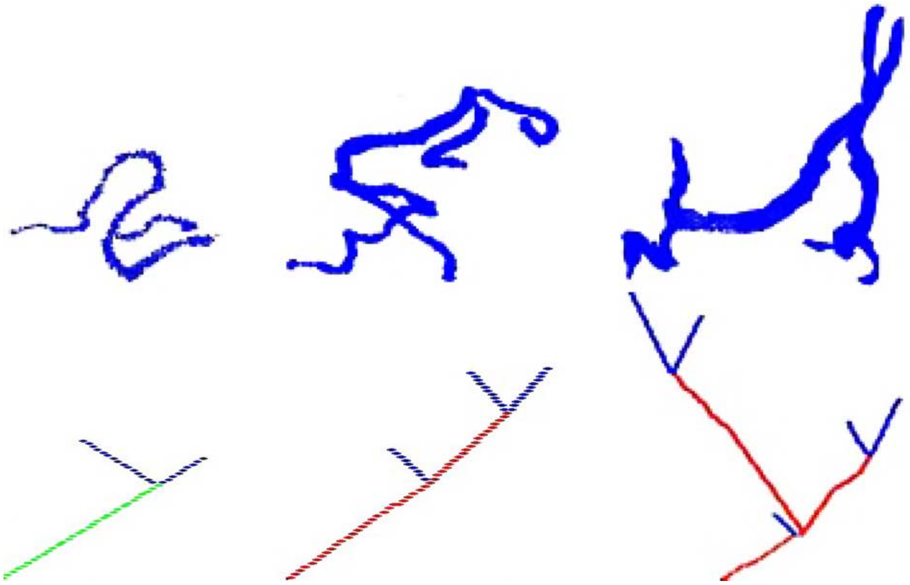


Fig. 4. Instances of blood vessels and their graph representations.

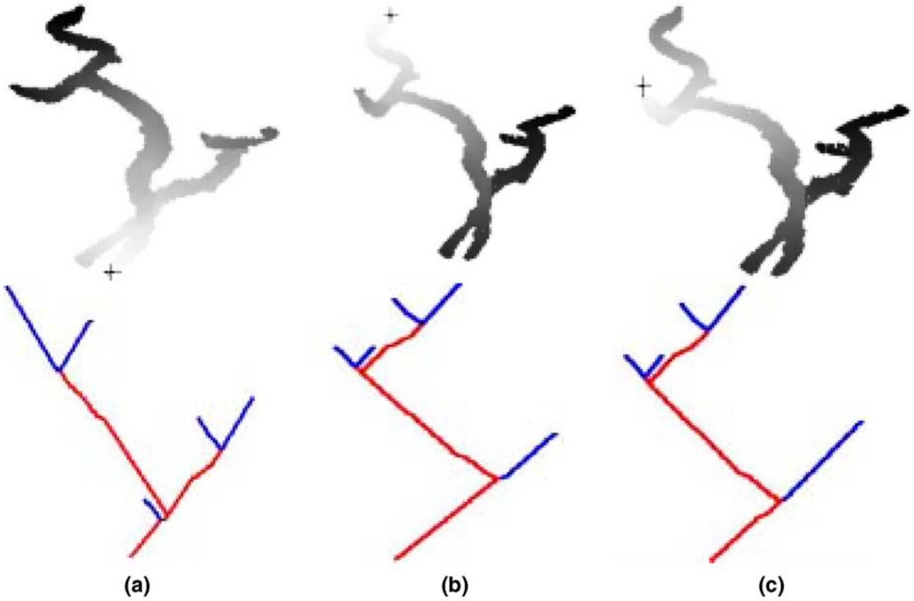


Fig. 5. Grey level mappings of the geodesic function corresponding to different source points (marked by a "+") for a rotated blood vessel model and the resulting graph representation.

mention that the data used in the experiments are characterized by irregular distribution as shown by the zoomed area of the camel in Fig. 7. Firstly we corrupted some objects by a boundary noise as depicted in the first row of Fig. 6. The second row shows the corresponding trees exhibiting a correct structure. To further check the robustness of our method we intentionally corrupted some models by creating artificial holes at different locations of their surfaces (Fig. 7). Despite these severe alterations, the approach produced correct graphs as depicted by the figure. These results illustrate particularly that the connectivity graph analysis described in Section 5 succeeds in rejecting the O-type nodes in the connectivity graph.

Other experiments have been conducted on both synthetic and real articulated objects to test how the approach cope with shape deformation. Fig. 8: first row, shows instances of a synthetic object having undergone various deformation. The next row depicts the resulting trees (the first four trees). We can see that all the trees have the same structure. Some trees shows different orientations for some branches, but this does not reflect any geometric properties. The third row in Fig. 8 shows animated instances of a frog. These instances were generated using the animation software Poser.[3] The next rows depicts the corresponding

[3] www.curiouslabs.com.

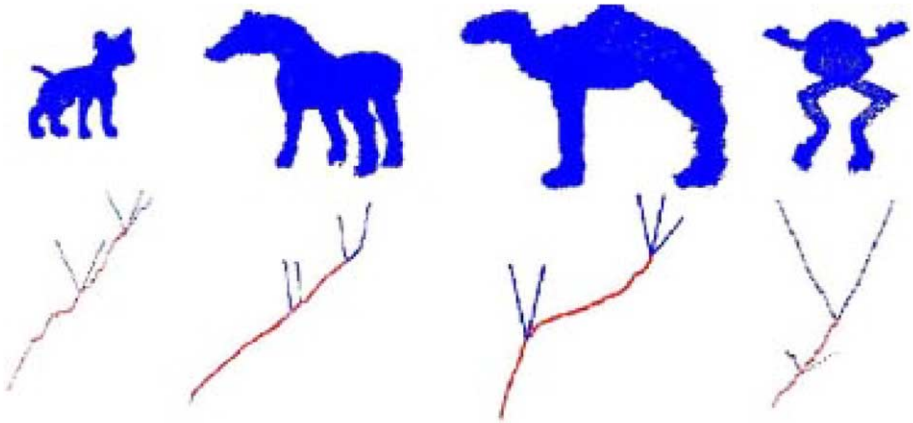


Fig. 6. Object corrupted by a boundary noise and their corresponding DRG's.

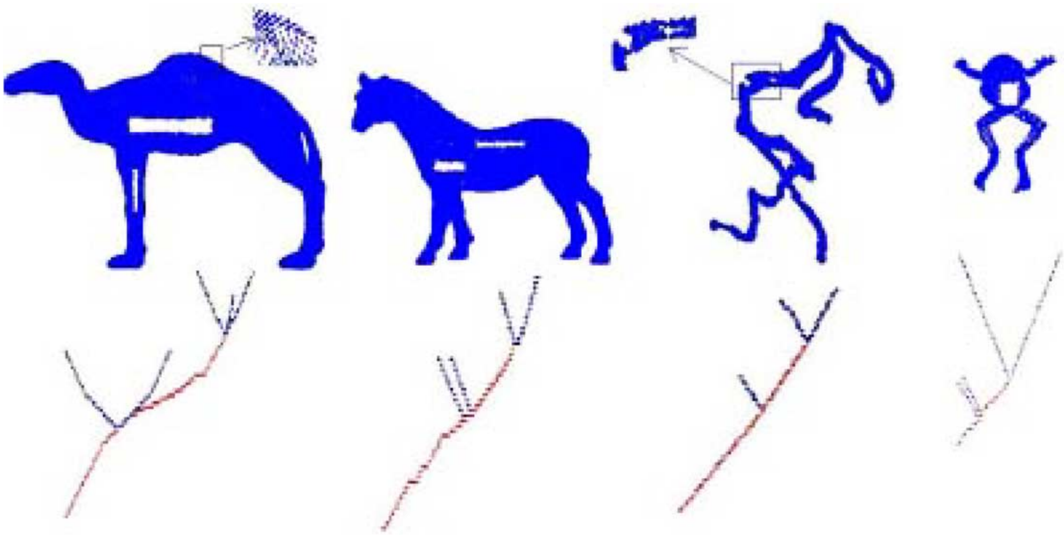


Fig. 7. Objects corrupted by artificial holes and their corresponding graphs.

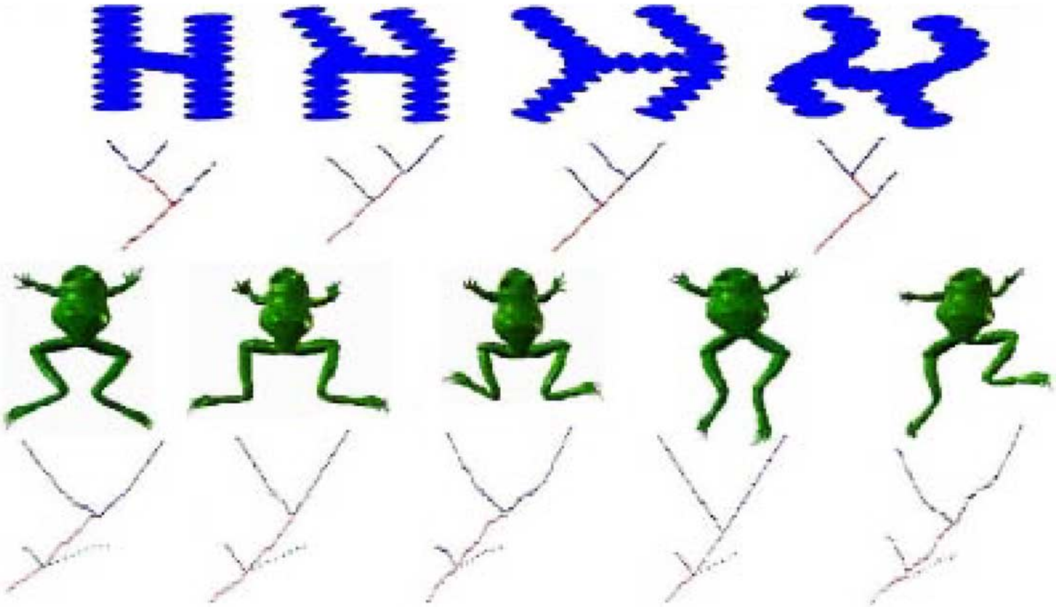


Fig. 8. Row 1: instances of a synthetic objects. Row 2: their corresponding graphs. Row 3: instances of a frog model in different shapes. Row 4: the corresponding trees.

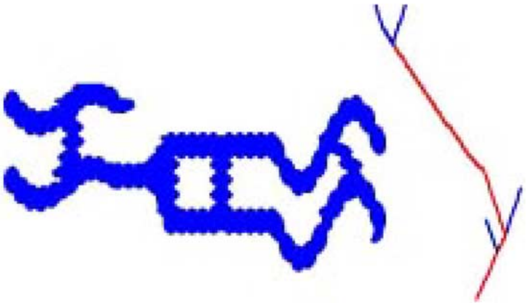


Fig. 9. A complex shape presenting cavities and the corresponding graph.

trees. The stability of the tree representation is clearly noticed. Fig. 9 illustrates instance of shape that our method cannot handle. The object presents cavities, which infer topological discontinuities. Such discontinuities are not detected by our algorithm which appears to consider the segments forming a cavity as a single segment. We plan to address this issue in future work.

The last bunch of tests has been conducted to illustrate the usefulness of our approach in a particular application namely object segmentation. We developed a simple segmentation method based on that graph representation. Basically the method consists in mapping the branches of the tree with the model data. We would like to note the aim was not to develop a complete segmentation framework but rather to demonstrate the usefulness and the applicability of our approach. The tests were conducted with a variety of synthetic and real objects. Results are depicted in Fig. 10. We can see that the different segments in each object have been retrieved. The extraction is not accurate however as it can be seen for instance in the first synthetic object (H-shape) in row 1, and also in the horse and the second blood vessel. We believe that a more thorough segmentation approach involving analysis of the level-sets around the joint nodes of the graphs would yield better segmentation.

The approach has been implemented with Matlab on a 1.2 GHz Pentium III machine. The code, however, is not optimized. To give an idea about the running time, the graph construction of the third blood vessel model (which contains 14,960 points) took about 8 s.

8. Discussion and conclusion

In this paper we proposed an approach for automatically constructing a topological representation of 3D objects. The main features of this approach are: (1) It operates on crude 3D scattered data. (2) It is robust against irregular data point distributions and severe data deficiencies such as gaps. (3) It involves an efficient technique that computes simultaneously the geodesic function and the associated level-sets. This technique demonstrates a novel algorithm characterized by a low computational costs. The experiments conducted on a variety of objects and shapes confirmed the effectiveness and the robustness of the approach. We illustrate the applicability of the proposed graph representation in object segmentation. Yet it can be also exploited for data registration and object recognition.

Compared to Reeb-graph based methods, our approach is more efficient due to the fast technique used for computing the level-sets. It is also more robust as it can cope with severe data alteration as it has been illustrated in the experiments. To the best of our knowledge we are not aware of any other Reeb-graph construction technique that can handle effectively such altered data.

With respect to skeleton construction, our approach is not qualified to compete with other approaches (e.g. the

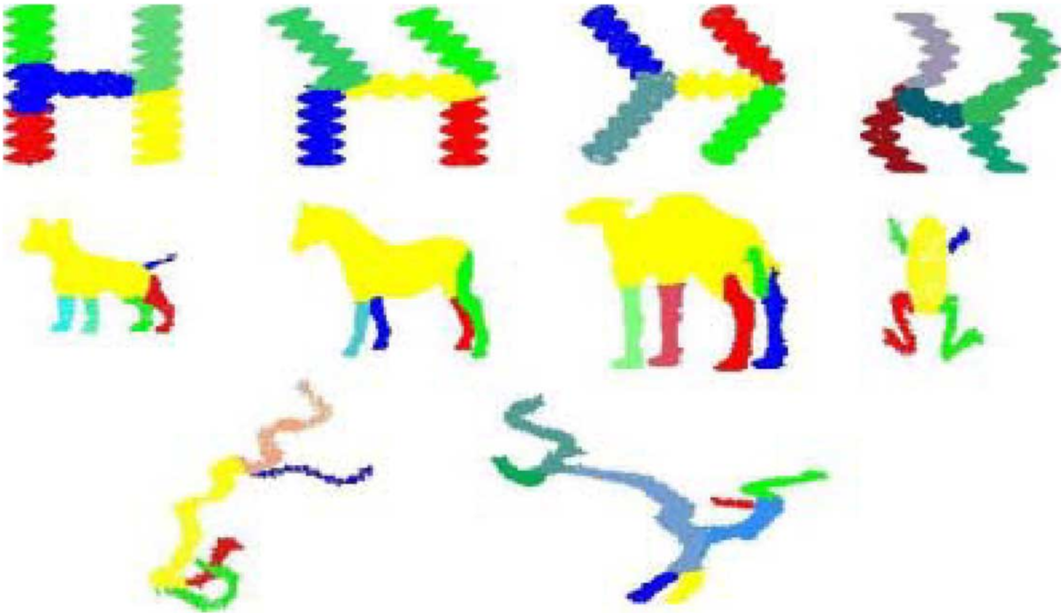


Fig. 10. Examples of segmented models.

medial-axis technique), as it is a graph construction approach intended essentially to deliver a representation that encompasses only topological and a structural information. However in the medial-axis construction approaches, reported in the literature so far, it is not clear how and to what extent they can cope with instances of data deficiencies such the ones we did test in our experiments (Fig. 7). Our assumption is that unless detected and handled properly such gaps will cause distorted axes that might result on incorrect skeleton.

Our approach presents some limitation however. It needs some user intervention to set adequate value of L_{lim} (Section 5). So far the approach cannot handle objects presenting topological continuities (e.g. cavities). We plan to address these issues in the future. For the scalar function in particular, we need to investigate source point-invariant functions. This work can be also further explored in other directions. We plan to investigate the extension of our approach to skeleton extraction. Such extraction will inherit the robustness of our approach against severe data deficiencies.

References

Biasotti, S., Mortara, M., Spagnuolo, M., 2002. Compression and Reconstruction using Reeb graphs and Shape Analysis. In: Proceedings of Spring Conference on Computer Graphics, Bratislava, Slovakia, pp. 175–184.

Bouix, S., Siddiqi, K., Tannenbaum, A., 2005. Flux driven automatic centerline extraction. Med. Image Anal. 9 (3), 209–221.

Chuang, J.H., Tsai, C.H., Ko, M.C., 2000. Skeletonization of three-dimensional object using generalized potential field. IEEE Trans. Pattern Anal. Mach. Intell. 22 (11), 1241–1251.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., 1990. Introduction to Algorithms. McGraw-Hill, New York, NY.

Gavani, N., Silver, D., 1999. Parameter controlled volume thinning. Graph. Models Image Process. 61 (3), 149–164.

Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T., 2001. Topology matching for fully automatic similarity estimation of 3d shape. In: Proceedings of SIGGRAPH, New York, USA, pp. 203–212.

Mitchell, J.S.B., Mount, D.M., Paradimitriou, C.H., 1987. The discrete geodesic problem. SIAM J. Comput. 16 (4), 647–668.

Näf, M., Kübler, O., Kikinis, R., Shenton, M.E., Székely, E., 1996. Characterization and recognition of 3D organ shape in medical image analysis. IEEE Workshop Math. Meth. Biomed. Image Anal., 139–150.

Reeb, G., 1946. Sur les points singuliers d'une forme de Pfaff completement integrable ou d'une fonction numèrique. Comptes Rendus Acad. des Sci., Paris, France 222, 847–849.

Sanniti di Baja, G., Svensson, S., 2002. A new shape descriptor for surfaces in 3D images. Pattern Recognition Lett. 23, 703–711.

Scott, D.W., Sain, S.R., 2004. Multi-dimensional estimation. In: Handbook of Statistics. In: Rao, C.R., Wegman, E.J. (Eds.), Data Mining and Computational Statistics, vol. 23. Elsevier, Amesterdam.

Siddiqi, K., Bouix, S., Tannenbaum, A.R., Zucker, S.W., 2002. Hamilton–Jacobi skeletons. Internat. J. Comput. Vision 48 (3), 215–231.

Svensson, S., Sanniti di Baja, G., 2002. Curve skeletonization of surface-like objects in 3D images guided by voxel classification. Pattern Recognition Lett. 23, 1419–1426.

Tai, C., Shinagawa, Y., Kunii, T., 1998. A Reeb graph-based representation for non-sequential construction of topologically valid shapes. Comput. Graph. 2 (22), 255–268.

Tangelder, W.H., Veltkamp, R.C., 2004. A survey of content based 3D shape retrieval methods. In: International Conference on Shape Modeling and Applications, Genova, Itlay, pp. 145–156.

Xiao, Y., Siebert, P., Werghi, N., 2003a. A discrete Reeb graph for the segmentation of human body scans. In: Proceedings of IEEE Conference of 3D Digital Imaging and Modeling, Alberta, Canada, pp. 378–385.

Xiao, Y., Werghi, N., Siebert, P., 2003b. A topological approach for segmenting human body shape. In: Proceedings of the International Conference on Image Analysis and Processing, Mantova, Italy, pp. 82–87.

Xiao, Y., Siebert, P., Werghi, N., 2004. Topological segmentation of discrete human body shapes in various postures based on geodesic distance. In: Proceedings of the International Conference on Pattern Recognition, Cambridge, UK, pp. 131–135.

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Pattern Recognition Letters 26 (2005) 663–677

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A discriminative 3D wavelet-based descriptors: Application to the recognition of human body postures

Naoufel Werghi

College of Information Technology, Dubai University College, P.O. Box 14143, Dubai, United Arab Emirates

Received 22 October 2003; received in revised form 31 August 2004

Abstract

This paper deals with the recognition of human body postures from a cloud of 3D points acquired by a human body scanner. Motivated by finding a representation that embodies a high power of discrimination between posture classes, a new type of 3D shape descriptors is suggested, namely wavelet transform coefficients (WC). These features can be seen as an extension to 3D of the 2D wavelet shape descriptors developed by (Shen, D., Ip, H.H.S., 1999. Pattern Recognition, 32, 151–165). The WC is compared with other 3D shape descriptors, within a Bayesian classification framework. Experiments with real scan data show that the WC outperforms other standard 3D shape descriptors in terms of discrimination power and classification rate.

Keywords: 3D Human body scan data; 3D Human posture recognition; 3D Shape descriptors; Wavelet transform; Bayesian classification

1. Introduction

The emergence of 3D imaging technology that enables full scanning of the human body surface with reasonable measurement accuracy and acceptable computational cost is a recent phenomenon. This advance facilitates the exploitation of the human body form in various areas such as anthropometrical research (e.g., Jones and Rioux, 1997; Paquet et al., 2000), clothing design (e.g., Jones et al., 1995; Pargas et al., 1996; Dekker et al., 1998; Cordier et al., 2003) and virtual human animation (e.g., Sun et al., 2001; Starck et al., 2002). The raw data delivered by the human body scanner requires substantial main memory and back-up storage resources but it contains too little semantic information to be useful for potential applications. The recognition of body posture has a major role in many applications requiring automatic processing of the scanner data. Automatic segmentation techniques of the human body

E-mail address: nwerghi@duc.ac.ae

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.09.018

scanner data usually use prior information on the human body posture (e.g., Dekker et al., 1998; Cordier et al., 2003; Xiao et al., 2003). Applications that exploit scanned human body data in TV and cinema production involve the fitting of a generic model to the scanned data to obtain a realistic model that, for instance, can be integrated into a movie sequence. Here, the identification of the posture from the scanner data is useful as a good initialization for iterative techniques that may be involved in the fitting algorithm, in particular to guarantee and accelerate the convergence of the algorithm.

The work presented in this paper describes a method of recognizing human body postures from 3D scanner data by adopting a model-based approach. The problem is stated as follows. Given a set of posture models and a query posture, find which posture model corresponds to the query posture. The paradigm followed to solve this problem is built upon three premises: representation, feature extraction and classification. The emphasis in this paper is on representation and feature extraction.

2. Representation

In shape recognition techniques, objects are represented by numerical features, which are grouped into vectors, to remove data redundancy and reduce data dimension. The data we deal with consist of scattered 3D points that represent the surface shape of the human body. Most of the human body scanners provide a complete data set that covers the entire body surface. This encourages investigation of global features that can be exploited in 3D shape identification. Moments as global 2D shape features have been used extensively in image analysis and description. Attention has been mainly oriented towards moments that are invariant with respect to translation, rotation and scale. Such moments were first proposed by Hu (1962). After that, a variety of moments were developed. Examples include, statistical moments (Chim et al., 1999), orthogonal moments, such as Legendre moments, Fourier–Mellin moments, Zernike moments and pseudo-Zernike moments. It has been shown that orthogonal moments are less redundant, less sensitive to noise and more informative than geometrical moments (Teague, 1980). A good survey of 2D moments can be found in (Teh and Chin, 1988) and (Belkassim et al., 1991). However, less study has been done of the 3D case. One reason for this is that most of the 3D imaging devices do not provide a complete data set in terms of surface coverage. Being sensitive to missing data and occlusions, global features are not suitable for such cases. Nevertheless, there have been some attempts to define frameworks for the construction of 3D moments. Sadjadi and Hall (1980) pioneered the development of 3D geometric moment invariants. Their framework built a family of three invariant moments with degrees up to the second order. Using complex moments, Lo and Don (1989) constructed a family of 12 invariant moments with orders up to the third degree. Their moments were mainly used to estimate 3D transformations and their performance was not assessed for classification. In addition, these moments are not derived from a family of orthogonal functions, and they are therefore subject to correlation and redundancy. Motivated rather by computational efficiency, Sheynin and Tuzikov (2001) proposed a computational framework for the calculation of Cartesian moments. However, their approach is restricted to polyhedral objects. The desirable properties of orthogonal moments, in terms of sensitivity to noise and information redundancy, motivated the development of families of orthogonal 3D moments. Examples include 3D Zernike moments (Canterakis, 1997) and 3D Haar moments (Schael, 1997). These efforts, however, did not provide experimental frameworks for testing these moments. 3D shape descriptors based on the 3D discrete Fourier Transform were proposed by Vranic and Saupe (2001) for model retrieval applications. However, the discriminative power of this type of feature was not assessed.

In this work, we present a new family of 3D shape descriptors, namely the wavelet-based descriptors. The performance of these features is evaluated and compared with 3D Zernike moments and 3D Fourier descriptors. In a previous study Werghi and Xiao (2002), it was shown that

3D geometric moments proposed by Lo et al. (1998) are far less powerful than the wavelet-based 3D shaped descriptors in terms of their discriminative capabilities.

2.1. Wavelet-based representation

The wavelet was introduced by Morlet and Grossman (1984) as a time-scale analysis tool for non-stationary signals. It was further developed by many authors (e.g., Mallat, 1989; Daubechies, 1990; Meyer, 1997; Jaffard et al., 2001) and rapidly found applications in many areas. A wavelet function is a function that is well localized in the space and frequency domains. From a mother function $\psi(r)$, a family of wavelet functions

$$\psi_{a,b}(r) = \frac{1}{a}\psi\left(\frac{r-b}{a}\right), \quad a > 0$$

is derived. This family is obtained by shifting the wavelet mother by b (the shifting parameter) and by dilating (stretching) it with a (the scaling parameter). The wavelet transform at the scale a and shift b is

$$\int_{-\infty}^{\infty} f(r)\psi_{a,b}(r)\,\mathrm{d}r$$

The wavelet transform embodies information about the regularity and the spectrum of the frequency around the position b at the scale a. From this perspective, it is a local operator. However, by varying the parameter b along the domain of the function $f(r)$, a global description of the function can be obtained. Consider $f(r,\theta,\phi)$, a 3D binary representation for the cloud of 3D data points in spherical coordinates, which in its discrete form, can be seen as spherical voxel representation of the 3D data.

In order to analyse the distribution of the cloud of points over the space (r,θ,ϕ), the following function is used:

$$F(r)_{m,n} = \int_0^{2\pi}\int_0^{\pi} f(r,\theta,\phi)U_{m,n}(\theta,\phi)r^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi, \quad 0 \leqslant m \leqslant n \tag{1}$$

This function integrates the distribution $f(r,\theta,\phi)U_{m,n}(\theta,\phi)$ over the sphere of radius r. $U_{m,n}$, $0 \leqslant m \leqslant n$ are the set of spherical harmonics of order m and n. These functions are defined on the unit sphere and form an orthogonal family (Ferrers, 1877). Their expression is $U_{m,n} = \mathrm{e}^{jm\phi}V_n(\theta)$, where $V_n(\theta)$ is a polynomial function of order n in $\cos\theta$ and $\sin\theta$.

$F(r)_{m,n}$ $0 \leqslant m \leqslant n$, represent the projections of the distribution $f(r,\theta,\phi)$ over the space of the spherical harmonics. Therefore, they describe the spectrum of $f(r,\theta,\phi)$ with respect to θ and ϕ. To make the description of the distribution complete, we must also analyse the set $F(r)_{m,n}$ in terms of the radius r. For this, we propose wavelet-based analysis in which the function $F(r)_{m,n}$ is projected on an orthogonal family of wavelet functions. The set of projections forms a unique representation of $F(r)_{m,n}$ and therefore of the distribution $f(r,\theta,\phi)$. From that set, a group of features are selected according to the criteria described in Section 3.1.

Consider the projections of $F(r)_{m,n}$ on the family of wavelet functions $\psi_{a,b}$.

$$\begin{aligned} C_{a,b}^{m,n} &= \int_0^{\infty} F(r)_{m,n}\psi_{a,b}(r)\,\mathrm{d}r \\ &= \int_0^{\infty}\int_0^{2\pi}\int_0^{\pi} f(r,\theta,\phi)\psi_{a,b}(r)\mathrm{e}^{jm\phi}V_n(\theta)r^2 \\ &\quad\times \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\,\mathrm{d}r \end{aligned} \tag{2}$$

$C_{a,b}^{m,n}$, also called the wavelet transform coefficients, represent according to (2) the projections of the distribution $f(r,\theta,\phi)$ on the orthogonal family $L_{a,b}^{m,n} = \psi_{a,b}(r)U_{m,n}(\theta,\phi)$. It can be shown that: $\langle L_{a,b}^{m,n}, (L_{a',b'}^{m',n'})^*\rangle = K\delta_{aa'}\delta_{bb'}\delta_{mm'}\delta_{nn'}$, where $*$ denote the complex conjugate, δ is the Kroneker symbol ($\delta_{ij} = 1$ if $i = j$, 0 otherwise) and K is a constant. Therefore, the coefficients $C_{a,b}^{m,n}$ can be seen as a special type of 3D moments derived from the orthogonal family $L_{a,b}^{m,n}$. The orthogonal wavelet family we used is built with the Meyer's wavelet (Meyer, 1997). In addition to their orthogonality, Meyer's wavelet family exhibit high regularity in both the space and frequency domains. Because $C_{a,b}^{m,n}$ is a complex entity, the feature considered here is rather its norm defined as $\sqrt{\langle C_{a,b}^{m,n}, (C_{a',b'}^{m',n'})^*\rangle}$.

2.2. Feature invariance

For translation and scale invariance, the human body scan data is first rasterized into a voxel grid. Then the centre of mass of the scan data is aligned with the centre of the grid. The scale invariance is obtained by scaling the 3D points' coordinates so that the data volume defined by the moment $m_{000} = \sum_x\sum_y\sum_z f(x,y,z)$ is equal to V_0, where V_0 is a predetermined value.

For the rotational invariance, we must know that, within the scanner device, the rotation of the body has only one degree of freedom (Fig. 1(a)), that affects only the spherical coordinate ϕ. Therefore, rotational invariance has to be proved only with respect to this coordinate. Consider a rotation that changes the ϕ value by an amount γ: we can show that the coefficient related to the rotated body is $C^{m,n}_{a,b}\mathrm{e}^{jm\gamma}$. The coefficient module is therefore $\sqrt{\langle C_{a,b,m,n}\mathrm{e}^{jm\gamma}, C^*_{a,b,m,n}\mathrm{e}^{-jm\gamma}\rangle} = \sqrt{\langle C_{a,b,m,n}, C^*_{a,b,m,n}\rangle} = \|C_{a,b,m,n}\|$.

However, this property has an unwelcome aspect, namely that the symmetric postures may have close feature values. This symmetry problem is alleviated by considering a pair of symmetric postures as belonging to a single class. The correct posture can be checked afterwards using simple heuristic methods.

2.3. Feature extraction

The Cartesian voxel grid is transformed into a spherical voxel grid using the transformation $x = r\sin(\theta)\cos(\phi)$, $y = r\sin(\theta)\sin(\phi)$, $z = r\cos(\theta)$. The distribution of the 3D data points is now represented by the function $f(r,\theta,\phi)$. Because the data points' space is confined to be within a sphere of a given radius, S, and the set of features should be finite, the parameters a and b should have a finite range. Generally a dyadic discretization is adopted for the scale and shift parameters of a wavelet transform. The parameters a and b are set as follows:

$$a = S2^{-p}, \quad p = 0,1,2,3 \tag{3}$$

$$b = qa/2, \quad q = 0,1,\ldots,2^{p+1} \tag{4}$$

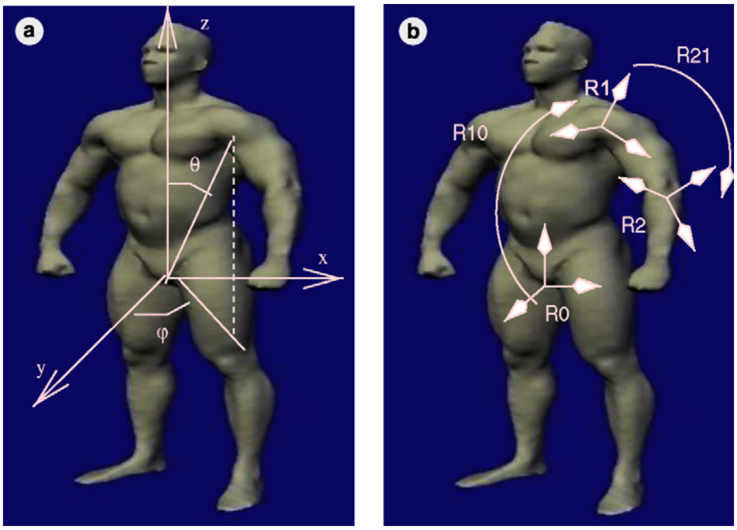


Fig. 1. (a) A standard posture of a human body in a reference frame (x,y,z) attached to the scanner. A rotation of the whole human body is constrained to be around the z axis, affecting only the angle ϕ. (b) The body parts' orientations are hierarchically defined. The rotation R21 between R1 (reference attached to the left upper arm) and R2 (reference attached to the left lower arm) defines the orientation of the lower left arm with respect to the upper left arm, and this orientation is defined by the rotation R10 between R1 and R0 (principal reference).

The scaling parameter a takes the values S, $S/2$, $S/4$, $S/8$, as scales below $S/8$ cover a very reduced space that reveals little significant information. The shifting parameter b is varied in proportion to the scale parameter and within the range $[0, S]$. This makes 34 pairs (a,b).

The first four spherical harmonic functions are used, namely, $U_{0,0} = 1$, $U_{0,1} = \cos\theta$, $U_{1,1} = e^{j\phi}\sin\theta$, and $U_{1,2} = -3e^{j\phi}\sin\theta\cos\theta$. This gives a total number of wavelet features (WC) $C^{m,n}_{p,q}$ of $34 \times 4 = 136$. However, this number will be reduced by removing the redundant features as described in the Section 3.1.

Computation of the wavelet coefficients is implemented using the Matlab Wavelet package. First, the function $F(r)_{m,n}$ (1) is calculated by means of a standard integral discretization technique. Then, the wavelet transform (2) is calculated using the Matlab *cwt* function, which approximates the continuous wavelet transform. More details can be found in the Matlab documentation.

2.4. 3D Zernike coefficient features

Zernike moments have been extensively used in 2D image analysis for their good performance with regard to noise resilience, information redundancy and reconstruction capability (Teh and Chin, 1988; Khotanzad and Hong, 1990). This was a motivation to put them into our trial and compare them with the wavelet features.

2D Zernike moments are obtained by projecting the image function on the Zernike polynomials, which form a complete orthogonal basis. These functions are complex polynomials defined over the unit disk by

$$z_{p,l}(r) = R_{p,l}(r)e^{jl\theta} \tag{5}$$

where the radial function $R_{p,l}(r)$ is defined for p and l integers with $p \geqslant l \geqslant 0$ by

$$R_{p,l}(r) = \begin{cases} \sum_{t=0}^{(p-l)/2} \dfrac{(-1)^t (p-t)!}{t!\left[\frac{1}{2}(p+l)-t\right]!\left[\frac{1}{2}(p-l)-t\right]!} r^{p-2t} \\ \text{if } p-l \text{ even} \\ 0 \\ \text{if } p-l \text{ odd} \end{cases}$$

The first few non-zero polynomials are as follows:

$$R_{0,0} = 1, \quad R_{2,2} = r^2, \quad R_{4,0} = 6r^4 - 6r^2 + 1$$

$$R_{1,1} = r, \quad R_{3,1} = 3r^2 - 2r, \quad R_{4,2} = 4r^4 - 3r^2$$

$$R_{2,0} = 2r^2 - 1, \quad R_{3,3} = r^3, \quad R_{4,4} = r^4$$

The extension of the Zernike functions to the 3D case is obtained by substituting the angular exponential function in (5) with the spherical harmonics $U^m_n(\theta, \phi)$

$$z^{m,n}_{p,l} = R_{p,l} U_{m,n}(\theta, \phi) \tag{6}$$

$z^{m,n}_{p,l}$ form a family of orthogonal functions. Indeed, it can be easily shown that $\langle z^{m,n}_{p,l}, (z^{m',n'}_{p',l'})^* \rangle = K\delta_{p,p'}\delta_{l,l'}\delta_{m,m'}\delta_{n,n'}$. By projecting the data distribution $f(r, \theta, \phi)$ on the basis $z^{m,n}_{p,l}$, we obtain a set of coefficients, called 3D Zernike coefficient features (ZC), expressed by

$$Z^{m,n}_{p,l} = \langle F(r, \theta, \phi), z^{m,n*}_{p,l} \rangle = \int_0^{\infty} \int_0^{\pi} \int_0^{2\pi} F(r, \theta, \phi) Z^{m,n*}_{p,l} \sin\theta \, d\phi \, d\theta \, dr \tag{7}$$

Like the wavelet features, the Zernike features are invariant with respect to a tilt rotation affecting the angle ϕ. By combining the first four spherical harmonics $U_{0,0}, U_{0,1}, U_{1,1}, U_{1,2}$ with the first 36 non-zero polynomials, $R_{p,l}$, 144 Zernike features are obtained. From this collection, the best discriminative features are selected using the technique described in Section 3.1.

The computation of the Zernike features was implemented using the Matlab package. The integrals in (7) are simply replaced by summations. The explicit forms of the Zernike polynomials make the discretization of that expression trivial.

2.5. 3D Fourier coefficients

In Cartesian coordinates, the 3D Fourier transform coefficients (FC) of a 3D discrete function $F(i, t, k)$ defined over the voxel grid of size N $(-N/2 \ll i, t, k \ll N/2)$, are expressed as

$$FC_{uvw} = \frac{1}{\sqrt{N^3}} \sum_{i=-N/2}^{i=N/2-1} \sum_{t=-N/2}^{t=N/2-1} \sum_{k=-N/2}^{k=N/2-1} F(i,t,k) \times \mathrm{e}^{-j\frac{2\pi}{N}(iu+tv+kw)}.$$

Theoretically the frequency parameters u, v, w have unlimited range, but in practice they are bounded in $-K \leqslant u, v, w \leqslant K$, where K depends on some prior assumption on the spectrum of the function $F(i,t,k)$. Because in our application posture changes are inferred by the movements of body limbs, and given that each limb occupies a large area of the posture space (approximately one sixth of the whole space), the spectrum of the posture data distribution is concentrated in the low frequencies. Based on this, K was set to 3.

Because we are interested in the norm of the Fourier coefficients, and taking into account the fact that the coefficients FC_{uvw} occur in complex conjugate pairs (except for FC_{000}), the number of FC features is $((2K + 1)^3 + 1)/2$, thus forming a vector of 172 features for $K = 3$. Note also that the FC coefficients are not invariant with respect to rotation. Approaches utilizing the Fourier transform must first align the data to the canonical reference, defined by the principal axes.

3. The classification

The classification problem is stated as follows. Given a set of posture classes $C_1, \ldots, C_N$ and given a query posture Q, to which class does the posture Q belong? The query posture is represented by an observation feature vector of dimension d, $X = [x_1, x_2, \ldots, x_d]$. For each class C_i, consider the discriminative functions $d_i(X)$. The observed feature vector is associated with the class C_i if $d_i(X) > d_j(X)$ for all $j \neq i$. The optimal discriminative function, in Bayes' sense, is that defined as the *posteriori* conditional probability function $P(C_i|X)$, expressed according to Bayes' rule by $P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$. Because any monotonically increasing function of $P(C_i|X)$ leads to identical classification, the following function is preferred: $d_i(X) = \ln(P(X|C_i)P(C_i))$. This expresses the separation distance as the logarithm of the product of the likelihood of the class C_i with respect to X and the *prior* probability function $P(C_i)$. Assuming that $P(X|C_i)$ is a normal distribution $\mathcal{N}(\mu_i, \Sigma_i)$ defined by $p(X|C_i) = \frac{1}{2\pi|\Sigma_i|^{1/2}} \exp[-\frac{1}{2}(X - \mu_i)^{\mathrm{T}} \Sigma_i^{-1}(X - \mu_i)]$ and that all the classes have equal *prior* probability, the expression of the discriminative function can be brought to

$$d_i(X) = -\frac{1}{2}(X - \mu_i)^{\mathrm{T}} \Sigma_i^{-1}(X - \mu_i) - \frac{1}{2}\ln|\Sigma_i| \tag{8}$$

The statistics (μ_i, Σ_i) of a class C_i are obtained from a training process using the standard EM technique (Redner and Walker, 1984).

3.1. Selection of discriminative features

Naturally, not all the features contribute effectively to the classification. To avoid redundancy, only features having reasonable discriminative power are selected. The discriminative power is assessed by the interclass distance defined as a metric for measuring the separation between two classes. A selection criterion based on that metric is therefore utilized in the search for the optimal set of features. Feature selection has been the subject of intensive work in the literature (Fukunaga, 1990). There are two main categories of technique: the first operates on feature vectors, the second treats each feature individually. We adopted a technique belonging to the second category. It is sub-optimal but relatively efficient. The selection algorithm is as follows: given a set of features $\{x_1, x_2, \ldots, x_h\}$ and given a selection criterion J,

(1) compute the selection criterion value $J(k)$ for each feature x_k, $k = 1, \ldots, h$;
(2) rank the features in descending order with respect to J; and
(3) select the top-ranked features.

There are various schemes for determining the optimal number of features to be selected. One method consists in rejecting the features for which the discriminative power criterion is below a certain lower bound (e.g., the minimum value of the separation distance between two classes). The optimal number can also be determined by means of training trials, in which the number of features is gradually increased until it reaches a value beyond which the classification performance does not improve. Section 4.2 will describe experiments illustrating this method.

3.2. The interclass distance

The selection criterion is closely related to the classification method and therefore it should be defined in the same framework. The interclass distance between two classes C_i and C_j having conditional probability density functions $P(x_k, C_i) = \mathcal{N}(\mu_i^k, \sigma_i^k)$ and $P(x_k, C_j) = N(\mu_j^k, \sigma_j^k)$ with respect to the feature x_k can be evaluated by the following probabilistic separation:

$$d_{ij}^k = \frac{1}{2}\left(\frac{\sigma_j^k}{\sigma_i^k} + \frac{\sigma_i^k}{\sigma_j^k} - 2\right) + \frac{1}{2}(\mu_i^k - \mu_j^k)^2\left(\frac{1}{(\sigma_i^k)^2} + \frac{1}{(\sigma_j^k)^2}\right) \tag{9}$$

This expression indicates that the larger the difference between the means with respect to the variances, the wider the separation between the two classes. The criterion that evaluates the discriminative power of the feature x_k is the sum of the interclass distances between each pair (9) for all the classes. Therefore, given N classes, the expression of the criterion is

$$J(k) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_{ij}^k \tag{10}$$

The larger J, the better the feature x_k discriminates between the classes.

The criterion (10) is then used to rank the three categories of feature: the wavelet features (WC), the Zernike features (ZC), and the Fourier features (FC). This process involved 32 classes corresponding to the postures shown in Fig. 3. The generation of this training data is described in Section 4.

Fig. 2(a–c) shows the criterion (10) mapped as a function of the ranked features. The variation of the mappings appears to categorize the features into two groups characterized respectively by high and low decreasing rate of the discriminative

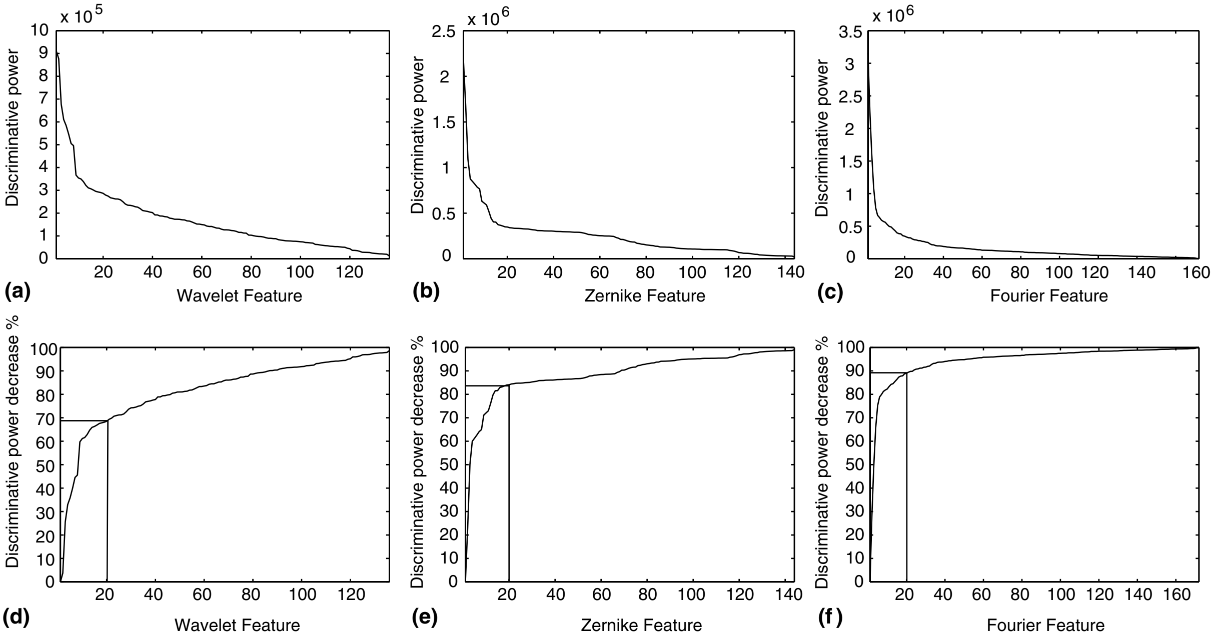


Fig. 2. The discriminative power and its rate of decrease mapped as a function of the ranked features, for the WC (a, d), ZC (b, e) and FC (c, f).

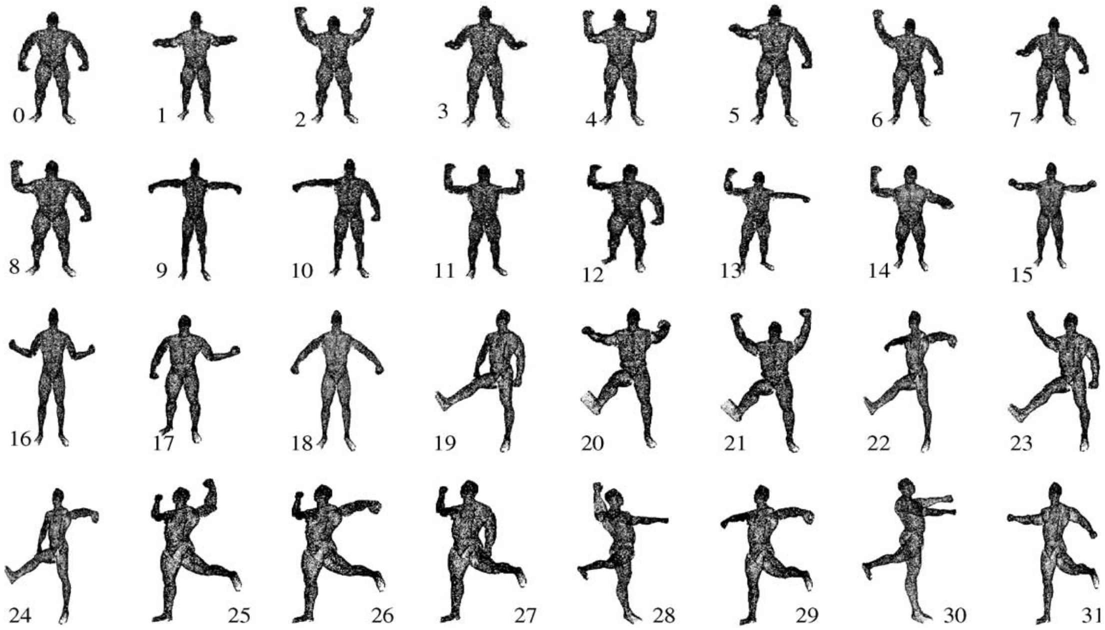


Fig. 3. The posture models labelled from 0 to 31.

power. The WC has a larger number of features in the first group compared with ZC and FC. Note that very few ZC and FC features have as high a discriminative power as WC. Fig. 2(d–f) illustrates the decreasing rate of the criterion (10) mapped as a function of the ranked features for each of the three types. The decreasing rates at the 20th feature are approximately 68%, 84% and 89% for WC, ZC and FC respectively. This shows again that the discriminative power remains reasonably high for a relatively large number of WC features, compared with ZC and FC, for which the discriminative power becomes more than 80% weaker after the 20th feature. These preliminary observations suggest that WC features are potentially more discriminative than those of ZC and FC. This was confirmed experimentally.

Table 1 shows the best 12 and the worst 12 WC features. Although the interpretation of these tables is not straightforward, some remarks can be made. For example, all the good features in Table 1(A) have a relatively large scale parameter, above *S*/8. Most of their shift parameters are around *S*/4. In spherical coordinates, this means that these features operate in areas around the sphere of radius *S*/4. These areas are indeed the most sensitive to posture changes, inferred by the gestures of the arms and legs.

For the worst features (Table 1(B)), note that they all share the same lowest scale parameter

Table 1
The best 12 WC features ranked in descending order (panel A); the worst 12 WC features ranked in ascending order (panel B)

(Panel A)												
Feature	$C^{1,1}_{2,2}$	$C^{1,1}_{3,0}$	$C^{1,1}_{0,1}$	$C^{1,1}_{3,6}$	$C^{0,0}_{2,2}$	$C^{1,1}_{3,7}$	$C^{1,1}_{3,4}$	$C^{0,1}_{2,2}$	$C^{0,0}_{3,3}$	$C^{1,1}_{3,5}$	$C^{0,0}_{2,3}$	$C^{0,0}_{1,1}$
$J \times 10e5$	9.10	8.76	6.77	6.10	5.84	5.47	5.05	4.95	3.66	3.53	3.49	3.38
(Panel B)												
Feature	$C^{0,1}_{3,11}$	$C^{1,2}_{3,14}$	$C^{1,1}_{3,14}$	$C^{1,1}_{3,16}$	$C^{0,0}_{3,14}$	$C^{1,2}_{3,12}$	$C^{1,1}_{3,12}$	$C^{0,0}_{3,12}$	$C^{1,2}_{3,16}$	$C^{1,2}_{3,11}$	$C^{1,2}_{3,15}$	$C^{1,2}_{3,14}$
$J \times 10e4$	1.22	1.95	2.02	2.11	2.15	2.25	2.34	2.47	2.73	2.79	2.80	2.85

value, namely (S/8), and that most of them have a relatively large shift parameter value close to S. This indicates that these features operate at a low scale, in the very periphery of the scan data space; therefore, they embed poor information about the posture.

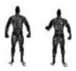
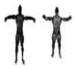
4. Experiments

A series of experiments was conducted to assess the performance of the WC, ZC and FC features in terms of power discrimination and classification rate. The experimental data consists of 32 different posture models. This set was generated as follows: a real 3D human body scan collected from the Cyberware website (http://www.cyberware.com) was fitted to a hierarchical jointed structure model satisfying the kinematics constraints of the human body. In this model, a body segment location (position and orientation) is defined relative to the upper segment in the body hierarchy. For example, the position and orientation of the right lower arm are defined with respect to a reference attached to the right upper arm (Fig. 1(b)). The relative orientations of the human body segments define the parameters that control the posture. By varying these parameters, a variety of postures with a reasonable human appearance was obtained, and 32 different posture models were generated (Fig. 3). The statistical characteristics of the posture models were determined as follows. For each posture, 30 training data sets were generated, the posture parameters of each sample were perturbed with Gaussian noise and the full data set was rotated randomly around the z axis, thus affecting the ϕ coordinate. The mean and the variance of the model vectors were computed for the 30 feature vectors associated with the training sets. This perturbing technique led to more realistic statistics than corrupting each 3D data point individually, because in real conditions, the noise in posture parameters is inferred mainly by the body's movements.

4.1. Comparison of the discriminative power

The discriminative power of the WC, ZC and FC features was assessed by testing their capabilities in discriminating close postures. For this purpose, eight pairs of close postures were selected. They are shown in Table 2 and labelled (0, 7), (2, 4), (2, 11), (3, 7), (4, 11), (8, 12), (9, 15) and (20, 21). In a first stage, the three top-ranked features, according to the criterion (10), were tested. The features' values were plotted for the 30 training samples of each posture in the pair, and thus the distributions corresponding to each pair of postures could be compared visually. Figs. 4 and 5 depict the results related to the pairs ((0, 7), (2, 4), (2, 11), (3, 7)) and ((4, 11), (8, 12), (9, 15), (20, 21)) respectively. The distribution of the WC features looks reasonably separated for all the pairs of postures except the pair (8, 12), for which the corresponding distributions are very close to each other; however, they are distinguishable and do not overlap. For the ZC features, the four pairs (2, 11), (4, 11), (9, 15) and (20, 21) show separated distributions. The pair (0, 7) shows a close distribution, whereas the distributions related to the pairs (2, 4), (3, 7) and (8, 12) overlap. The FC features show a modest separation for the pairs (4, 11) and (9, 15), and overlapping distributions for the remaining pairs.

Table 2
Pairs of close postures (first row) and their related separation distance involving the three top-ranked features of the WC, ZC and FT

	(0, 7)	(2, 4)	(2, 11)	(3, 7)	(4, 11)	(8, 12)	(9, 15)	(20, 21)
								
FC	16	2	13	0	72	54	82	6
ZC	64	3	438	8	356	7	544	136
WC	201	306	984	45	635	39	533	477

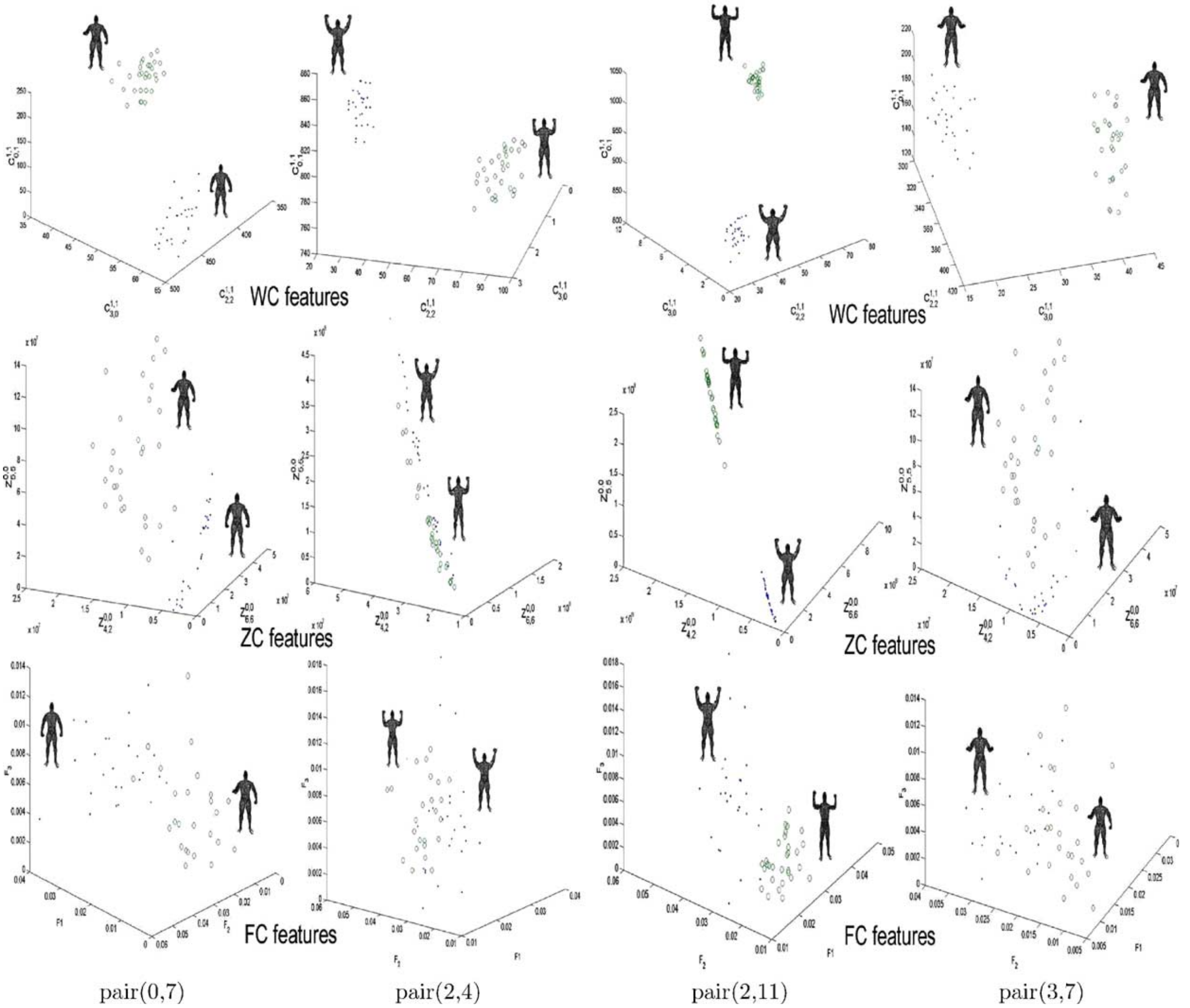
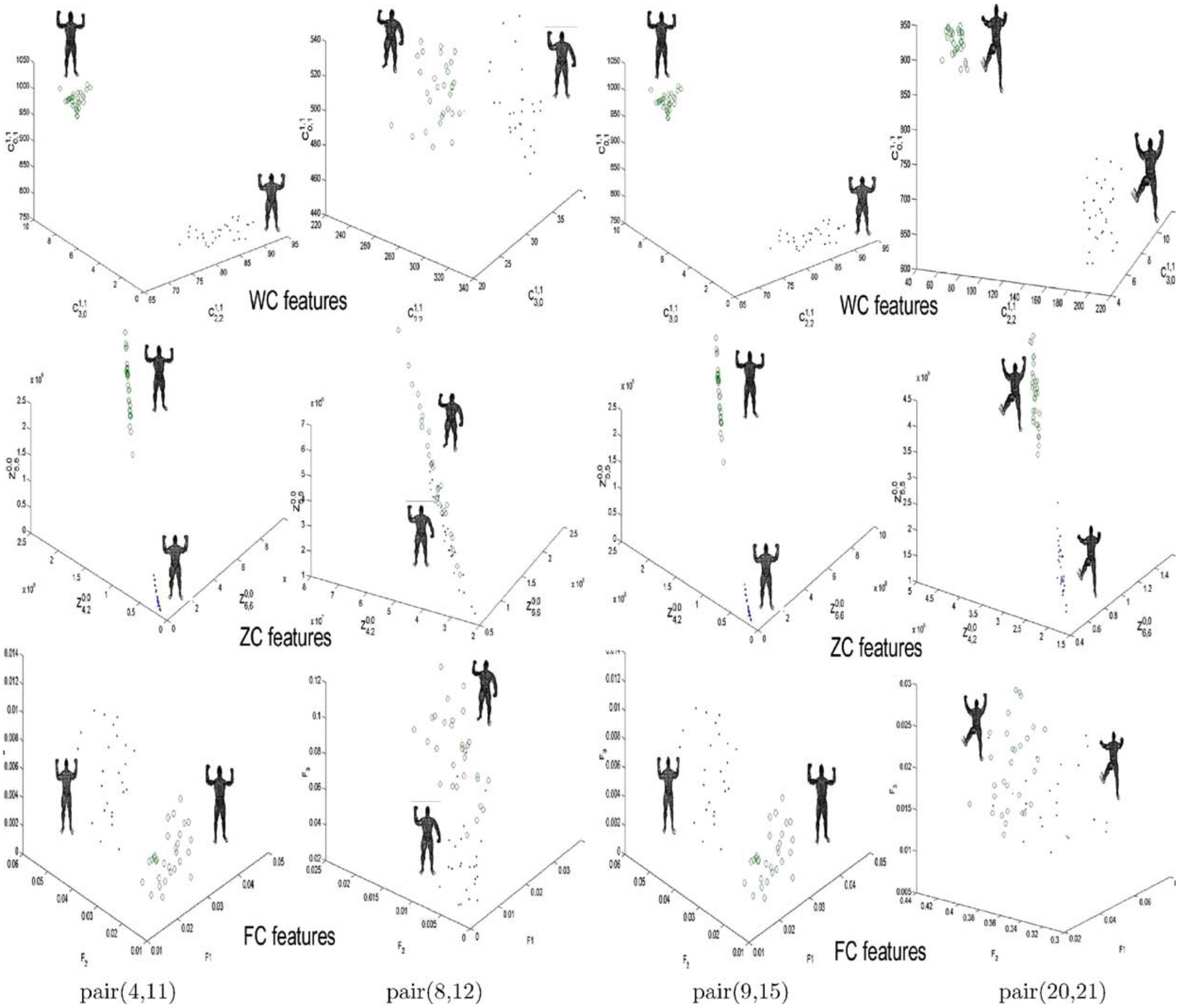


Fig. 4. Distribution of the three best WC features, ZC features and FC features for the pairs of close postures (0, 7) (2, 4), (2, 11) and (3, 7).

The above observations are confirmed by Table 2 (rows 2, 3 and 4) containing the separation distance calculated for each of the eight pairs of close postures, over the three types of feature. The separation distance: $\sum_{k=1}^{3} d_{ij}^{k}$ involved the three top-ranked features, and where d_{ij}^{k} is the distance defined in Eq. (9). In the table, we note that the WC have a much higher score than the FC or the ZC. Except for the pairs (8, 12) and (9, 15), where they are ranked second, yet close to the first ranked one.

The results of the first trial suggest that WC features appear to be more capable of distinguishing close postures than ZC and FC features. However, this judgment may not be totally fair because the three top-ranked features used in the fist trial were selected based on a criteria that was defined upon a scheme involving all the postures. Therefore, they are not necessarily the optimal features for differentiating the subset of eight pairs of close postures. Therefore, a second experiment was conducted involving the 50 top-ranked features from each



pair(4,11) pair(8,12) pair(9,15) pair(20,21)

Fig. 5. Distribution of the three best WC features, ZC features and FC features for the pairs of close postures (4, 11), (8, 12), (9, 15) and (20, 21).

category. The separation distance between close postures was computed for groups of k features, where $k = 1, 2, \ldots, 50$. The separation distance, involving k features is then $d_k = \sum_{t=1}^{k} d_{ij}^t$. Note that for this implementation, the summation form is more appropriate than a matrix form as the latter may involve the inversion of an ill-conditioned covariance matrix. Fig. 6 shows the separation distance d_k mapped as a function of the number of features for the eight pairs of postures. At first sight, the WC features appear to exhibit the best performance. The increase rate of the separation distance for the WC is clearly larger than that for the ZC and the FC. This is illustrated by the increasing gap between the WC mapping and the ZC and FC mappings.

For the five close pairs (0, 7), (2, 4), (2, 11), (3, 7) and (20, 21), the WC features exhibit the largest separation, whatever the number of features involved. For a number of features larger than 29, the WC case presents the largest separation distance for all the pairs.

The three types of feature show a similar behaviour with respect to the variation of the increase

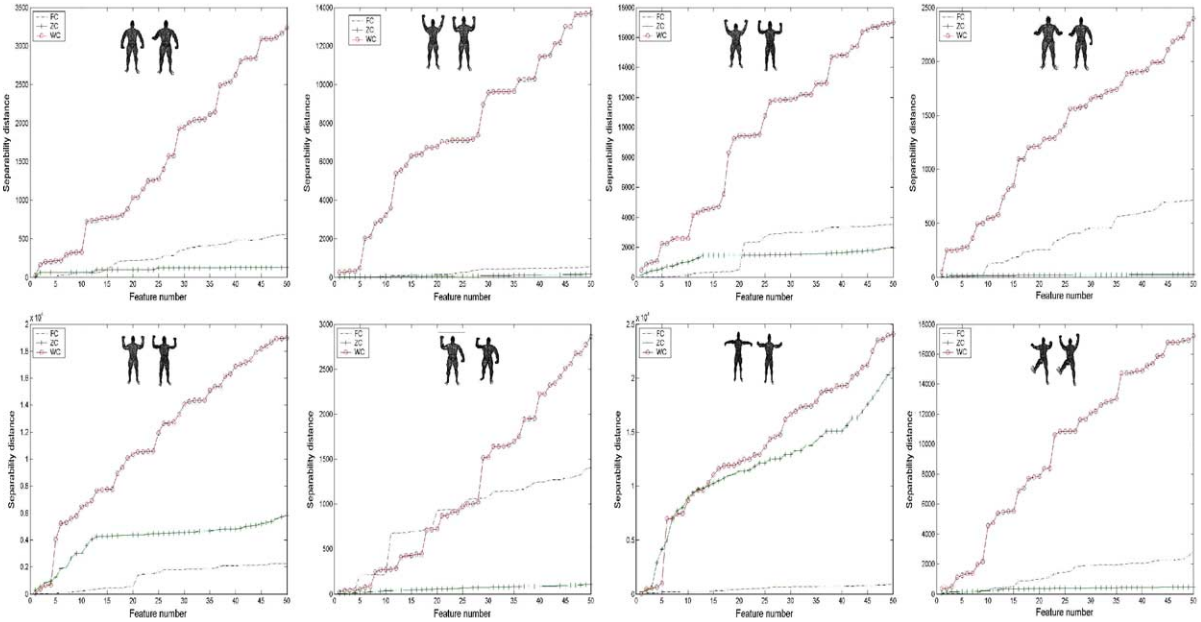


Fig. 6. Variation of the discriminative distance between the pairs of close postures (0, 7), (2, 4), (2, 11), (3, 7), (4, 11), (8, 12), (9, 15) and (20, 21) mapped as a function of the number of features.

rate of the separation distance. Instances of this behaviour are illustrated in Fig. 7 showing the increase rate corresponding to the WC, ZC and FC features for the pairs of postures (0, 7) and (2, 4). This behaviour is characterized by a fluctuating variation, which is surprising on initial examination, as one would expect a monotonic variation, based on the fact that the number of features progresses according to the ranking established according to the selection criteria. We believe that the roots of this behaviour can be traced first to the sub-optimality of the ranking progress, and second to the fact that the ranking was issued from a process that involved all the pairs of postures, and therefore, it might not be optimal for the specific postures.

4.2. Comparison of the classification rate

In these experiments, a set of query postures were matched to the posture models, and the performances of the WC, ZF and FC features were assessed by evaluating the rate of successful classifications. Query postures were obtained in the same way as the posture models; that is, by using 30 randomly perturbed and rotated versions for each artificially generated posture, producing a set of 30 × 32 query samples. These experiments aimed at comparing the feature performance and also assessing the optimality of the feature selection and ranking process described in Section 3.1. The scheme consisted of repeatedly determining the classification rate for a group of features, starting with the group of the seven top-ranked features. Then, at each trial, the number of features was incremented by one (adding the next top-ranked feature to the group), and this process was repeated until the number of features reached 50. The classification rate was then plotted as a function of the number of features, permitting examination of its evolution. The results are shown in Fig. 8. We observe that WC outperformed ZC and FC for all the features, with a maximum rate of 98% reached with 32 features. For the ZC and FC cases, the maximum rates were 90% and 86%, with 36 and 42 features respectively. We also noted an overall enhancement in the classification performance as the number of features

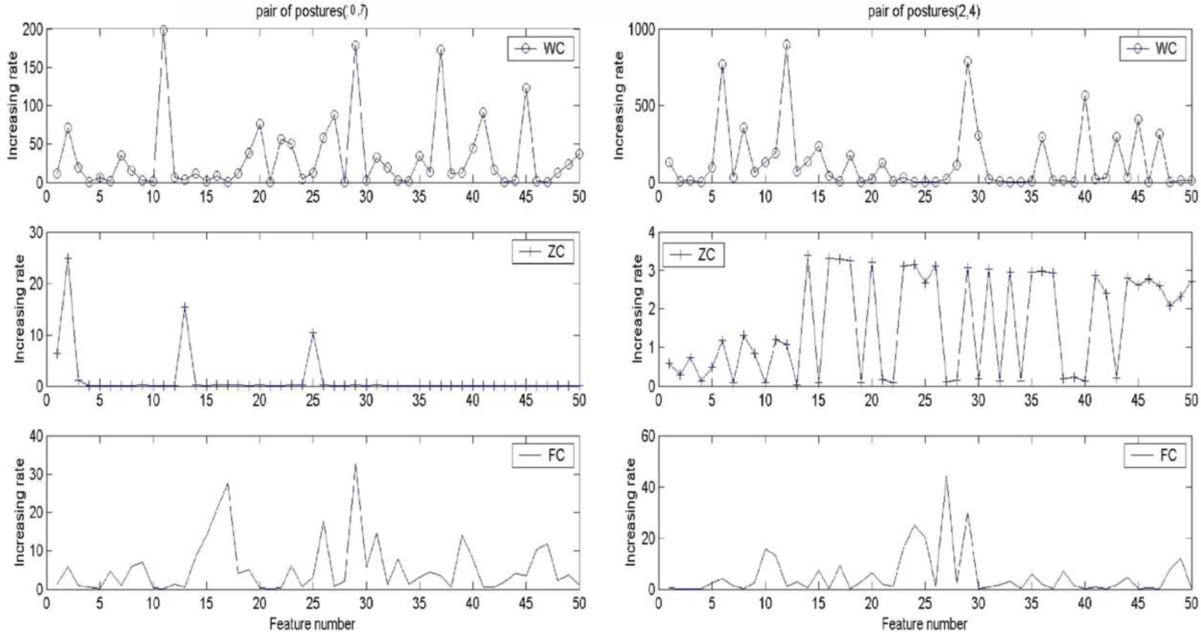


Fig. 7. Increase rate of the separation distance corresponding to the pairs of postures (0, 7) and (2, 4) for three types of feature.

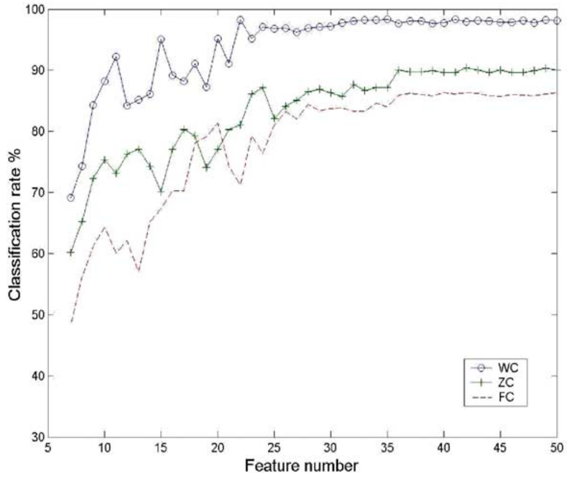


Fig. 8. Classification rate of the WC, ZC and FC features mapped as a function of the number of features.

increased. However, the classification rate variation presents fluctuations that start at the 11th feature for the WC and at the 10th for ZC and FC. This behaviour is similar to that observed in the experiments conducted on close postures in Section 4.1 (Fig. 7). This confirms the sub-optimality of the feature selection technique.

We also noted that after a certain number of features, the classification rate became stable for the three types of feature. At that stage, increasing the number of features no longer improved the performance, as the discriminative power of the features became increasingly weaker.

5. Conclusion

This work has described a methodology for recognizing human body postures from 3D scanner data. It proposes new 3D shape descriptors based on the wavelet transform. These features, exploited within a model-based approach, demonstrated high discriminative power compared with the Zernike and Fourier features. Using the three best features, WC features were able to differentiate seven of eight pairs of close postures, whereas ZC features and FC features could not differentiate more than four and one respectively. The good performance of WC was also confirmed for a larger number of features. The mapping of the separation distance as a function of the number of features shows that WC has the highest increase rate, well above those of FC and ZC. The experiments conducted on a set of 32 posture models confirmed the high performance of the WC, which achieved a top rate of 98% compared with 90% and 86% for the ZC and the FC respectively.

Naturally, the set of posture models can be enriched by a greater variety of postures. The method we adopted remains very applicable. However, a question may arise as to the number of different postures that can be recognized. We believe that this is linked first, to what extent the recognition process can differentiate between close postures, and second, to the ability to set a metric to measure the closeness of the postures. The parametric description of the posture in terms of the orientation of each body segment can be used for that purpose. What remains is to determine the minimum changes in posture parameters that would produce a distinguishable new posture. We are currently investigating this.

References

Belkassim, S.O. et al., 1991. Pattern recognition with moment invariants: A comparative study and new results. Pattern Recognition 24 (12), 1117–1138.

Canterakis, N., 1997. Fast 3D Zernike moments and invariants. Tech. Report 5/97, Institute of Informatics, University of Freiburg, Germany.

Chim, Y. et al., 1999. Character recognition using statistical moments. Image Vision Comput. 17 (3–4), 299–307.

Cordier, H., Seo, H., Magnenat-Thalmann, N., 2003. Made-to-measure technologies for an online clothing store. Comput. Graphics Appl. (January–February), 38–48.

Daubechies, I., 1990. The wavelet transform, time-frequency localization and signal analysis. IEEE Trans. Info. Theory 36 (5), 961–1005.

Dekker, L., Khan, S., West, E., Buxton, B., 1998. Models for understanding the 3D human body form. In: Proc. IEEE Workshop on Model-Based 3D Image Anal., Bombay, India, pp. 65–74.

Ferrers, N.M., 1877. An elementary treatise in spherical harmonics. MacMillan.

Fukunaga, K., 1990. Introduction to statistical pattern recognition, second ed. Academic Press, New York.

Hu, M., 1962. Visual pattern recognition by moment invariants. IRE Trans. Inform. Theory IT-8, 179–187.

Jaffard, S., Meyer, Y., Dyan, R., 2001. Wavelets: Tools for Science and Technology. SIAM, Philadelphia.

Jones, P.R.M., Rioux, M., 1997. Three dimensional surface anthropometry: Applications to human body. Optics Lasers Eng. 28 (2), 89–117.

Jones, P., Li, P., Brook-Wavel, K., West, G., 1995. Format of human body modelling from 3D body scanning. Int. J. Cloth. Sci. 7 (1), 7–16.

Khotanzad, A., Hong, Y.H., 1990. Invariant image recognition by Zernike moments. IEEE Trans. Pattern Anal. Machine Intell. 12 (5), 489–797.

Lo, C., Don, H., 1989. 3-D Moment forms: Their construction and application to object identification and positioning. IEEE Trans. Pattern Anal. Machine Intell 11 (10), 1053–1064.

Mallat, S., 1989. A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. Machine Intell 11 (7), 674–693.

Meyer, Y., 1997. Wavelets and operatorsCambridge Studies in Advanced Mathematics, Vol. 37. Cambridge University Press.

Morlet, J.M, Grossman, A., 1984. Decomposition of Hardy functions into square integrable wavelets of constant shape. SIAM J. Math. Anal. 15 (4), 723–736.

Paquet, E., Robinette, K.M., Rioux, M., 2000. Management of three-dimensional and anthropometric databases: Alexandria and Cleopatra. J. Electron. Imaging 9, 421–431.

Pargas, R., Staples, N., Davis, J., 1996. Automatic measurement extraction for apparel for three-dimensional body scan. J. Optics Laser Eng. 28 (PT2), 157–172.

Redner, R., Walker, H., 1984. Mixture densities, maximum likelihood and the EM algorithm. SIAM Rev. 26 (2), 195–239.

Sadjadi, F.A., Hall, E.L., 1980. Three-dimensional moment invariants. IEEE Trans. Pattern Anal. Machine Intell. 2 (2), 127–135.

Schael, M., 1997. Invariant 3D features. Tech. Report 4/97, Institute of Informatics, Albert-Ludwigs-Universität Freiburg.

Sheynin, S.S., Tuzikov, A.V., 2001. Explicit formulae for polyhedra moments. Pattern Recognition Lett. 22 (10), 1103–1109.

Sun, W., Hilton, A., Smith, R., Illingworth, J., 2001. Layered animation of captured data. Internat. J. Comput. Graphics 17 (8), 457–474.

Starck, J., Collins, G., Smith, R., Hilton, A., Illingworth, J., 2002. Animated statues. J. Machine Vision Appl. 14 (4), 248–259.

Teague, M., 1980. Image analysis via the general theory of moments. J. Opt. Soc. Amer. 70 (8), 920–930.

Teh, C.H., Chin, R.T., 1988. On image analysis by the methods of moments. IEEE Trans. Pattern Anal. Machine Intell. 10, 496–513.

Vranic, D.V., Saupe, D., 2001. 3D shape descriptor based on 3D Fourier transform. In: Proc. EURASIP Conf. Digital Signal Process. Multimedia Commun. Services (ECMCS 2001) Budapest, Hungary, pp. 271–274.

Werghi, N., Xiao, Y., 2002. Wavelet moments for recognizing human body posture from 3D scans. In: Proc. Internat. Conf. Pattern Recognition, Quebec City, Canada, pp. 123–126.

Xiao, Y., Werghi, N., Siebert, P., 2003. A discrete Reeb graph approach for the segmentation of human body scans. In: Proc. Internat. Conf. 3D Digital Imag. Model., Alberta, Canada, pp. 378–385.

The Mesh-LBP: A Framework for Extracting Local Binary Patterns From Discrete Manifolds

Naoufel Werghi, *Member, IEEE*, Stefano Berretti, *Member, IEEE*, and Alberto del Bimbo, *Member, IEEE*

***Abstract*—In this paper, we present a novel and original framework, which we dubbed mesh-local binary pattern (LBP), for computing local binary-like-patterns on a triangular-mesh manifold. This framework can be adapted to all the LBP variants employed in 2D image analysis. As such, it allows extending the related techniques to mesh surfaces. After describing the foundations, the construction and the main features of the mesh-LBP, we derive its possible variants and show how they can extend most of the 2D-LBP variants to the mesh manifold. In the experiments, we give evidence of the presence of the uniformity aspect in the mesh-LBP, similar to the one noticed in the 2D-LBP. We also report repeatability experiments that confirm, in particular, the rotation-invariance of mesh-LBP descriptors. Furthermore, we analyze the potential of mesh-LBP for the task of 3D texture classification of triangular-mesh surfaces collected from public data sets. Comparison with state-of-the-art surface descriptors, as well as with 2D-LBP counterparts applied on depth images, also evidences the effectiveness of the proposed framework. Finally, we illustrate the robustness of the mesh-LBP with respect to the class of mesh irregularity typical to 3D surface-digitizer scans.**

***Index Terms*—Local binary patterns, ordered ring facets, mesh manifold, 3D texture analysis.**

I. Introduction

The Local Binary Pattern (LBP) is a local shape descriptor that has been introduced by Ojala et al. [1], [2] for describing 2D textures in still images. Its computational simplicity and discriminative power attracted the attention of the image processing and pattern recognition community, and rapidly it has found other applications in visual inspection [3], [4], remote sensing [5]–[7], face recognition [8]–[11], facial expression recognition [12], and motion analysis [13], [14]. However, all the LBP-based methods developed so far operate either on photometric information provided by 2D color images or on geometric information in 2D depth images. The few solutions that extract surface features directly in 3D (typically in the form of surface normals), resort to the 2D case by converting the 3D extracted features to depth values, and then use ordinary LBP processing on depth images [15]–[17].

The triangular mesh manifold is a simple, compact and flexible format for encoding 3D shape information, which is widely used in many fields, such as animation, medical imaging, computer-aided design and many others. The recent advances in shape scanning and modeling have also allowed the integration of both photometric and geometric information into a single support defined over a 2D mesh-manifold. Despite the abundance and the richness of the mesh manifold modality, to the best of our knowledge, there is no a computational support that allows the computation of LBP on this format. One factor that plagued the development of an LBP-based description on the mesh is the lack of an intrinsic order in the triangular mesh manifold, since the LBP requires an ordered support for its computation. On the contrary, computation of LBP on 2D images benefits from the implicit ordering of the pixels in the 2D image array.

Providing such a framework for computing LBP on a mesh could be of great interest for describing 3D texture reflecting the presence of repeatable geometric patterns on the mesh surface (this being a completely separate concept from photometric texture). In fact, there are many applications that require local surface shape analysis and interpretation of 3D textured surfaces. In quality control, texture description can be used for detecting local surface pattern defection. In medicine, most of the imaging data (e.g., ultrasound, microscopic images) are shifting to a 3D mesh format. Many diagnostic rules related to these modalities need description and classification of some organs local surfaces. More generally, texture description on the mesh is useful for any application that needs 3D texture analysis, classification, and retrieval. For example, a typical scenario in the last application is to have a sample of specific 3D texture pattern and detect regions which match that model in a gallery of surfaces.

Motivated by these facts, in this paper we address the challenge of computing LBP on a mesh manifold by proposing an original computational framework, which we called mesh-LBP that allows the extraction of LBP-like patterns directly from a triangular mesh manifold, without the need of any intermediate representation in the form of depth images. With this framework, we can therefore build on the current 2D-LBP analysis methods, extending them to mesh manifolds as well as to the modality that also embeds photometric information into mesh models. To motivate our solution and to relate it to the state of the art approaches, next we provide an overview of the LBP literature, then the main features and the contribution of our proposal are discussed.

A. LBP Overview and Related Work

In its original definition, the LBP operator [1] assigns labels to image pixels by first thresholding the 3 × 3 neighborhood

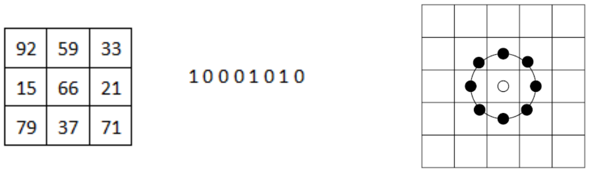
Manuscript received January 30, 2014; revised June 18, 2014; accepted November 6, 2014. Date of publication November 12, 2014; date of current version December 12, 2014. This work was supported by the National Research Foundation, United Arab Emirates, under Grant ref: UIRCA 2013-24877. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nikolaos V. Boulgouris.

N. Werghi is with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: naoufel.werghi@kustar.ac.ae).

S. Berretti and A. del Bimbo are with the Department of Information Engineering, University of Florence, Florence 50121, Italy (e-mail: stefano.berretti@unifi.it; alberto.delbimbo@unifi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2370253



92	59	33
15	66	21
79	37	71

(a) (b)

Fig. 1. (a) Computation of the basic LBP code from the 3×3 neighborhood of a central pixel. Each pixel, starting from the upper-left corner is compared with the central pixel to produce 1 if its value is greater or equal, 0 otherwise. The result is an 8-bit binary code; (b) Example of a central pixel with a circular neighborhood of a given radius.

of each pixel with the center value (i.e., each pixel in the neighborhood is regarded as 1 if its value is greater or equal to the central value, 0 otherwise), then considering the sequence of 0/1 in the pixel neighborhood as a binary number according to a positional coding convention. This is shown in Fig. 1(a), where the upper left pixel in the neighborhood is regarded as the most significant bit in the final code. This eight bits number encodes the mutual relationship between the gray levels of the central pixel and its neighboring pixels. The histogram of the numbers obtained in such a way can then be used as a texture descriptor. This operator is distinguished by its simplicity and its invariance to monotonic gray-level transformations.

An extended LBP version that can operate on circular neighborhood of different radii, also allowing sub-pixel alterations was proposed later in [2] (see Fig. 1(b)). These initial formulations led subsequently to the definition of other neighborhood variants, like the oriented elliptic neighborhood LBP (elongated LBP) proposed by Liao et al. [18], which accounts for anisotropic information, and the multiblock LBP (MB-LBP) that compares the averages of the gray level intensity of neighboring pixels rather than the value of individual pixels, in order to capture macrostructural features in the image [19]. Other versions have been proposed to improve the discriminative power of the descriptor, such as the improved LBP (ILBP) [20], in which pixel values are compared with the average of the neighborhood, and the extended LBP (ELPB) [21], which encodes, in addition to the binary comparison between pixels values, the amplitude of their difference using additive binary digits. To improve the robustness of LBP, Tan et al. [22] introduced the so-called local ternary pattern (LTP), which substitutes the original binary code by a three-values code (1, 0 and -1) by means of a user-defined threshold. This new operator addressed the sensitivity to noise, though at the cost of the invariance to monotonic gray-level transformations. A fuzzy-logic version of the LTP was proposed later in [23], where a fuzzy membership function substituted the crisp three-states association used in [22]. A more complete list and discussion on the many LBP variants appeared in the literature can be found in [24].

Considering the case of 3D shape analysis, most if not all the LBP-based approaches have been developed for face recognition applications. Many of the techniques developed in this context operate on standard depth images, where the z-coordinate is mapped to a gray-level value. This format allowed a straightforward application of the 2D-LBP operator as it was demonstrated in the pioneering work of Li et al. [25]. Later, Huang et al. [26] proposed a 3D-LBP operator that also encodes depth differences of neighboring pixels, and more recently Huang et al. [27] extended the 3D-LBP to a multiscale extended LBP (eLBP), which consists of several LBP codes in multiple layers accounting for the exact gray value differences between the central pixel and its neighbors. Sandbach et al. [15] proposed a local normal binary pattern (LNBP), which used the angle between normals at two points rather than the depth value to obtain the local binary code. Similar to this, in [16] the surface normals are extracted in 3D, then the values of the normal components along the direction of the three coordinate axes are interpreted as depth values, and LBP is computed on these depth maps reporting the values of the normal components. The idea of exploiting surface normals is further extended in [17], where azimuthal projection distance images are constructed. The azimuthal equidistant projection is able to project normals onto points in an Euclidean space according to the direction. Though the projected information is not the depth, depending on the normals of the 3D surface, 2D LBP are still computed on the projection images. Fehr and Burkhardt [28] attempted an LBP definition specifically tailored for volumetric data by sampling a sphere of a given radius around a central voxel. The approach is computationally expensive in that the rotation-invariance is addressed with complex techniques involving spherical correlation in the frequency domain.

B. Paper Contribution and Organization

From the analysis above, it emerges that since its introduction the LBP descriptor has attracted great interest for the analysis of 2D images, mainly for its simple and efficient computation and for the effective results that can be achieved relying on the LBP theory. Recently, various attempts have been done for extending the LBP framework to the case of 3D meshes, but none of them succeeded in addressing all the issues posed by the need for a simple and effective processing directly performed on a mesh-manifold. Indeed, existing solutions address the LBP extraction on 3D meshes by resorting to the easier 2D case, through the projection of 3D meshes on 2D depth maps.

In this paper, we propose a framework that we call mesh-LBP, for designing and extracting local binary patterns directly from a 2D mesh-manifold. In addition to its originality, the proposed framework is characterized by the following features:

- *Effectiveness* – The mesh-LBP operates directly on 3D triangular meshes, thus avoiding any expensive pre-processing, such as registration and normalization, required to obtain depth images;
- *Generalization* – By its ability of handling mesh data, this framework can deal with a larger spectrum of surfaces (e.g., closed, open, self-occluded) as compared to its counterpart defined on depth images;
- *Adaptability* – This framework can be adapted to hold most if not all the LBP variants proposed in the literature for 2D and depth images;
- *Simplicity* – The mesh-LBP preserves the simplicity of the original LBP, not requiring any surface parametriza-

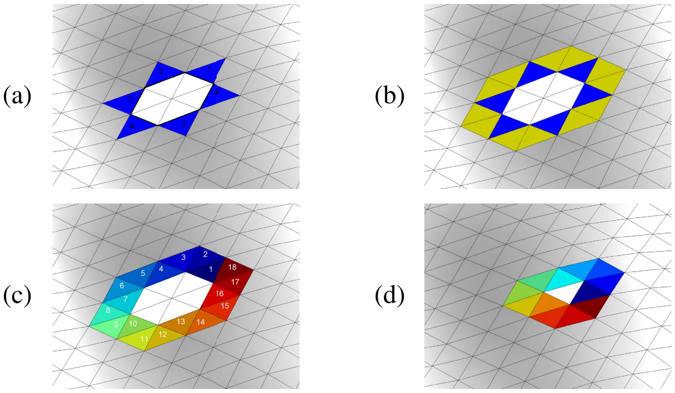


Fig. 2. Construction of an ordered ring: (a) Initial $Fout$ facets on a convex contour; (b) Bridging the gap between the pairs of consecutive $Fout$ facets with the $Fgap$ facets; (c) The obtained ordered ring; (d) Ordered ring constructed around a central facet.

tion, apart the standard mesh arrangement into facets and vertex arrays, while keeping linear computational complexity.

The rest of the paper is organized as follows: In Section II, we introduce our framework by giving the foundation of the mesh-LBP and presenting its multi-resolution extension; Some mesh-LBP variants aiming to reduce the dimensionality of the descriptor are introduced in Section III (a comprehensive view of the mesh-LBP variants is provided in the Appendix), together with solutions addressing the invariance of the descriptor, and its robustness to irregular tessellations of the mesh; Experimental evidence of the potential of the mesh-LBP in different application scenarios and in comparison to state of the art solutions is reported in Section IV; Finally, concluding remarks and future research directions are drawn in Section V.

II. The Mesh-LBP

The construction of LBP-like patterns on a mesh, first requires a scheme for constructing rings of facets around a central one and for traversing them in an ordered fashion.

Let $S = \langle V, F \rangle$ be the triangular mesh representation of an open or closed surface, where V and F are, respectively, the sets of vertices and facets of the mesh. Let us start by considering the general case of a convex contour on the mesh, which we assume regular, i.e., each vertex has a valence of six (we will show later that our framework can also cope with meshes that do not comply with this ideal case). Consider the facets that have an edge on that contour (Fig. 2(a)). We call these facets $Fout$ facets, as they seem pointing outside the contour. Let us consider also the set of facets that are one-to-one adjacent to the $Fout$ facets and which are located inside the convex contour. Each facet in this set, that we call Fin, shares with its corresponding $Fout$ facet an edge located on the convex contour. Let us assume that the $Fout$ facets are initially ordered in a circular fashion across the contour. Given that initial arrangement, we bridge the gap between each pair of consecutive $Fout$ facets, that is we extract the sequence of adjacent facets, located between the two consecutive $Fout$ facets and which share their common vertex (the vertex on the contour). We call these facets $Fgap$ facets (see Fig. 2(b)). The "Bridge" procedure reported in pseudocode in Algorithm 1 is used to compute the $Fgap$ facets. By iterating the process of bridging the gap between two consecutive $Fout$ facets with the $Fgap$ facets results in a ring of facets that are ordered in a circular fashion (see Fig. 2(c)). The resulting arrangement of the ring facets inherits the same direction (clock-wise or anti-clockwise) of the initial sequence of $Fout$ facets. The "GetRing" procedure of Algorithm 2 describes the ring construction, which is obtained by iterative calls to the "Bridge" procedure. We dubbed such obtained ordered ring, Ordered Ring Facets (ORF).

Algorithm 1 Bridge

Input: $fout_i$, $fout_{i+1}$ two consecutive $Fout$ facets sharing a vertex; fin_i facet which shares an edge with $fout_i$
Output: $Fgap_i$ set of consecutive $fgap$ facets bridging the gap between $fout_i$ and $fout_{i+1}$

```
procedure BRIDGE($fout_i, fout_{i+1}, fin_i$)
    $Fgap_i$ = [ ]
    $v \leftarrow$ vertex shared by $\langle fout_i,\ fout_{i+1} \rangle$
    $gf \leftarrow$ facet adjacent to $fout_i$, different from $fin_i$
            and containing $v$
    $prev \leftarrow fout_i$
    while $gf \neq fout_{i+1}$ do
        append $gf$ to $Fgap_i$
        $new\_gf \leftarrow$ facet adjacent to $gf$, different from $prev$
                    and containing $v$
        $prev \leftarrow gf$
        $gf \leftarrow new\_gf$
    end while
    return $Fgap_i$
end procedure
```

Algorithm 2 GetRing

Input: $Fout$, set of n ordered facets, $fout_1$, $fout_2, \ldots,$ $fout_n$, lying on a convex contour; Fin, set of n ordered facets, $fin_1, fin_2, \ldots, fin_n$, one-to-one adjacent to the $Fout$ facets and located inside the region delimited by the convex contour (depending on the contour, Fin might include duplicates)
Output: $Ring$, ring of ordered facets

```
procedure GETRING($Fout, Fin$)
    $Ring$ = [ ]
    for all $\langle fout_i, fout_{i\%n+1} \rangle$, $i \leftarrow 1, \ldots, n$ do
        append $fout_i$ to $Ring$
        $Fgap_i \leftarrow$ BRIDGE($fout_i, fout_{i\%n+1}, fin_i$)
        append $Fgap_i$ to $Ring$
    end for
    return $Ring$
end procedure
```

In the above discussion, we referred to the general case where the ORF is constructed around a convex contour. Actually, the usual case is constituted by an initial seed formed by an individual facet (central facet), whose three edges represent the initial convex contour. This case is considered in this work, since it corresponds to the situation where an ordered ring is constructed around the facets of a mesh surface. In this particular case, the $Fout$ set includes the three facets adjacent to the central one, and the obtained ring is composed of 12 ordered facets (i.e., the three $Fout$ facets, plus the nine $Fgap$ facets bridging the gap between consecutive $Fout$ facets), as shown in Fig. 2(d).

Let $h(f) : S \rightarrow \mathcal{R}$ be a scalar function defined on the mesh S (e.g., photometric data or curvature). The circular ordering of the facets obtained with ORF allows us to derive a binary pattern (i.e., sequence of 0 and 1 digits) from it, and thus to compute a local binary operator in the same way as in the standard LBP. We define the basic mesh-LBP operator at a central facet f_c by thresholding its ordered ring neighbourhood constituted by the 12 facets in the ORF:

$$meshLBP(f_c) = \sum_{k=0}^{11} s(h(f_k) - h(f_c)) \cdot \alpha(k)$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0, \end{cases} \qquad (1)$$

where $\alpha(k)$ is a weighting function. Different definitions of the function $\alpha(k)$ permit us to obtain different binary patterns, and thus different mesh-LBP values can be derived from the central facet and its ring neighborhood. For example, with $\alpha(k) = 2^k$ the basic LBP operator firstly suggested by Ojala et al. [1] is obtained; for $\alpha(k) = 1$, the sum of the digits of the pattern is computed (i.e., the number of digits equal to 1). We remark here that for the present discussion it is not necessary to detail the particular scalar function $h(f)$, whose values are computed on the mesh facets. The effect of different choices of this function will be investigated in Section IV.

A. Multi-Resolution Mesh-LBP

The mesh-LBP is extended to a multi-resolution framework by deriving a sequence of concentric rings, which preserve the ordering property. From the first ring, the sequence of facets that are one-to-one adjacent to the $Fgap$ facets are extracted (Fig. 3(a)). This sequence, which inherits the order property of the $Fgap$ facets, constitutes the set of $Fout$ facets for the subsequent ring. So, by filling the gap between each two consecutive facets of this sequence (Fig. 3(b)), a new ring, which exhibits the same ordered structure of its predecessor is obtained (Fig. 3(c)). By iterating this procedure, we build a sequence of concentric ordered rings, which represent the primitive entity for computing multi-resolution mesh-LBP (Fig. 3(d)). Details of the procedure used for computing the multi-ring structure are reported in Algorithm 3. In this case, the "GetRing" procedure of Algorithm 2 is slightly modified, so that it also returns the set of $Fgap$ facets of the current ring and the set of $Fout$ facets of the subsequent ring (indicated as $NewFout$).

It is worth mentioning that, when the regularity assumption for the mesh is satisfied, the number of facets ν across the rings evolves according to the following arithmetic progression from ring i to ring $i+1$:

$$\nu_{i+1} = \nu_i + 12. \qquad (2)$$

This can be intuitively seen referring to Fig. 3: the 1st-ring comprises 12 facets (3 $Fout$ plus 9 $Fgap$); 24 facets are included in the 2nd ring (i.e., 9 $Fout$ plus 15 $Fgap$); 36 facets in the third ring, and so on.

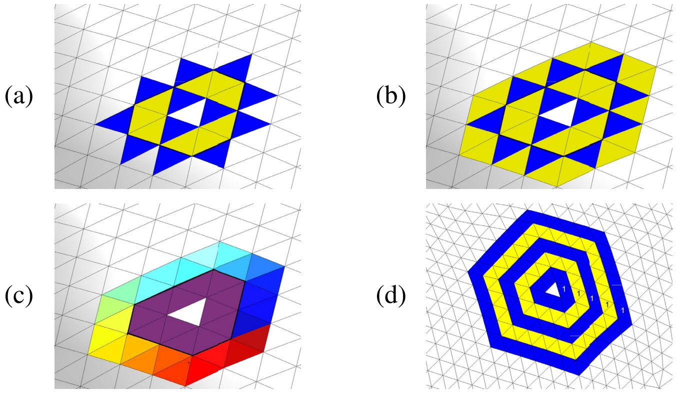


Fig. 3. Construction of multi-resolution mesh-LBP: (a) Extraction of the next set of $Fout$ facets, as the facets adjacent to $Fgap$ which are not part of the current ring; (b) Extracting the $Fgap$ facets; (c) The second ordered ring extracted; (d) Five concentric ordered rings. Notice that the first facet of each ring (marked by 1) is located at the same relative position.

Algorithm 3 MultiRing

```
Input: $Fout\_root$, initial set of ordered $Fout$ facets; $Fin\_root$, initial set of ordered $Fin$ facets one-to-one adjacent to the $Fout$ facets; $Nr$, number of rings to be constructed around $Fin\_root$
Output: $Rings$, set of $Nr$ rings of ordered facets constructed around $Fin\_root$

procedure MULTIRING($Fout\_root, Fin\_root, Nr$)
    $Rings \leftarrow [\ ]$
    $Fout \leftarrow Fout\_root$
    $Fin \leftarrow Fin\_root$
    for $i \leftarrow 1, Nr$ do
        $(Ring, NewFout, Fgap) \leftarrow$ GETRING($Fout, Fin$)
        append $Ring$ to $Rings$
        $Fout \leftarrow NewFout$
        $Fin \leftarrow Fgap$
    end for
    return $Rings$
end procedure
```

In a real mesh, because of mesh tessellation irregularities, it might happen that the "GetRing" procedure gets trapped into a closed loop resulting in $NewFout$ facets being located on the current ring or on duplicated instances. We fix such potential anomalies by simply checking the consistency of the obtained $NewFout$ facets after each iteration. However, after this post-processing procedure, the arithmetic progression of the number of facets across rings is no longer satisfied, and this latter case can be used as an indicator of the local mesh irregularity. We will elaborate further on this aspect in Section III-C.

Given a multi-ring constructed around a central facet f_c, a multi-resolution mesh-LBP operator is derived as follows:

$$meshLBP_m^r(f_c) = \sum_{k=0}^{m-1} s(h(f_k^r) - h(f_c)) \cdot \alpha(k), \qquad (3)$$

where r is the ring number, and m is the number of facets uniformly spaced on the ring. The parameters r and m control, respectively, the radial resolution and the azimuthal quantization of the operator. In principle, any predefined number of samples per ring can be used. In this work, we considered, in almost all the cases, a number of samples per ring $m = 12$.

III. Mesh-LBP Implementation

In the following, we provide more insights on the practical implementation of mesh-LBP. In particular, we propose mesh-LBP variants to reduce the descriptor size (Section III-A), together with solutions to make the mesh-LBP descriptor invariant with respect to the selection of the initial ORF facet (Section III-B), and to make it computable on meshes with non-regular tessellation (Section III-C).

A. Reducing Descriptor Size

The LBP operator produces rather long histograms and is therefore difficult to use as a region descriptor. A first solution to this problem was obtained by using just "uniform" patterns (i.e., binary patterns with a number of bitwise 0-1 transitions equal at most to 2) instead of all the possible ones [2].

The problem of reducing the dimensionality of the LBP descriptor also inspired the LBP variant called *center-symmetric* (CSLBP) [29], which modifies the pixels comparison scheme by computing the difference between center-symmetric pairs of pixels rather than comparing each pixel with the central pixel. This halves the number of comparisons for the same number of neighbors. In the context of mesh-LBP, the same result can be obtained using the following equation for the center symmetric mesh-LBP (mesh-CSLBP):

$$meshCSLBP_m^r(f_c) = \sum_{k=0}^{m/2-1} s(h(f_k^r) - h(f_{k+m/2}^r)) \cdot \alpha(k). \tag{4}$$

This is illustrated in the case (d) of Table II in the Appendix. In the experiments, we show the existence of the uniformity aspect in the mesh-LBP patterns, and the capability of the mesh-CSLBP of keeping virtually the same results than the basic mesh-LBP, while reducing the computational cost.

B. Achieving Invariance to Facets Ordering

In order to make the mesh-LBP invariant to the ordering of the facets in the ring and its traversal, two aspects should be addressed: The position of the first facet (i.e., the first $Fout$ facet) in the ring, that is from which of the facets the ring starts from; The direction of the ring traversal (clock-wise or anti-clockwise). The last aspect can be easily fixed by orienting the normals of the mesh-manifold. For the first aspect, when the ORF are constructed around a central facet, three different orderings of the facets inside each ring can be obtained, depending from which of the three $Fout$ facets, adjacent to the central facet, the first ring starts from. Therefore three different patterns can be derived from each ring. To address this ambiguity several solutions can be used:

- *Method-1:* Performing a circular bit-wise shift of the binary pattern, as was suggested in the standard LBP [2], and selecting as initial facet that resulting in the minimum LBP value. However, this method reduces the range of the LBP values and might seriously affect the discriminative power of the operator [30];
- *Method-2:* Adopting intrinsically rotation invariant descriptors only. This set includes the number of transitions, the number of 1-valued bits (i.e., the sum of the binary digits obtained when using $\alpha(k) = 1$ variant), and the number of 1-valued runs of a given length in the binary patterns. This method preserves the range of the LBP values, yet might still compromise the discrimination power, though to a less extent than the first method;
- *Method-3:* Considering all the binary pattern values that originate by moving the initial facet along the ring, but this solution creates redundancy and further burden the computation;
- *Method-4:* Selecting the first facet with respect to a local reference frame (LRF) determined based on the local morphology of the ring neighborhood. For this purpose, the method proposed by Tombari et al. [31], which ensures a unique and unambiguous LRF can be used. Afterwards, the nearest facet to the x or y axis of the LRF can be selected as the first facet.

From the above, the *method*-4 looks the most reliable and generic, but its implementation requires histograms construction, which might burden the computational complexity. For this reason, we rather adopted a simpler yet practical solution, tailored to our problem, and which consists of the following steps: (i) First, we generate the sequence of rings starting from any arbitrary adjacent facet to the central facet; (ii) Then, from the obtained sequence of ordered rings, we select as a first facet in each ring-*r*, the facet f_i which satisfies the following condition:

$$\min_i \; dist(c_o, c_i^r), \quad f_i \in \text{ring-}r, \tag{5}$$

where $dist(.)$ is the Euclidean distance, c_i^r is the center of facet f_i in the ring-*r* (union of the $Fout$ and $Fgap$ facets), and c_o is the centroid of the centers of the facets in the rings weighted by their area; (iii) Finally, in each ring, we apply a circular shifting to the current facets ordering to bring the facet selected in step *(ii)* to the first position.

Fig. 4 shows the mesh-LBP maps obtained with the *method*-1 and *method*-2 (the number of 1-valued bits in the pattern has been used) listed above, and our proposed method for selecting the first facet of a ring. The repeatability and behavior obtained using the different methods can be appreciated. In particular, the zoomed maps in Fig. 4(b), obtained for a rectangular region at the base of the nose, show a clear overall repeatability of the mesh-LBP (last column) obtained with the proposed method. The minor disparities between the three instances emanate from the mesh variability across the scans, which in turn affects to some extent the binary patterns. The same behavior is observed for the *method*-1 and -2. In particular, we notice the reduced range of the pattern values in *method*-1. For *method*-2, we can notice the limited description ability reflected in the similar values observed at the curve sides. On the opposite, our method looks the most effective in detecting the shape variability at that neighbors.

C. Mesh Quality Assessment

One issue that can hamper the repeatability of the mesh-LBP is the local irregularity of mesh tessellation, for which the assumption of vertex valence of six does not hold,

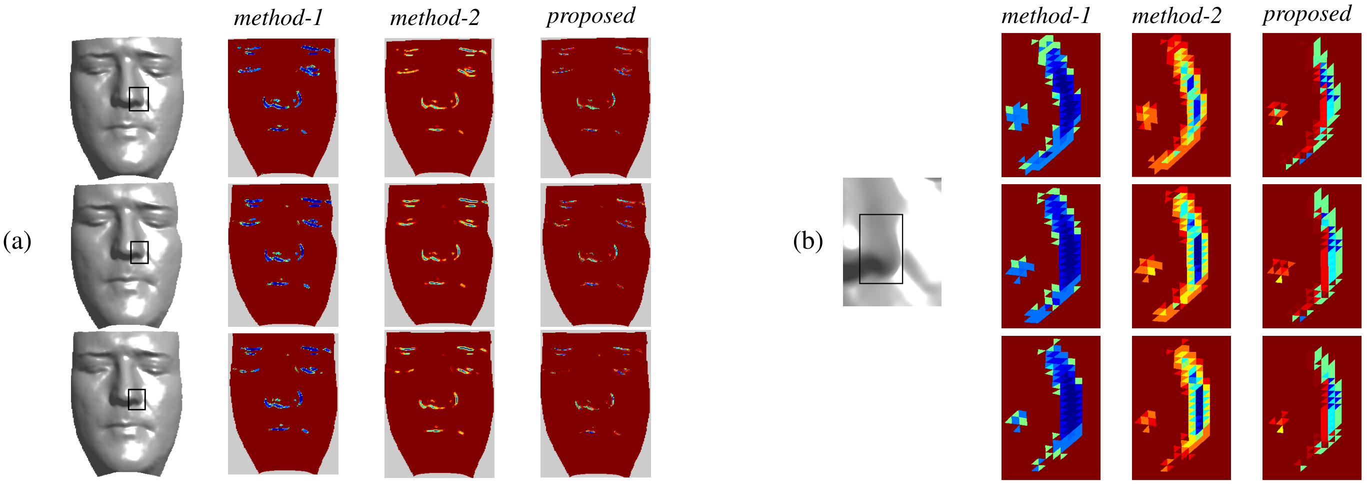


Fig. 4. Comparison of the mesh-LBP maps obtained with different methods for selecting the first facet of a ring ($r = 1$ and $m = 12$ are used). The maps shown represent the face surface mesh after coloring each facet in the mesh with a color representing its mesh-LBP value. (a) mesh-LBP maps obtained using, respectively, *method*-1, *method*-2 (number of 1s), and our proposed method, on three different face scans of a same subject; (b) A region at the base of the nose of each scan in (a) is cropped (rectangular region framed in black), and the corresponding mesh-LBP maps are zoomed in. (The maps are best viewed on the soft-copy version).

and consequently the regular progression of Eq. (2) is not satisfied. This issue can be addressed in different ways:

- Adding a pre-processing stage that regularizes the density of the mesh triangulation;
- Deriving iso-geodesic contours from the ordered rings that act as a support region for computing mesh-LBP operators;
- Applying the local density invariant smoothing, proposed by Darom and Keller [32] to the ring vertices around the central facet.

In our experiments, we rather used a simpler technique that interpolates the scalar function used to compute mesh-LBP across each ring, so as to obtain a sequence of samples that matches the ideal progression.

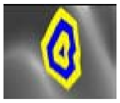
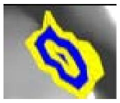
We note that the progression of the number of facets across the ordered rings (see Eq. (2)), also allows establishing a simple criteria for assessing the local regularity of a triangular mesh. Indeed, given a facet neighborhood comprising r rings, we define the *local irregularity criterion* by:

$$\delta_r = \frac{\|\boldsymbol{\Gamma}_r - \hat{\boldsymbol{\Gamma}}_r\|}{\|\hat{\boldsymbol{\Gamma}}_r\|}, \tag{6}$$

where $\hat{\boldsymbol{\Gamma}}_r$ is a vector representing the ideal sequence of the number of facets across an r-ring ORF (i.e., [12, 24, ..., 12r]) according to the arithmetic progression of Eq. (2), and $\boldsymbol{\Gamma}_r$ is the actual sequence. Fig. 5(a) depicts examples of 3-ring ORF exhibiting different $\boldsymbol{\Gamma}_3$ and δ_3.

Intuitively, the idea behind the δ_r coefficient is that the greater is the relative deviation between the actual number of facets across the r rings with respect to its ideal number, the more the mesh is irregular in the local surface spanned by these r rings computed around a central facet. This criterion can be used to assess the local regularity of a mesh, thus to regularize the support region used in the computation of the multi-resolution mesh-LBP. Fig. 5(b)-(c) depict, respectively, a mesh sample and its corresponding map with the values of δ_r originated using the local irregularity criterion. In the δ_r map in (c), dark areas correspond to larger values of δ_r; it can be observed that dark areas correspond well to the most irregular regions of the facial mesh in (b) (see, for instance, the left nostril or the right eye).

(a)

sample			
$\boldsymbol{\Gamma}_3$	[12,24,36]	[13,26,38]	[15,30,47]
δ_3	0	0.066	0.287

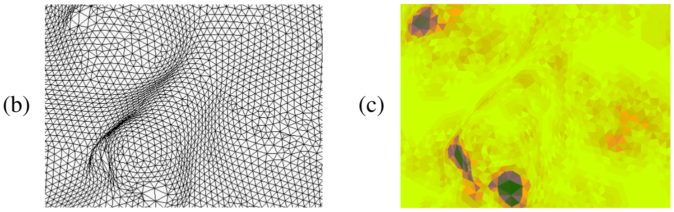


Fig. 5. (a) Examples of 3-ring ORF with their related Γ_3 and δ_3; (b) Sample of a facial mesh showing local irregularities in the eye and nose regions; (c) Corresponding map obtained by computing the local irregularity criterion δ_r at each facet.

With this criteria, once an irregular mesh region is detected, a local mesh regularization approach can be applied to it so as to recover the ideal mesh tessellation for mesh-LBP computation. Using an opposite perspective, the value of δ_r computed for the r-ring neighborhood of a facet can be used as a criteria to assess the significance of the mesh-LBP computed for the facet. According to this, $1-\delta_r$ could be used to weight the contribution of individual mesh-LBP values accumulated in a global histogram descriptor: the more irregular the mesh is in a facet neighborhood, the lower the contribution of the corresponding mesh-LBP to the overall descriptor. In the experiments, we found that, even without recurring to this procedure, the mesh-LBP can actually cope to a large extent with mesh irregularities.

IV. Experimental Results

Experiments have two main goals: On the one hand, we investigate the basic properties of the mesh-LBP descriptor,

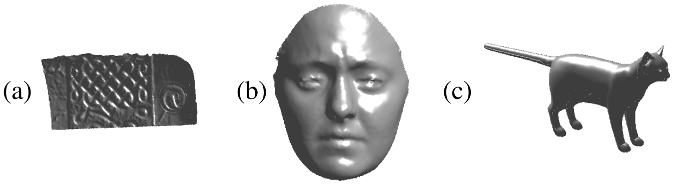


Fig. 6. Mesh models used in the uniformity experiment: (a) Portion of a pot (MIT CSAIL textured 3D models); (b) Face surface (BU-3DFE); (c) Cat model (TOSCA high-resolution).

evidencing the presence of mesh-LBP uniform patterns (Section IV-A) and the repeatability of the descriptor (Section IV-B); On the other, we evaluated the proposed framework on the specific task of 3D texture classification: first, we compare the different mesh-LBP surface functions and operators, also in comparison with some of the mesh-LBP variants (Section IV-C); then, we provide a comparative analysis of the mesh-LBP descriptor with respect to state of the art solutions which describe 3D meshes through surface descriptors, or by applying the standard 2D-LBP on depth images of the 3D meshes (Section IV-D); finally, the robustness of the mesh-LBP to mesh irregularities is also shown (see Section IV-E).

A. Uniform Patterns

By studying the statistics of the number of bitwise 0-1 transitions in the binary patterns, Ojala et al. [2] noticed that the majority of the patterns in textured 2D images have a number of transitions U equal at most to 2. These patterns are called “uniform”. In our investigation, we considered a representative set of three surface meshes collected from different sources. The first surface is a portion of a pot object from the “MIT CSAIL textured 3D models database” [33]. This object exhibits textured shape patterns on the surface. The second surface represents a face scan from the “Binghamton University 3D facial expression database” (BU-3DFE) [34], and shows the case of an open surface. The third one is a closed surface of a cat model from the “TOSCA high-resolution database” [35]. These models are shown in Fig. 6, from (a) to (c), respectively.

Four scalar functions ($h(.)$ in Eq. (3)) on the mesh manifold have been studied, namely, the *mean* curvature (H), the *gaussian* curvature (K), the *curvedness* (C), and the *angle between facets normal* (D). For each of these functions, we computed the number of transitions U in the binary patterns computed by using the mesh-LBP operator of Eq. (3), across six levels of spatial resolution (r from 1 to 6), and using 12 samples for the azimuthal quantization ($m = 12$ at each r).

The results, depicted in Fig. 7, show the percentage of facets, exhibiting a number of transitions U less or equal than 4. We can observe that this number exceeds 90% up to the third ring, across the four scalar functions, for all the three surfaces. The angle between normals is the function exhibiting the largest score with an overall percentage above 80%. The mean curvature and the curvedness show virtually the same rates. Overall, all the scalar functions show a percentage of $U \leq 4$ above 70%. These observations provide evidence on the existence of a “uniformity” aspect of the mesh-LBP computed on triangular mesh manifolds, and thus the mesh-LBP has the potential of adapting to the uniformity-driven description suggested by Ojala et al. [2]. Based on the obtained results, considering an azimuthal quantization of $m = 12$, that is 4096 possible patterns, we define the set of uniform patterns as the set including all binary patterns for which U is at most equal to 4. This set contains exactly 1123 patterns against 2973 for the non-uniform patterns. Following the same partition scheme of [2], where all the non-uniform patterns are grouped into a single label, whereas a separate label is assigned to each non-uniform pattern, the number of labels (or classes) is reduced to 1234 for our mesh-LBP. We will adopt this partition in the rest of the experiments. Notably, this partition will be used for the mesh-LBP operator involving $\alpha(k) = \alpha_2(k) = 2^k$. For $\alpha_1(k) = 1$ the distinction into uniform/non-uniform patterns does not make too much sense since the number of patterns is already small (13 patterns exactly).

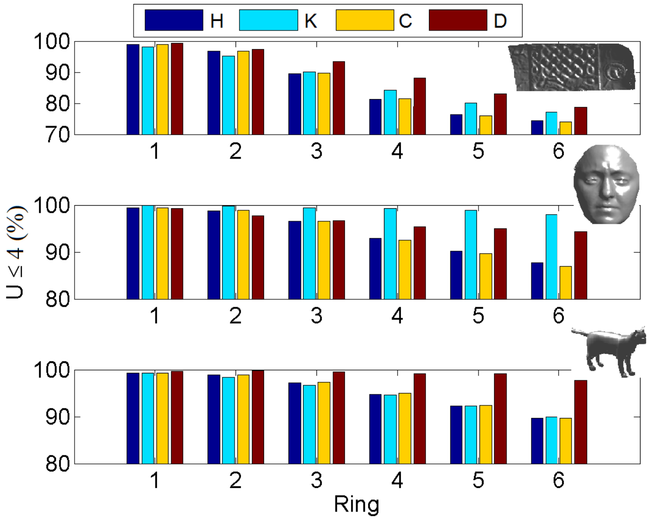


Fig. 7. Percentage of facets whose mesh-LBP have a number of transitions U less than or equal to 4 (legend: H - Mean curvature; K - Gaussian curvature; C - Curvedness; D - Angle between facets normals).

B. Repeatability

Repeatability of mesh-LBP measures the capability of the descriptor to assume comparable values when extracted from corresponding facets of different instances (i.e., scans) of a same 3D object. For this experiment, we acquired 32 facial scans of a same subject with neutral or moderate facial expressions. The four scalar surface functions reported in the previous Section, namely, *mean* curvature, *gaussian* curvature, *curvedness* and *angle between facets normal* have been used for computing mesh-LBP. For each of these functions, we considered two different mesh-LBP operators, that is, $\alpha_1(k) = 1$, $\alpha_2(k) = 2^k$. A third mesh-LBP representation has been obtained by applying the $\alpha_2(k)$ operator just to the uniform patterns (i.e., according to the results of Section IV-A, we considered a pattern uniform if its number of transitions U is $U \leq 4$). Different spatial resolutions corresponding to eight rings $r = 1, \ldots, 8$, have been also accounted. To compute the repeatability of mesh-LBP we followed an approach similar to that proposed in [36] for 3D keypoints. With this solution, first a scan is selected as *reference*, and each of the other scans (*probe*) is aligned to the reference one using ICP registration.

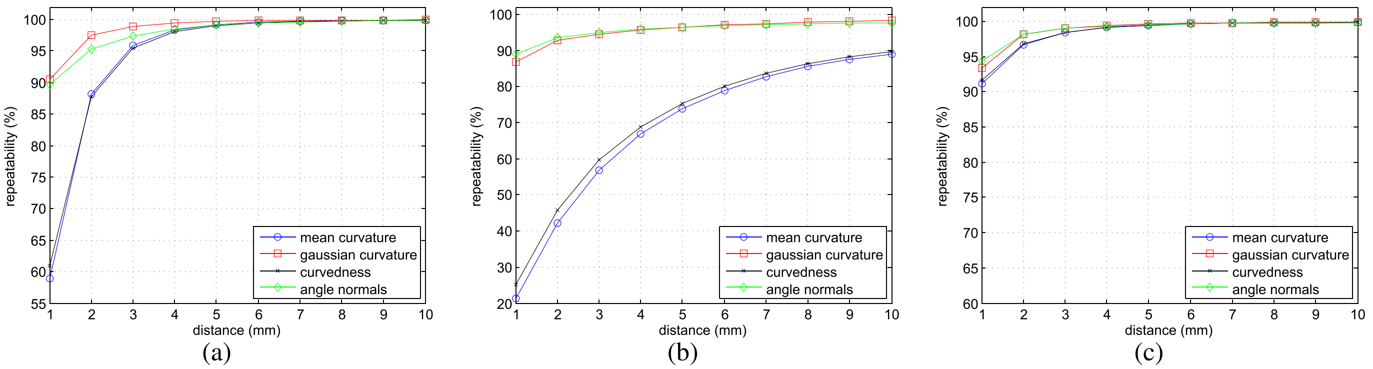


Fig. 8. Repeatability of mesh-LBP: (a) $\alpha_1(k) = 1$; (b) $\alpha_2(k) = 2^k$; (c) $\alpha_2(k) = 2^k$ applied to uniform patterns (i.e., number of transitions $U \leq 4$).

Then, for each facet in the probe, the nearest neighbor facet in the reference is found, whose mesh-LBP value is equal to the mesh-LBP value of the probe facet (the nearest neighbor distance between facets is computed between the 3D coordinates of their centroid). This operation is repeated for each facet in the probe and the distances of the nearest neighbor facets in the reference computed as above are recorded. Varying a proximity radius around the facets, it is possible to count the percentage of repeated mesh-LBP values between probe and reference scans for each value of the radius. The overall repeatability is finally obtained by iteratively using one of the scan as reference, and all the remaining as probes.

Figs. 8(a)-(c) show the obtained average repeatability as a function of increasing values of the proximity radius, respectively, for the three used mesh-LBP descriptors. The plots reported in the figure concern the mesh-LBP computed on the 1st-ring, but a similar behaviour resulted for the rings at increasing values of r. In general, we observe that the gaussian curvature and the angle between facets normal show a similar behaviour, obtaining the highest repeatability in all the cases. The mean curvature and curvedness, instead, score similar results each other, showing a lower performance especially for the α_1 and α_2 operators. Interestingly, for all the scalar surface functions, the best repeatability is obtained for the uniform patterns U (see the plot (c) in the figure).

C. Discriminating 3D Texture Patterns

2D-LBP has been successfully used in a number of different applications, the most notables being texture classification and face recognition. We have shown that mesh-LBP inherits many of the positive aspects of the standard LBP, further extending the range of possible applications to the direct analysis of 3D triangular meshes. As a consequence, it is expected that mesh-LBP can found application in a number of 3D scenarios, inspiring also new one. In the following, we focus on the problem of 3D texture classification. We remark here that in this study textures are intended as 3D repeatable patterns corrugating the object surface; This concept is completely different and separated from the 2D texture, which is related to the photometric appearance of the model and, if present, is coded by a 2D image. In fact, 3D objects have been analyzed for classification and retrieval purposes mainly using their 3D shape. This is largely motivated by the almost complete absence of 3D textures in CAD and synthetic models used in the majority of benchmark datasets [35], [37], [38]. Instead, the 3D surface texture is of fundamental importance to discriminate the 3D scans of real objects, which can show very similar shapes, but be well differentiable based on their 3D texture.

According to these considerations, in this experiment we investigate the potential of the mesh-LBP for discriminating texture patterns on 3D meshes. In so doing, our goal is to probe the capability of mesh-LBP as a framework for 3D texture classification, rather than to elaborate a proper method for such task. For this purpose, we used surface samples exhibiting a variety of 3D shape textures, collected from eight different object models of the "MIT CSAIL textured 3D models database" [33]. These objects are *bagel*, *bird*, *gargoyle*, *head*, *lion*, *owl*, *plaque* and *pot*. All these models are characterized by a reasonably uniform mesh, and we were able to identify 10 distinct 3D texture patterns from them, as reported in the 1st row of Fig. 9 (in particular, three texture patterns were derived from the *owl* object). For each sample, we computed a 1D-histogram of the mesh-LBP operator (Eq. (3)) using the operator functions $\alpha_1(k) = 1$ and $\alpha_2(k) = 2^k$, a varying spatial resolution $r = 1, \ldots, 7$, and an azimuthal quantization $m = 12$. For the operator function α_1, the resulting mesh-LBP take values in [0,12] (i.e., in this case, the number of 1-valued bits in a pattern of 12 bits is counted), and these values are accumulated in a 1D histogram with 13 bins for each ring. For the α_2 operator, for which the range of mesh-LBP is [0,4095], we adopted the uniform/non-uniform mesh-LBP partition described in Section IV-A, that is 1123 bins are used for the uniform patterns, i.e., one bin for each of the patterns having a number of transitions equal at most to four, and one bin for all the remaining patterns (the 2973 non-uniform ones). Based on this setting, two 2D histograms of size (7,13) and (7,1124) are computed for each texture, which are associated, respectively, with the α_1 and α_2 operators. The histograms are computed for each sample surface by considering an area of 19 rings around the central facets in the computation of mesh-LBP, which is sufficient for covering the 3D texture variation in each sample. To compute the distance between two histograms H_1 and H_2, the complement of the Bhattacharyya coefficient $B(.)$, i.e., $\sqrt{1 - B(H_1, H_2)}$ was used.

We repeated the histogram computation for each model using four scalar surface functions, namely, the *mean*

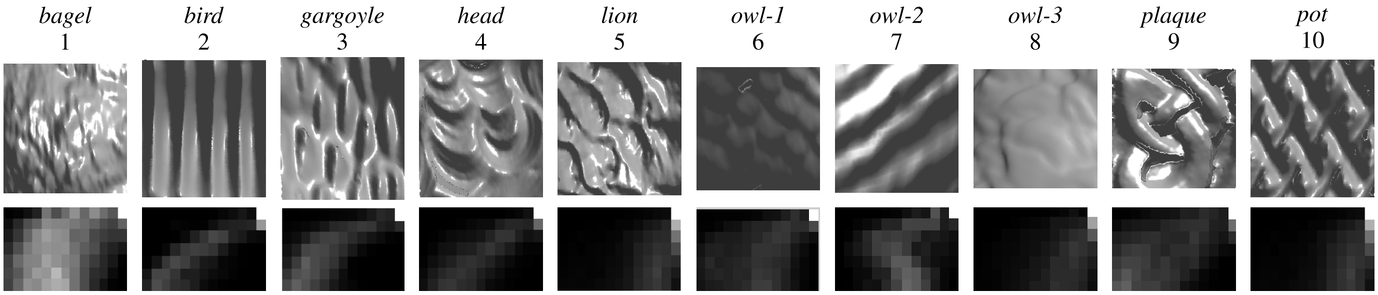


Fig. 9. Top: 3D texture samples from the ten classes. Bottom: The corresponding histograms obtained with the *angle between facets normal* and the α_1 weighting function using 7 rings and 12 samples per ring (i.e., histograms with 7 rows and 13 columns). Each histogram bin cumulates the frequency of a mesh-LBP pattern computed for all the facets of a sample surface (histograms are represented as gray-level images, where lighter pixels correspond to histogram bins with higher values).

curvature, the *gaussian* curvature, the *shape index* (instead of the *curvedness*) and the *angle between facets normal*. As an example, Fig. 9 (2nd row) depicts the histograms of the first type (i.e., α_1 operator) obtained with the *angle between facets normal*, and computed for the sample surfaces in the first row. The histograms are obtained by reporting the frequency of the mesh-LBP patterns computed for all the facets of the sample surfaces (i.e., histograms are represented as gray level images, where lighter pixels correspond to histogram bins with higher values).

The assessment of the discriminative power of the different descriptors is performed as follows. For each texture class, we considered 30 different instances and for each of them the different descriptors have been computed. From the set associated to each texture class, we evaluate the mean and the variance. Since all the descriptors have a histogram structure, the variance we consider here is the variance of the Bhattacharyya distances between descriptor instances and their mean. For each descriptor, we compute the distance matrix of the ten texture classes, where each diagonal term is the mean intra-class distance, and the non-diagonal term is the distance between the mean of class i and the mean of class j. The so-obtained 10×10 distance matrices provide a coarse assessment of the discriminative power of the descriptors.

Figs. 10 and 11 depict, respectively, the distance matrices related to the different mesh-LBP surface descriptors for $\alpha_1(k) = 1$ and $\alpha_2(k) = 2^k$. For the mesh-LBP descriptor, we notice that the intra-class distance is quite below the inter-class distance across all the different descriptors and the two operator functions. To evidence this behavior, in the confusion matrices reported for the different cases, we highlighted the intra-class and inter-class distances that are less separated (in gray and yellow, respectively), and so that are more susceptible to be confused with each other. Even in the worst cases, it can be observed that the ratio between the inter-class distances and the corresponding intra-class distance is greater than 2.33, for α_1 and SI, and of 3.37 for α_2 and SI. This is a clear indication of the potential and the appropriateness of the mesh-LBP descriptors for discriminating textured shapes.

Fig. 12 reports the distance matrices between all the classes' instances (i.e., 30 instances for each of the 10 classes). Results for the mesh-LBP computed with the scalar functions H, K, SI and D, for the α_1 and α_2 operators are depicted in the top and bottom row, respectively. In the mesh-LBP distance matrices, we can easily distinguish the 30×30 blocks related to the inter-class distances between class pairs. This observation confirms the discriminant capability of the mesh-LBP descriptors. The classification accuracy, estimated as the percentage of occurrences where the inter-class distance is greater than the intra-class distance across all the classes is also reported for each descriptor on top of the distance matrices. A perfect classification of 100% is obtained in all the cases.

H

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.13	0.08	0.08	0.05	0.10	0.07	0.06	0.08	0.07
2	-	0.03	0.13	0.13	0.13	0.17	0.13	0.14	0.14	0.12
3	-	-	0.01	0.05	0.06	0.13	0.07	0.08	0.06	0.06
4	-	-	-	0.01	0.07	0.14	0.07	0.09	0.06	0.06
5	-	-	-	-	0.01	0.11	0.06	**0.05**	0.07	0.05
6	-	-	-	-	-	0.02	0.10	0.09	0.13	0.12
7	-	-	-	-	-	-	0.01	0.07	0.08	0.06
8	-	-	-	-	-	-	-	**0.02**	0.08	0.07
9	-	-	-	-	-	-	-	-	0.02	0.06
10	-	-	-	-	-	-	-	-	-	0.01

K

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.16	0.16	0.07	0.35	0.49	0.66	0.23	0.07	0.19
2	-	0.02	0.12	0.16	0.40	0.54	0.69	0.32	0.16	0.16
3	-	-	0.01	0.16	0.43	0.55	0.70	0.35	0.15	**0.06**
4	-	-	-	0.02	0.34	0.49	0.66	0.23	0.07	0.19
5	-	-	-	-	0.02	0.25	0.51	0.18	0.36	0.45
6	-	-	-	-	-	0.02	0.28	0.38	0.51	0.56
7	-	-	-	-	-	-	0.02	0.60	0.68	0.69
8	-	-	-	-	-	-	-	0.01	0.25	0.37
9	-	-	-	-	-	-	-	-	0.01	0.17
10	-	-	-	-	-	-	-	-	-	**0.02**

SI

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.11	0.06	0.07	0.04	**0.07**	0.05	0.04	0.06	0.07
2	-	0.03	0.13	0.12	0.11	0.14	0.12	0.12	0.11	0.13
3	-	-	0.01	0.06	0.06	**0.07**	0.06	0.07	0.05	0.06
4	-	-	-	0.01	0.06	0.09	0.05	0.08	0.06	0.05
5	-	-	-	-	0.01	**0.07**	0.05	0.06	0.05	0.07
6	-	-	-	-	-	**0.03**	**0.07**	**0.07**	0.08	0.10
7	-	-	-	-	-	-	0.01	0.06	0.06	0.06
8	-	-	-	-	-	-	-	0.01	0.07	0.08
9	-	-	-	-	-	-	-	-	0.01	0.07
10	-	-	-	-	-	-	-	-	-	0.01

D

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.38	0.28	0.33	0.50	0.27	0.48	0.21	0.18	0.30
2	-	0.03	0.27	0.24	0.54	0.42	0.51	0.33	0.34	0.26
3	-	-	**0.02**	0.34	0.61	0.42	0.59	0.33	0.23	**0.08**
4	-	-	-	0.01	0.39	0.32	0.35	0.22	0.32	0.34
5	-	-	-	-	0.02	0.43	0.10	0.37	0.49	0.62
6	-	-	-	-	-	0.02	0.38	0.21	0.37	0.42
7	-	-	-	-	-	-	0.01	0.34	0.49	0.60
8	-	-	-	-	-	-	-	0.03	0.23	0.35
9	-	-	-	-	-	-	-	-	0.02	0.26
10	-	-	-	-	-	-	-	-	-	**0.02**

Fig. 10. Distance matrices between the 3D texture classes. The α_1 operator is used in the mesh-LBP computation using the following descriptors (from top): mean curvature (H), gaussian curvature (K), shape index (SI) and angle between facets normal (D). The intra-class and inter-class distances that are less separated are highlighted in gray and yellow, respectively.

1) Mesh-LBP Variants: We conducted the same texture classification experiment with the mesh-CSLBP variant (Eq. (4)). In this variant, we kept the same spatial resolution

H

	1	2	3	4	5	6	7	8	9	10
1	0.10	0.58	0.47	0.48	0.45	0.58	0.49	0.46	0.51	0.47
2	-	0.13	0.56	0.56	0.55	0.63	0.59	0.56	0.61	0.54
3	-	-	**0.11**	**0.43**	**0.43**	0.59	0.46	0.46	0.45	**0.43**
4	-	-	-	0.10	0.44	0.61	0.50	0.48	0.48	0.45
5	-	-	-	-	**0.11**	0.56	0.46	0.44	0.48	**0.43**
6	-	-	-	-	-	0.13	0.57	0.55	0.61	0.57
7	-	-	-	-	-	-	0.11	0.47	0.48	0.46
8	-	-	-	-	-	-	-	0.10	0.48	0.44
9	-	-	-	-	-	-	-	-	0.11	0.46
10	-	-	-	-	-	-	-	-	-	**0.11**

K

	1	2	3	4	5	6	7	8	9	10
1	0.10	0.59	0.50	0.47	0.52	0.62	0.69	0.50	0.52	0.51
2	-	**0.16**	**0.55**	0.56	0.63	0.70	0.75	0.62	0.62	0.57
3	-	-	0.11	0.46	0.56	0.67	0.72	0.55	0.48	0.45
4	-	-	-	0.10	0.48	0.62	0.67	0.50	0.47	0.49
5	-	-	-	-	0.08	0.49	0.54	0.44	0.56	0.59
6	-	-	-	-	-	0.06	0.44	0.54	0.68	0.68
7	-	-	-	-	-	-	0.06	0.64	0.71	0.73
8	-	-	-	-	-	-	-	0.10	0.56	0.58
9	-	-	-	-	-	-	-	-	0.12	0.52
10	-	-	-	-	-	-	-	-	-	0.11

SI

	1	2	3	4	5	6	7	8	9	10
1	0.09	0.57	0.45	0.45	0.43	0.51	0.45	0.42	0.46	0.46
2	-	**0.16**	0.56	**0.54**	**0.54**	0.62	0.55	0.57	0.55	0.56
3	-	-	0.11	0.40	0.42	0.52	0.44	0.45	0.43	0.43
4	-	-	-	0.10	0.40	0.53	0.44	0.45	0.42	0.41
5	-	-	-	-	0.11	0.51	0.42	0.43	0.42	0.44
6	-	-	-	-	-	0.14	0.53	0.51	0.53	0.54
7	-	-	-	-	-	-	0.11	0.45	0.45	0.46
8	-	-	-	-	-	-	-	0.10	0.46	0.47
9	-	-	-	-	-	-	-	-	0.12	0.44
10	-	-	-	-	-	-	-	-	-	0.10

D

	1	2	3	4	5	6	7	8	9	10
1	0.10	0.68	0.56	0.63	0.67	0.69	0.77	0.69	0.58	0.56
2	-	0.11	0.60	0.58	0.69	0.75	0.76	0.72	0.66	0.58
3	-	-	0.10	0.53	0.71	0.75	0.79	0.73	0.48	**0.41**
4	-	-	-	0.09	0.55	0.77	0.71	0.73	0.55	0.53
5	-	-	-	-	0.09	0.76	0.64	0.73	0.68	0.71
6	-	-	-	-	-	0.05	0.49	0.35	0.79	0.74
7	-	-	-	-	-	-	0.04	0.40	0.78	0.79
8	-	-	-	-	-	-	-	0.04	0.74	0.72
9	-	-	-	-	-	-	-	-	0.11	0.52
10	-	-	-	-	-	-	-	-	-	**0.11**

Fig. 11. Distance matrices between the 3D texture classes. In this case, the α_2 operator function is used in the mesh-LBP computation.

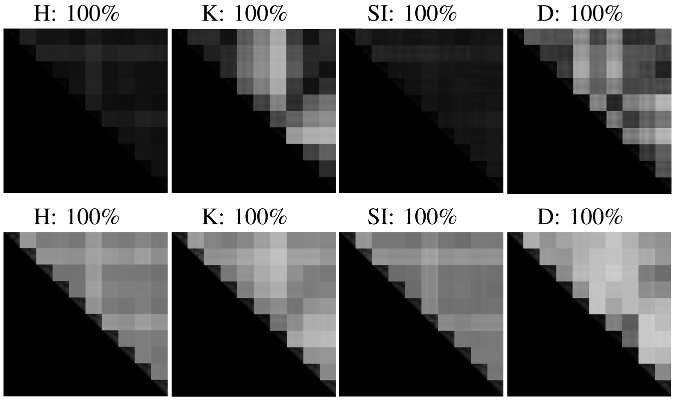


Fig. 12. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for the mesh-LBP obtained using H, K, SI, and D scalar surface descriptors (top row for α_1 and bottom row for α_2, respectively). The classification accuracy, estimated as the percentage of occurrences where the inter-class distance is greater than the intra-class distance across all the classes is also reported for each descriptor.

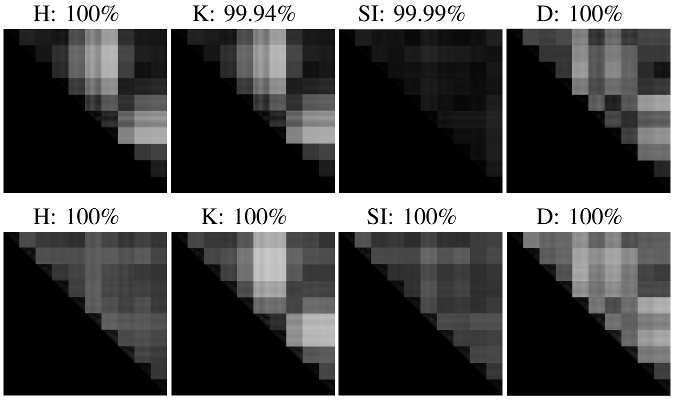


Fig. 13. Distance matrices obtained for the mesh-CSLBP using H, K, SI and D scalar surface descriptors (top row for α_1 and bottom row for α_2, respectively), and their related classification accuracies.

H

	1	2	3	4	5	6	7	8	9	10
1	0.02	0.13	0.07	0.07	0.07	0.16	0.13	0.09	0.13	0.09
2	-	0.02	0.10	0.11	0.15	0.15	0.12	0.11	0.13	0.09
3	-	-	0.01	0.08	0.10	0.11	0.08	0.04	0.08	0.05
4	-	-	-	0.01	0.08	0.16	0.13	0.09	0.14	0.10
5	-	-	-	-	0.01	0.17	0.15	0.12	0.16	0.12
6	-	-	-	-	-	0.02	0.09	0.10	0.09	0.11
7	-	-	-	-	-	-	**0.02**	0.07	0.06	**0.05**
8	-	-	-	-	-	-	-	0.01	0.08	0.05
9	-	-	-	-	-	-	-	-	0.01	0.07
10	-	-	-	-	-	-	-	-	-	0.01

K

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.13	0.11	0.09	0.30	0.49	0.61	0.19	0.10	0.09
2	-	0.01	0.09	0.19	0.40	0.59	0.71	0.27	0.12	0.11
3	-	-	0.01	0.16	0.38	0.57	0.69	0.25	0.07	0.09
4	-	-	-	0.01	0.23	0.43	0.56	0.12	0.14	0.16
5	-	-	-	-	0.02	0.21	0.34	0.19	0.35	0.37
6	-	-	-	-	-	**0.05**	**0.16**	0.38	0.54	0.56
7	-	-	-	-	-	-	0.03	0.51	0.66	0.67
8	-	-	-	-	-	-	-	0.01	0.22	0.26
9	-	-	-	-	-	-	-	-	0.01	0.11
10	-	-	-	-	-	-	-	-	-	0.01

SI

	1	2	3	4	5	6	7	8	9	10
1	0.01	0.11	0.08	0.07	0.04	0.10	0.06	0.04	0.06	0.13
2	-	0.01	0.09	0.06	0.11	0.14	0.11	0.11	0.08	0.07
3	-	-	0.01	0.06	0.07	0.09	0.06	0.07	0.04	0.09
4	-	-	-	0.01	0.06	0.11	0.08	0.07	0.05	0.07
5	-	-	-	-	**0.01**	0.09	0.06	**0.03**	0.05	0.11
6	-	-	-	-	-	0.02	0.07	0.09	0.09	0.14
7	-	-	-	-	-	-	0.02	0.06	0.06	0.12
8	-	-	-	-	-	-	-	**0.01**	0.05	0.12
9	-	-	-	-	-	-	-	-	0.01	0.09
10	-	-	-	-	-	-	-	-	-	0.01

D

	1	2	3	4	5	6	7	8	9	10
1	0.02	0.27	0.25	0.29	0.50	0.21	0.41	0.22	0.27	0.27
2	-	0.02	0.21	0.21	0.55	0.35	0.49	0.38	0.22	0.22
3	-	-	**0.02**	0.21	0.60	0.35	0.53	0.39	0.17	**0.08**
4	-	-	-	0.02	0.48	0.30	0.42	0.36	0.19	0.22
5	-	-	-	-	0.02	0.36	0.14	0.34	0.59	0.64
6	-	-	-	-	-	0.03	0.27	0.17	0.35	0.38
7	-	-	-	-	-	-	0.02	0.25	0.53	0.57
8	-	-	-	-	-	-	-	0.02	0.41	0.43
9	-	-	-	-	-	-	-	-	0.02	0.15
10	-	-	-	-	-	-	-	-	-	**0.02**

Fig. 14. Distance matrices between the 3D texture classes computed with the mesh-CSLBP, α_1 operator, and the four descriptors.

and the azimuthal quantization ($r = 7$, $m = 12$), but the mesh-LBP patterns are now coded on 6 digits, setting thus their ranges to [0,6] and [0,63] for α_1 and α_2, respectively. We also adopted the uniform/non-uniform partition as for mesh-LBP, though the resulting number of classes (i.e., the histogram bins) is not significantly reduced (62 instead of 64). Fig. 13 depicts the distance matrices between all the 30 classes' instances, computed with the four previously used scalar functions (H, K, SI and D), together with their corresponding accuracy rates. The 10 × 10 distance matrices related to α_1 and α_2 are also depicted in Fig. 14 and 15, respectively.

We notice that the accuracy rate is virtually 100% and exactly 100% for α_1 and α_2, respectively, across the four descriptors. Also, in the worst cases, the ratio between the interclass distances and the corresponding intra-class distance is scoring 2.4 for both α_1 (SI) and α_2 (SI). These scores confirm the discriminant capability of the mesh-CSLBP, though to a less extent than the mesh-LBP, for which the corresponding ratios are 2.33 and 3.37. However, this inferiority is expected because of the lower range of the mesh-CSLBP pattern.

H

	1	2	3	4	5	6	7	8	9	10
1	0.04	0.28	0.21	0.20	0.18	0.32	0.26	**0.20**	0.25	0.21
2	-	0.06	0.30	0.29	0.30	0.35	0.34	0.30	0.35	0.26
3	-	-	0.06	0.20	0.25	0.34	0.27	0.24	0.21	0.19
4	-	-	-	0.05	0.24	0.36	0.27	0.24	0.25	0.21
5	-	-	-	-	0.05	0.35	0.30	0.24	0.30	0.22
6	-	-	-	-	-	0.07	0.36	0.32	0.34	0.31
7	-	-	-	-	-	-	0.08	0.26	0.30	0.26
8	-	-	-	-	-	-	-	**0.08**	0.25	0.22
9	-	-	-	-	-	-	-	-	0.04	0.21
10	-	-	-	-	-	-	-	-	-	0.04

K

	1	2	3	4	5	6	7	8	9	10
1	0.04	0.27	0.22	0.20	0.34	0.66	0.66	0.27	0.26	0.20
2	-	0.05	0.27	0.30	0.46	0.75	0.76	0.38	0.31	0.29
3	-	-	0.05	0.24	0.44	0.75	0.75	0.34	0.21	0.21
4	-	-	-	0.06	0.30	0.63	0.63	0.24	0.26	0.25
5	-	-	-	-	0.05	0.40	0.42	0.26	0.42	0.40
6	-	-	-	-	-	**0.08**	**0.16**	0.55	0.70	0.71
7	-	-	-	-	-	-	0.05	0.55	0.71	0.72
8	-	-	-	-	-	-	-	0.08	0.33	0.35
9	-	-	-	-	-	-	-	-	0.03	0.27
10	-	-	-	-	-	-	-	-	-	0.04

SI

	1	2	3	4	5	6	7	8	9	10
1	0.04	0.32	0.19	0.21	0.17	0.25	**0.18**	0.17	0.23	0.25
2	-	0.05	0.28	0.27	0.27	0.39	0.31	0.35	0.23	0.28
3	-	-	0.05	0.17	0.18	0.27	0.19	0.23	0.17	0.20
4	-	-	-	0.05	0.18	0.31	0.22	0.25	0.17	0.19
5	-	-	-	-	0.05	0.27	0.19	0.21	0.18	0.22
6	-	-	-	-	-	0.08	0.27	0.26	0.30	0.30
7	-	-	-	-	-	-	**0.07**	0.23	0.22	0.24
8	-	-	-	-	-	-	-	0.06	0.26	0.27
9	-	-	-	-	-	-	-	-	0.04	0.21
10	-	-	-	-	-	-	-	-	-	0.04

D

	1	2	3	4	5	6	7	8	9	10
1	0.04	0.47	0.38	0.40	0.55	0.38	0.51	0.31	0.35	0.37
2	-	0.04	0.35	0.39	0.63	0.55	0.63	0.54	0.38	0.38
3	-	-	**0.07**	0.31	0.64	0.48	0.59	0.46	0.25	**0.22**
4	-	-	-	0.06	0.55	0.44	0.53	0.45	0.28	0.28
5	-	-	-	-	0.06	0.44	0.29	0.42	0.64	0.68
6	-	-	-	-	-	0.07	0.42	0.36	0.48	0.51
7	-	-	-	-	-	-	0.09	0.39	0.59	0.64
8	-	-	-	-	-	-	-	0.07	0.46	0.48
9	-	-	-	-	-	-	-	-	0.04	0.24
10	-	-	-	-	-	-	-	-	-	0.05

Fig. 15. Distance matrices between the 3D texture classes computed with the mesh-CSLBP, α_2 operator, and the four descriptors.

2) Mesh-LBP Discriminative Power: In order to compare quantitatively the different cases using a synthetic performance indicator, the discriminative power of the mesh-LBP descriptors and the mesh-CSLBP descriptors has been evaluated according to the following criterion:

$$\mathcal{J} = \sum_{i=1}^{M} \sum_{j=i+1}^{M} \mathcal{D}_{ij}, \tag{7}$$

where M is the number of texture classes. $\mathcal{D}_{ij}$ is the probabilistic-like inter-class separation between texture classes i and j defined as follows:

$$\mathcal{D}_{ij} = \frac{1}{2} dist(\bar{H}_i, \bar{H}_j)^2 \left(\frac{1}{\sigma^2_{H_i}} + \frac{1}{\sigma^2_{H_j}}\right) + \frac{1}{2}\left(\frac{\sigma^2_{H_i}}{\sigma^2_{H_j}} + \frac{\sigma^2_{H_j}}{\sigma^2_{H_i}} - 2\right),$$

where $(\bar{H}_i, \bar{H}_j)$ and $(\sigma_{H_i}, \sigma_{H_j})$ are the mean histograms and the variances of the texture classes i and j, respectively.

The criterion $\mathcal{J}$ computed for the different mesh-LBP and mesh-CSLBP descriptors is reported in Table I. For both variants, we notice that K and D score the best performance for the α_1 and α_2 operators, respectively. The same ranking is kept for the other descriptors H and SI.

TABLE I

Discriminative Power $\mathcal{J}$ Computed for the Different Mesh-LBP and Mesh-CSLBP Descriptors

		H	K	SI	D
mesh-LBP	$\mathcal{J}(\alpha_1)$	16.8	228.3	9.7	128.4
	$\mathcal{J}(\alpha_2)$	61.1	100.0	54.4	172.8
mesh-CSLBP	$\mathcal{J}(\alpha_1)$	20.4	233.0	15.0	187.1
	$\mathcal{J}(\alpha_2)$	41.4	111.2	34.5	105.0

D. Comparative Evaluation

In the following, we compared the mesh-LBP descriptors performance, in terms of 3D texture classification, with other standard 3D surface descriptors (Section IV-D.1) and the 2D-LBP applied to depth images (Section IV-D.2).

1) 3D Surface Descriptors: In this analysis, we considered the following 3D surface descriptors: the *Geometric Histograms* (GH) [39]; the *Shape Distribution* variants [40], namely, the distance between a fixed point and one random point on the surface ($D1$), the distance between two random points on the surface ($D2$), the square root of the area of the triangle between three random points on the surface ($D3$), the cube root of the volume of the tetrahedron between four random points on the surface ($D4$), and the angle between three random points on the surface ($A3$); the *Spin-Images* [41]; and the *mesh-HOG* [42]. Using these descriptors, we performed the same experiments discussed above for the mesh-LBP. The distance matrices between all the classes' instances are reported in Fig. 16. Comparing these distance matrices with those obtained for the mesh-LBP using different descriptors and reported in Fig. 12, it clearly emerges the performance improvement obtained using the mesh-LBP approach.

Similarly to the results presented for the mesh-LBP, we also provide the distance matrices obtained between the different texture classes. In this case, we report just the results for the best competing solutions as resulted from Fig. 16, that is, the descriptor $D4$, which resulted the best among the *Shape Distributions*, and the *Spin Images* that resulted the most effective among the other descriptors. In these matrices, depicted in Fig. 17, the cases in which the intra-class distance is greater than the corresponding inter-class distances are highlighted in gray and red, respectively. It can be observed that this case occurs for several pairs of texture classes for both the matrices, whereas this is never the case for the distance matrices obtained for the mesh-LBP descriptors, where the intra-class distances are lower than the corresponding inter-class distances across all the cases.

2) 2D-LBP on Depth Images: We conducted an additional experiment to assess the mesh-LBP performance with respect to the 2D-LBP counterpart applied on depth images [25]. For this purpose, we considered 30 depth image samples for each texture surface (see Fig. 18). In each set, the samples were constructed at different rotation angles, varying from 0 to $2\pi/3$, around the surface's principal orientation, to avoid self-occlusion effects. For each sample, we computed multi-resolution 2D-LBP patterns with nearly the same setting than their mesh-LBP counterparts. That is, a radial resolution varying from 1 to 7, and an azimuthal resolution of 8 across all the radii. In addition, we adopted the local descriptors H, K, SI, and C, rather than the depth value (e.g., z coordinate) used usually in the standard 3D-LBP.

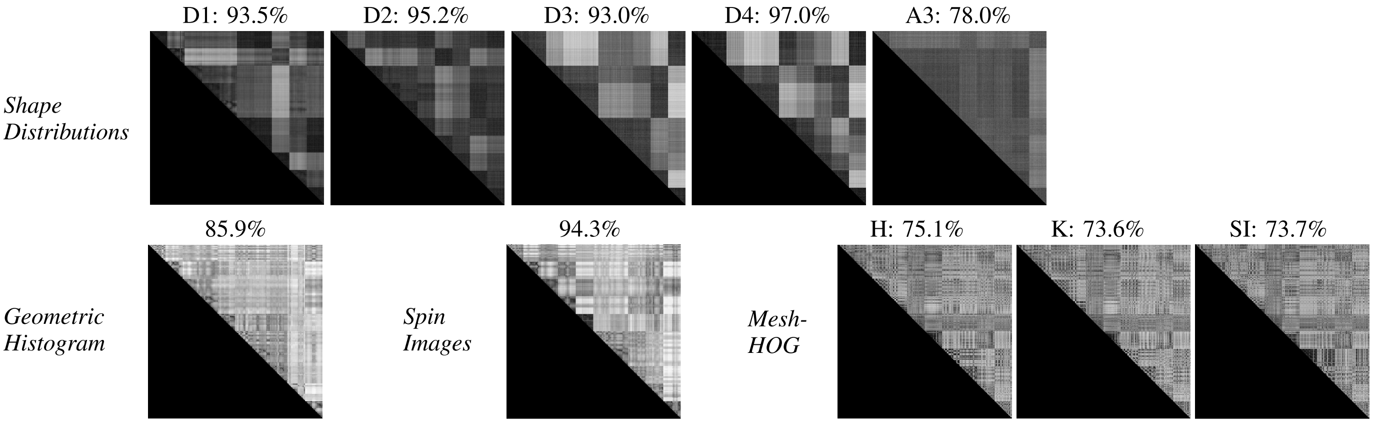


Fig. 16. Matrices reporting the distances between all the instances of the texture classes (30 instances per class). Distances are computed for: *Shape Distributions* (top); *Geometric Histogram* (bottom left); *Spin Images* (bottom middle); *mesh-HOG* (bottom right) computed for different surface scalar functions, namely, H, K and SI. For each descriptor, the overall classification accuracy is also reported in percentage.

D4

	1	2	3	4	5	6	7	8	9	10
1	**0.11**	**0.09**	0.56	0.73	0.68	0.17	0.31	0.45	0.66	**0.11**
2	-	**0.13**	0.55	0.71	0.66	0.20	0.31	0.43	0.64	**0.12**
3	-	-	**0.18**	0.24	**0.19**	0.46	0.32	**0.14**	**0.13**	0.61
4	-	-	-	**0.23**	**0.09**	0.65	0.53	0.36	**0.22**	0.76
5	-	-	-	-	**0.22**	0.60	0.48	0.30	**0.18**	0.71
6	-	-	-	-	-	0.13	0.17	0.34	0.57	0.25
7	-	-	-	-	-	-	0.15	**0.19**	0.42	0.38
8	-	-	-	-	-	-	-	**0.23**	0.25	0.50
9	-	-	-	-	-	-	-	-	**0.18**	0.70
10	-	-	-	-	-	-	-	-	-	**0.13**

Spin Images

	1	2	3	4	5	6	7	8	9	10
1	**0.64**	**0.54**	**0.37**	**0.60**	0.65	**0.59**	0.67	**0.60**	**0.44**	**0.41**
2	-	**0.38**	0.50	**0.33**	0.74	0.64	0.72	**0.55**	**0.33**	0.72
3	-	-	0.30	0.57	0.48	0.61	0.62	0.62	**0.33**	**0.37**
4	-	-	-	**0.35**	0.74	0.60	0.68	**0.48**	**0.34**	0.80
5	-	-	-	-	0.16	0.70	0.60	0.73	0.55	0.67
6	-	-	-	-	-	0.50	**0.35**	**0.27**	0.56	0.70
7	-	-	-	-	-	-	**0.42**	**0.46**	0.60	0.73
8	-	-	-	-	-	-	-	**0.55**	**0.52**	0.75
9	-	-	-	-	-	-	-	-	**0.52**	0.60
10	-	-	-	-	-	-	-	-	-	**0.39**

Fig. 17. Distance matrices for the *Shape Distribution D*4 and the *Spin Images*. The cases in which the intra-class distance is greater than the corresponding inter-class distances are highlighted (gray and red colors are used for, respectively, the intra- and inter-class distances).

We computed, the classification rate for the three 2D-LBP variants, namely, the uniform LBP ($u2$), the rotation invariant LBP (ri) and the uniform rotation-invariant LBP ($riu2$). Fig. 19 depicts the obtained classification rates and the distance matrices for each variant. First, we notice the low performance of the $u2$ variant, which naturally is expected because its sensitivity to surface rotation. The ri and the $riu2$ show much better performance, but they remain lower than their mesh-LBP counterpart across all the instances. We also computed the distance matrices, showing inter-class and the intra-class distances, between the different texture classes. We reported only the results related to the best competing variants, namely, K_{riu2} and SI_{ri}, which are depicted in Fig. 20.

E. Robustness to Mesh Irregularities

Ideally, a mesh is formed entirely by equal-sized triangles (not necessarily equilateral), and 6-valence vertices. As we mentioned previously, though nowadays triangle mesh surfaces acquired by shape digitizers have overall good quality in terms of uniformity, they often contain areas of non-uniform tessellation showing extremum triangles, such as needle or flat triangles, and whereby the assumption of vertex valence of six does not hold. These two aspects make the arithmetic progression of the number of facets across the rings, expressed in Eq. (2) no longer satisfied. We addressed this issue by interpolating or sub-sampling the scalar function on the mesh across the rings. In this experimentation, we wanted to assess to what extent this procedure can cope with mesh irregularities that can be encountered in real mesh data. To simulate the two aforementioned aspects that corrupt the mesh uniformity, we propose the following corruption procedure reported in Algorithm 4.

The random perturbation consists of applying the following transformation to one of the vertex of the facet:

$$t(v) = v + \sigma \vec{u}, \qquad (8)$$

where σ is a random positive variable taking values in the range $[0.2, 0.8]$, and $\vec{u}$ is a unit vector collinear with the line joining the vertex v to the middle point of its opposite edge. The combination of this transformation and the edge collapsing aims to obtain mesh irregularity instances close to the ones encountered in real mesh data. The extreme case of this corruption scheme is represented by meshes where 80% of the facets and 50% of the edges have undergone vertex perturbation and edge collapsing, respectively. Though real mesh data rarely exhibit such extreme corruption, at least after a basic pre-processing, considering such extreme cases, allows us to best assess the extent to which the adopted interpolation/subsampling procedure can address mesh irregularities. We applied this corrupting procedure to the textured shape surfaces included in the ten classes employed in the 3D texture matching experiments discussed above. For each texture class, we obtained 40 sets of mesh instances at increasing corruption amplitudes. In turn, each set contains the 30 instances of the class. Fig. 21 depicts an original mesh surface and four samples of corrupted instances at different levels.

For each mesh corruption level, we performed the full classification procedure involving all the 30 instances of each class. The obtained classification rates are depicted in Fig. 22. It can be observed that all the mesh-LBP descriptors keep a classification accuracy above 99% up to the 30th corruption level, and practically 100% up to the 20th level, especially for the $\alpha(k) = 2^k$ operator (Fig. 22(b)). For this category,

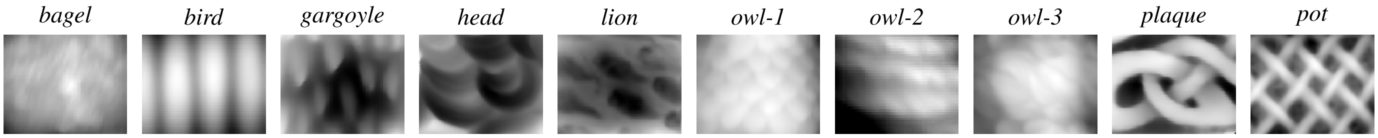


Fig. 18. Depth images of 10 3D texture classes.

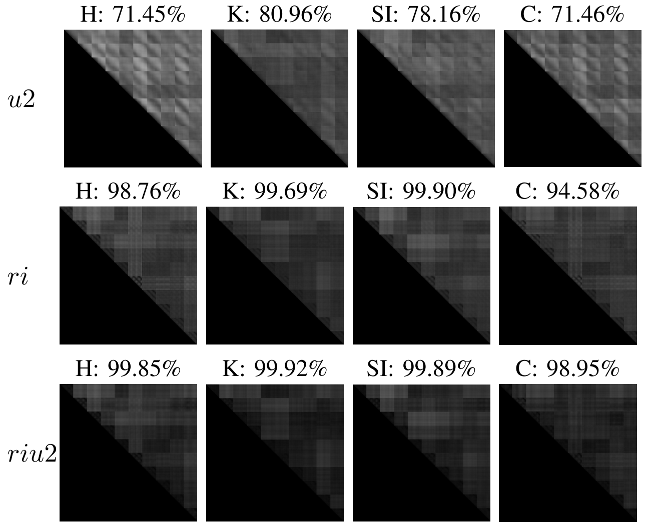


Fig. 19. Matrices reporting the distances between all the instances of the texture classes computed from depth images (30 depth images per class) using 2D-LBP patterns. Distances are computed for the uniform LBP ($u2$), rotation invariant (ri), and uniform rotation-invariant ($riu2$). The 2D-LBP patterns were computed for each of the scalar functions H, K, SI and C. The overall classification accuracy is also reported in percentage.

K_{riu2}

	1	2	3	4	5	6	7	8	9	10
1	0.06	0.23	0.18	0.21	0.08	0.09	0.14	0.16	0.22	0.18
2	-	0.07	0.11	0.14	0.19	0.21	0.15	0.14	0.10	0.13
3	-	-	0.06	0.07	0.13	0.14	0.08	0.07	0.07	0.07
4	-	-	-	0.05	0.15	0.15	0.10	0.09	0.10	0.09
5	-	-	-	-	**0.05**	**0.05**	0.08	0.10	0.16	0.13
6	-	-	-	-	-	0.06	0.10	0.11	0.19	0.14
7	-	-	-	-	-	-	0.05	0.06	0.12	0.07
8	-	-	-	-	-	-	-	0.05	0.12	0.08
9	-	-	-	-	-	-	-	-	0.05	0.12
10	-	-	-	-	-	-	-	-	-	0.05

SI_{ri}

	1	2	3	4	5	6	7	8	9	10
1	0.13	0.21	0.35	0.27	0.13	0.20	0.16	0.20	0.22	0.22
2	-	**0.16**	0.26	0.19	0.16	0.22	0.17	0.19	**0.14**	0.13
3	-	-	0.13	0.15	0.28	0.30	0.27	0.21	0.19	0.23
4	-	-	-	0.13	0.20	0.22	0.19	0.15	0.14	0.16
5	-	-	-	-	**0.13**	0.16	**0.11**	0.13	0.15	0.15
6	-	-	-	-	-	**0.16**	0.15	**0.16**	0.22	0.19
7	-	-	-	-	-	-	**0.13**	**0.13**	0.15	0.14
8	-	-	-	-	-	-	-	**0.14**	**0.14**	**0.14**
9	-	-	-	-	-	-	-	-	**0.13**	**0.12**
10	-	-	-	-	-	-	-	-	-	0.13

Fig. 20. Distance matrices for the 2D-LBP: K_{riu2} and SI_{ri}. Cases when the intra-class distance is greater than the corresponding inter-class distances are highlighted (gray and red colors are used for, respectively, the intra- and inter-class distances).

we notice in particular that with *gaussian* curvature, the descriptor keeps above 99% accuracy across all the corruption levels, seconded by the *Shape Index* (SI), which is showing similar performance up to the 37th level. In the first category (Fig. 22(a)), the *angle between facets normal* is virtually scoring 100% till the 29th level. Overall, the results indicate a clear resistance of the mesh-LBP descriptors to mesh irregularities, and bring evidence of the validity of the proposed interpolation/subsampling procedure.

Algorithm 4 Triangular Mesh Corruption Procedure

```
procedure MeshCorrupt( )
    for $m = 10 : 10 : 80$ do
        for $n = 10 : 10 : 50$ do
            Apply random perturbation to $m\%$ randomly
            selected facet's vertex
            Collapse $n\%$ randomly selected edges
        end for
    end for
end procedure
```

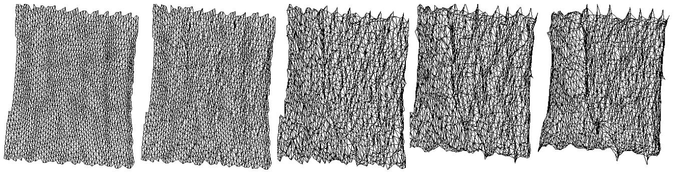


Fig. 21. The original mesh (left) and 4 corrupted instances at levels 1, 11, 21, and 31.

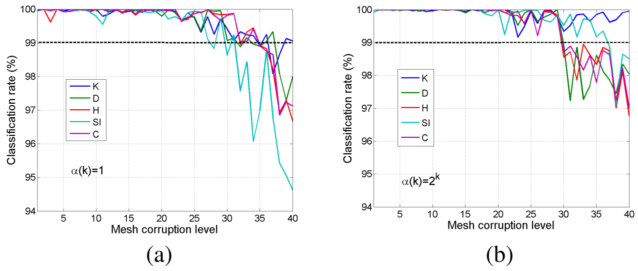


Fig. 22. Classification accuracy obtained for the different mesh corruption levels: (a) $\alpha(k) = 1$; (b) $\alpha(k) = 2^k$.

V. Discussion and Conclusions

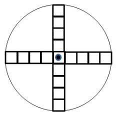
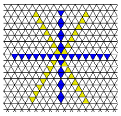
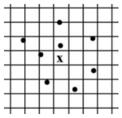
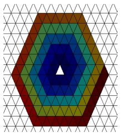
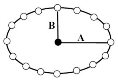
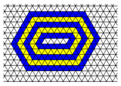
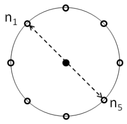
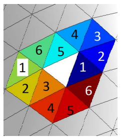
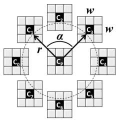
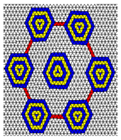
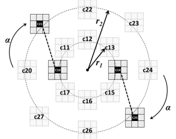
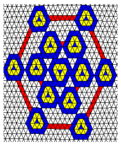
In this paper, we presented mesh-LBP as a novel framework for computing local binary patterns on triangular mesh manifolds. This framework keeps the simplicity and the elegance characterizing the original LBP and allows the extension of all its variants, developed in 2D image analysis, to the mesh manifold. The mesh-LBP reliefs object surface data from normalization and registration procedure required when using depth images, while it extends the spectrum of LBP analysis to closed surfaces.

The experimental tests revealed that mesh-LBP exhibits a "uniformity" aspect for the different types of scalar functions, pretty similar to the one noticed in 2D-LBP. We also provided a simple method for addressing rotation invariance that proved to be effective as was confirmed by repeatability and the other subsequent experiments.

Experiments on 3D texture classification showed clear evidence of the appropriateness of the mesh-LBP descriptors for such a task, and their superior discriminative power as compared to other popular 3D descriptors. Experiments related to the mesh-CSLBP variant showed that we can keep virtually the same performance, while reducing the computational cost. Regarding the choice of the scalar function, in summary, the angle between facets normal and the gaussian curvature seem

TABLE II

DIFFERENT VARIANTS OF THE LBP OPERATOR, GROUPED IN FOUR CATEGORIES, AND THE CORRESPONDING PATTERNS OBTAINED FOR THE 2D-LBP AND THE MESH-LBP ARE REPORTED. UNDER THE COLUMN "CLASS OF VARIATION", WE EVIDENCE THE MAIN ASPECT OF THE LBP COMPUTATION FRAMEWORK, WHICH IS VARIED BY A PARTICULAR SOLUTION

Class of variation	LBP variant	2D-LBP pattern	mesh-LBP pattern
Browsing path	(a) Local-line [43]		
	(b) Archimedian Spiral [44]		
Contour / initial seed	(c) Elongated [18]		
Comparison scheme	(d) Center-symmetric [29]		
Structural element	(e) Three-Patch [10]		
	(f) Four-Patch [10]	 c22 c23 c11 c12 c13 c20 c24 c17 c16 c15 c27 c26 α α	

the more effective surface descriptors to be used within the mesh-LBP framework as emerges throughout the different experiments. The same experiments that were carried out with depth image modalities, confirmed also the superiority of our mesh-LBP, noting also the constraints on the depth image construction procedure that we had to consider to obtain the desired quality in terms of pattern visibility. It is also noticeable, in particular, that the rotation invariant mesh-LBP α_1 outperforms its 2D-LBP 'ri' variant despite the lower size of its associate histogram (13 against 36 for 'ri').

The re-sampling scheme of the scalar function over each ORF ring proved to be an effective mechanism for addressing mesh irregularities. In the related experiment, the gaussian curvature and the shape index exhibited the best robustness score.

The comparison of the α_1 and α_2 operators does not provide conclusive results, apart that they perform best with gaussian curvature and the angle between facets normal, respectively. However, the compactness of the descriptor obtained with the α_1 operator, and the resulting lower computational complexity required to compare descriptors, vote for this solution especially in the cases where time constraints are relevant.

As future work, we plan extending the mesh-LBP to global analysis. One potential approach is extracting ordered blocks from the mesh surfaces and then construct from them, by concatenation, a global histogram. We believe that mesh-LBP will open-up new perspectives for mesh manifold analysis and will be an appropriate complement to other mesh manifold analysis techniques.

APPENDIX

In this appendix, we show that most, if not all, the different LBP neighborhood and operator variants proposed in the literature [24] can be easily derived from the ordered rings structure of the mesh-LBP. In fact, one important feature of the mesh-LBP is that the topology of the neighborhood from which the descriptor is computed can be changed to accommodate the specificities of a given shape analysis application. Some of the most effective and used LBP variants, their structure and the related mesh-LBP patterns are summarized in Table II, where the LBP variants are organized in four categories, according to which aspect of the basic LBP computation framework is varied. In the following, we provide more details about the definition and computation of the mesh-LBP variants:

Browsing Path: Considering a set of directions D_j, a mesh-LBP operator can be defined which uniformly samples m facets along the directions D_j:

$$meshLBP_m^{D_j}(f_c) = \sum_{k=0}^{m-1} s(h(f_k^j) - h(f_c)) \cdot \alpha(k). \qquad (9)$$

This directional extension of the mesh-LBP can be regarded as a generalization to the mesh case of the Local Line Binary Pattern (LLBP) [43], introduced in the context of face recognition to encode anisotropic information of neighbouring pixels by computing LBP across vertical and horizontal directions (Table II, case (a)). A variety of operators can be further derived from Eq. (9) by combining the different directional operators. Among these, the ORF framework can be used to arrange the facets according to a spiral-wise topology, thus allowing the derivation of the equivalent of the Archimedean spiral-like LBP, as originally defined in [44] (Table II, case (b)).

Contour/Initial Seed: Several LBP variants can be obtained by using a non-circular neighbourhood of the central facet, through a particular setting of the initial contour. For example, selecting the set of Fin_root facets of Algorithm 3 in a bar-like shape fashion produces elongated ORFs. This pattern can be viewed as the mesh-LBP version of the elongated local binary pattern (ELBP) proposed in [18] (Table II, case (c)).

Comparison Strategy: This category includes the variants aiming to reduce the dimensionality of the LBP descriptor. The uniform patterns and the central symmetric mesh-LBP (mesh-CSLBP in Table II, case (d)) variants have been presented in Section III-A and experimented in Section IV. More recently, a dimensionality reduction method for LBP, denoted as *orthogonal combination* of LBP (OC-LBP) has been proposed in [45]. In this case, the basic idea is to first split the neighboring pixels of the original LBP operator into several non-overlapped orthogonal groups, then compute the LBP code separately for each group, and finally concatenate them together. The same computation procedure can be used in the mesh-LBP framework, resulting in an equivalent mesh-OCLBP operator.

Structural Element: The *three-patch* and *four-patch* LBP (TPLBP and FPLBP, respectively) have been proposed by Wolf et al. [10] as an extension of the *center-symmetric* LBP (CSLBP) [29] for the purpose of extracting complementary information to pixel-based descriptors. In the mesh-LBP framework, we define a mesh-TPLBP like structure by constructing ORF patches of w rings at the central facet, and at m equally spaced facets on the r-th ring around the central facet. The case (e) of Table II depicts a mesh-TPLBP composed of ORF with 3-ring patches ($w = 4$), one at the central facet and six at equally spaced positions on the 12-th ring (e.g., $r = 12$). Varying the parameters w, r and m other mesh-TPLBP can be obtained as well. Formally, we express the mesh-TPLBP operator as follows:

$$meshTPLBP_{r,m,w}(f_c) = \sum_{k=0}^{m-1} s(Y) \cdot \alpha(k)$$
$$with\ Y = d(P_k, P_{f_c}) - d(P_{k+\delta\ mod\ m}, P_{f_c}),$$

where $d(.)$ is any distance function between two patches constructed on w rings (for example, $d(.)$ can be the L_2 norm or the Bhattacharyya distance between the geometric histogram [39] associated to the two patches); and δ controls the arc-length distance between the patches of a pair.

The FPLBP construction follows a similar approach to the three-patch solution, but considering four patches on two concentric rings (see Table II, case (f)). The construction of the mesh-FPLBP version of this operator follows virtually the same steps of the mesh-TPLBP, except that two groups, rather than one, of equally spaced ORF with w-rings are generated at two different radii (e.g., the inner ring with radius r_1, and the outer ring with radius r_2). The mesh-FPLBP operator is defined as follows:

$$meshFPLBP_{r_1,r_2,m,w}(f_c) = \sum_{k=0}^{m/2-1} s(Y) \cdot \alpha(k)$$
$$Y = d(P^1_k, P^2_{k+\delta\ mod\ m}) - d(P^1_{k+m/2}, P^2_{k+m/2+\delta\ mod\ m}).$$

Different variants of the mesh-FPLBP can be constructed by tuning the parameters r_1, r_2, m, w and δ. An example is shown in Table II, case (f), using $r_1 = 5$, $r_2 = 10$, $m = 6$, $w = 2$ and $\delta = 0$.

Acknowledgments

Preliminary ideas and experiments of the mesh-LBP framework appeared in [46].

References

[1] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[2] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[3] T. Mäenpää, J. Viertola, and M. Pietikäinen, "Optimising colour and texture features for real-time visual inspection," *Pattern Anal. Appl.*, vol. 6, no. 3, pp. 169–175, Mar. 2003.

[4] L. Cao, J. Luo, F. Liang, and T. S. Huang, "Heterogeneous feature machines for visual recognition," in *Proc. 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 1095–1102.

[5] D. Guo, V. Atluri, and N. Adam, "Texture-based remote-sensing image segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1472–1475.

[6] C. Song, F. Yang, and P. Li, "Rotation invariant texture measured by local binary pattern for remote sensing image classification," in *Proc. IEEE 2nd Int. Workshop Edu. Technol. Comput. Sci.*, vol. 3. Wuhan, China, Mar. 2010, pp. 3–6.

[7] A. Lucieer, A. Stein, and P. Fisher, "Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty," *Int. J. Remote Sens.*, vol. 26, no. 14, pp. 2917–2936, 2005.

[8] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Proc. 8th Eur. Conf. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 469–481.

[9] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[10] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Proc. ECCV Workshop Faces Real-Life Images*, Marseille, France, Oct. 2008, pp. 1–14.

[11] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Proc. 5th Int. Workshop Adv. Biometric Pers. Authentication*, Beijing, China, Oct. 2005, pp. 179–186.

[12] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.

[13] L. Cai, C. Ge, Y.-M. Zhao, and X. Yang, "Fast tracking of object contour based on color and texture," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1421–1438, Nov. 2009.

[14] X. Wang and M. Mirmehdi, "Archive film restoration based on spatiotemporal random walks," in *Proc. 11th Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 478–491.

[15] G. Sandbach, S. Zafeiriou, and M. Pantic, "Local normal binary patterns for 3D facial action unit detection," in *Proc. 19th IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep./Oct. 2012, pp. 1813–1816.

[16] H. Li, L. Chen, D. Huang, Y. Wang, and J. Morvan, "3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns," in *Proc. 1st Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 2577–2580.

[17] G. Sandbach, S. Zafeiriou, and M. Pantic, "Binary pattern analysis for 3D facial action unit detection," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Guildford, U.K., Sep. 2012, pp. 119.1–119.12.

[18] S. Liao and A. C. S. Chung, "Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude," in *Proc. 8th Asian Conf. Comput. Vis.*, Tokyo, Japan, Nov. 2007, pp. 672–679.

[19] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li, "Face detection based on multi-block LBP representation," in *Proc. Int. Conf. Biometrics*, Washington, DC, USA, Sep. 2007, pp. 11–18.

[20] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under Bayesian framework," in *Proc. 1st Int. Conf. Image Graph.*, Hong Kong, Dec. 2004, pp. 306–309.

[21] D. Huang, Y. Wang, and Y. Wang, "A robust method for near infrared face recognition based on extended local binary pattern," in *Proc. 3rd Int. Symp. Vis. Comput.*, Lake Tahoe, CA, USA, Nov. 2007, pp. 437–446.

[22] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Proc. 3rd Int. Workshop Anal. Modelling Faces Gestures*, Rio de Janeiro, Brazil, Oct. 2007, pp. 168–182.

[23] T. Ahonen and M. Pietikäinen, "Soft histograms for local binary patterns," in *Proc. Finnish Signal Process. Symp.*, Oulu, Finland, Aug. 2007, pp. 1–4.

[24] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision using Local Binary Patterns*. Berlin, Germany: Springer-Verlag, 2011.

[25] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3D+2D based face recognition at both feature and decision levels," in *Proc. 2nd Int. Workshop Anal. Modeling Faces Gestures*, Beijing, China, Oct. 2005, pp. 44–54.

[26] Y. Huang, Y. Wang, and T. Tan, "Combining statistics of geometrical and correlative features for 3D face recognition," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., Sep. 2006, pp. 879–888.

[27] D. Huang, M. Ardabilian, Y. Wang, and L. Chen, "3D face recognition using eLBP-based facial description and local feature hybrid matching," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1551–1565, Oct. 2012.

[28] J. Fehr and H. Burkhardt, "3D rotation invariant local binary patterns," in *Proc. Int. Conf. Pattern Recognit.*, Tampa, FL, USA, Dec. 2008, pp. 1–4.

[29] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, Mar. 2009.

[30] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognit.*, vol. 33, no. 1, pp. 43–52, Jan. 2000.

[31] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. 11th Eur. Conf. Comput. Vis.*, vol. 3. Crete, Greece, 2010, pp. 347–360.

[32] T. Darom and Y. Keller, "Scale-invariant features for 3D mesh models," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2758–2769, May 2012.

[33] (2008). *MIT CSAIL Database*. [Online]. Available: http://people.csail.mit.edu/tmertens/textransfer/data/

[34] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. IEEE 7th Int. Conf. Autom. Face Gesture Recognit.*, Southampton, U.K., Apr. 2006, pp. 211–216.

[35] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical Geometry of Non-Rigid Shapes*. New York, NY, USA: Springer-Verlag, 2008.

[36] A. S. Mian, M. Bennamoun, and R. Owens, "Keypoint detection and local feature matching for textured 3D face recognition," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 1–12, Aug. 2008.

[37] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Proc. Shape Modeling Int.*, Genoa, Italy, Jun. 2004, pp. 1–12.

[38] A. Cerri *et al.*, "SHREC'13 track: Retrieval on textured 3D models," in *Proc. 6th Eurograph. Workshop 3D Object Retr.*, Girona, Spain, May 2013, pp. 73–80.

[39] A. P. Ashbrook, R. B. Fisher, C. Robertson, and N. Werghi, "Finding surface correspondence for object recognition and registration using pairwise geometric histograms," in *Proc. 5th Eur. Conf. Comput. Vis.*, Freiburg im Breisgau, Germany, Jun. 1998, pp. 674–686.

[40] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. Graph.*, vol. 21, no. 4, pp. 807–832, Oct. 2002.

[41] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

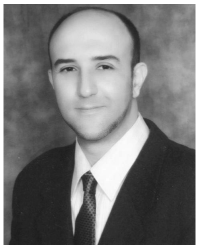
[42] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 373–380.

[43] A. Petpon and S. Srisuk, "Face recognition with local line binary pattern," in *Proc. 5th Int. Conf. Image Graph.*, Xi'an, China, Sep. 2009, pp. 533–539.

[44] L. Nanni, A. Lumini, and A. Brahnam, "Local binary patterns variants as texture descriptors for medical image analysis," *Artif. Intell. Med.*, vol. 49, no. 2, pp. 117–125, Jun. 2010.

[45] C. Zhu, C.-E. Bichot, and L. Chen, "Image region description using orthogonal combination of local binary patterns enhanced with color information," *Pattern Recognit.*, vol. 46, no. 7, pp. 1949–1963, Jul. 2013.

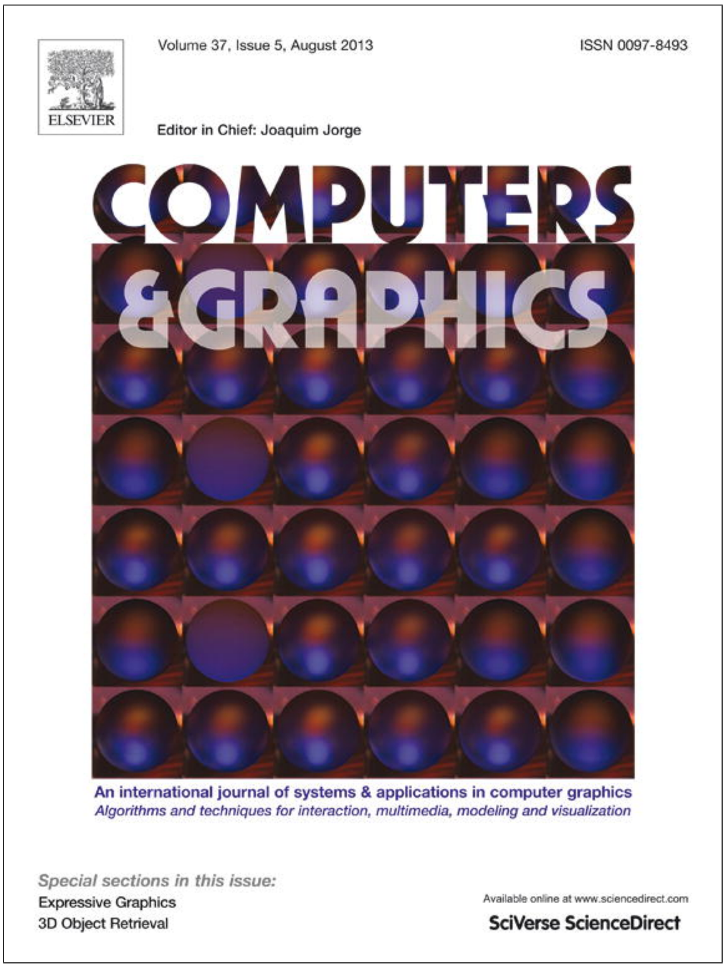
[46] N. Werghi, S. Berretti, A. Del Bimbo, and P. Pala, "The mesh-LBP: Computing local binary patterns on discrete manifolds," in *Proc. IEEE Int. Workshop 3D Represent. Recognit.*, Sydney, Australia, Dec. 2013, pp. 562–569.



Naoufel Werghi (SM'14) received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France. He has been a Research Fellow with the Division of Informatics, University of Edinburgh, Edinburgh, U.K., and a Lecturer with the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His main research area is 2D/3D image analysis and interpretation, where he has led several funded projects in the areas of biometrics, medical imaging, geometrical reverse engineering, and intelligent systems. He has been a Visiting Professor with the Department of Electrical and Computer Engineering, University of Louisville, Louisville, U.K., in 2002, and the Media Integration and Communication Center, University of Florence, Florence, Italy, in 2012. He has authored over 70 journal and conference papers.

Stefano Berretti received the Ph.D. degree in information and telecommunications engineering from the University of Florence, Florence, Italy, in 2001, where he is currently an Associate Professor with the Department of Information Engineering and the Media Integration and Communication Center. His current research interests focus on 3D object retrieval and partitioning, face recognition and facial expression recognition from 3D and 4D data, 3D face superresolution, and human action recognition from 3D data. He has been a Visiting Researcher with IIT Bombay, Mumbai, India, and a Visiting Professor with the Institute Telecom, TELECOM Lille 1, Lille, France, and the Khalifa University, Abu Dhabi, United Arab Emirates. He has authored over 100 papers appeared in conference proceedings and international journals in the area of pattern recognition, computer vision, and multimedia. He is on the Program Committee of several international conferences and serves as a frequent reviewer of many international journals. He has been the Co-Chair of the Fifth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment in 2012, held in conjunction with the European Conference on Computer Vision in 2012.

Alberto del Bimbo is currently a Full Professor of Computer Engineering, the Director of the Master in Multimedia, and the Director of the Media Integration and Communication Center with the University of Florence, Florence, Italy, where he was the Deputy Rector for Research and Innovation Transfer from 2000 to 2006. His scientific interests are multimedia information retrieval, pattern recognition, image and video analysis, and natural human–computer interaction. He has authored over 350 publications in some of the most distinguished scientific journals and international conferences, and a monograph entitled *Visual Information Retrieval*. From 1996 to 2000, he was the President of the IAPR Italian Chapter and a Member-at-Large of the IEEE Publication Board from 1998 to 2000. He was the General Chair of the IAPR International Conference on Image Analysis and Processing in 1997 and the IEEE International Conference on Multimedia Computing and Systems in 1999, and the Program Co-Chair of the ACM Multimedia in 2008. He is the General Co-Chair of the ACM Multimedia in 2010 and the European Conference on Computer Vision in 2012. He is a fellow of the International Association for Pattern Recognition and an Associate Editor of *Multimedia Tools and Applications*, *Pattern Analysis and Applications*, the *Journal of Visual Languages and Computing*, and the *International Journal of Image and Video Processing*, and was an Associate Editor of *Pattern Recognition*, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.



Volume 37, Issue 5, August 2013 ISSN 0097-8493

ELSEVIER

Editor in Chief: Joaquim Jorge

COMPUTERS &GRAPHICS

An international journal of systems & applications in computer graphics
Algorithms and techniques for interaction, multimedia, modeling and visualization

Special sections in this issue:
Expressive Graphics
3D Object Retrieval

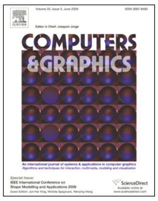
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Computers & Graphics 37 (2013) 509–525

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag



Special Section on 3D Object Retrieval

Matching 3D face scans using interest points and local histogram descriptors ☆

Stefano Berretti [a,*], Naoufel Werghi [b], Alberto del Bimbo [a], Pietro Pala [a]

[a] *Department of Information Engineering, University of Firenze, Italy*
[b] *Khalifa University of Science Technology & Research, Abu Dhabi, United Arab Emirates*

ARTICLE INFO

Article history:
Received 30 October 2012
Received in revised form
29 March 2013
Accepted 1 April 2013
Available online 1 May 2013

Keywords:
3D face recognition
3D keypoints
3D local descriptors

ABSTRACT

In this work, we propose and experiment an original solution to 3D face recognition that supports face matching also in the case of probe scans with missing parts. In the proposed approach, distinguishing traits of the face are captured by first extracting 3D keypoints of the scan and then measuring how the face surface changes in the keypoints neighborhood using local shape descriptors. In particular: 3D keypoints detection relies on the adaptation to the case of 3D faces of the meshDOG algorithm that has been demonstrated to be effective for 3D keypoints extraction from generic objects; as 3D local descriptors we used the HOG descriptor and also proposed two alternative solutions that develop, respectively, on the *histogram of orientations* and the *geometric histogram* descriptors. Face similarity is evaluated by comparing local shape descriptors across inlier pairs of matching keypoints between probe and gallery scans. The face recognition accuracy of the approach has been first experimented on the difficult probes included in the new 2D/3D Florence face dataset that has been recently collected and released at the University of Firenze, and on the Binghamton University 3D facial expression dataset. Then, a comprehensive comparative evaluation has been performed on the Bosphorus, Gavab and UND/FRGC v2.0 databases, where competitive results with respect to existing solutions for 3D face biometrics have been obtained.

1. Introduction

The humans' cognitive system has a peculiar attitude in recognizing faces with high accuracy, at least for familiar people in favorable viewing conditions (i.e., good illumination, small occlusions, etc.). Automatic identity recognition performed by machines has entered the scene some decades ago with the aim to extend the human capabilities by covering different and more general contexts. In particular, face has affirmed itself as one of the most important biometric trait due to the fact that images or videos of the face are collectable in an easy and non-intrusive way, whereas other biometrics, such as fingerprints or iris scans are impractical to implement in many scenarios (e.g., in a surveillance setting). Impressively, recent studies report that automatic face recognition can even outperform the human performance in some particular conditions [1]. However, the accuracy of automatic identity recognition based on faces still suffers from many factors, such as pose changes, illumination variations, facial expressions and occlusions.

To solve these problems, face recognition using 3D scans of the face has been recently proposed as an alternative or complementary solution to conventional 2D face recognition approaches using still images or videos, so as to allow accurate face recognition also in real-world applications with unconstrained acquisition. Confirming this recent research trend, several 3D face recognition approaches have been proposed and experimented in the last few years (see the survey in [2], and the literature review in [3–5] for a thorough discussion). However, many of the works appeared in this field, proposed conventional face recognition experiments, where both the probe and gallery scans are assumed to be acquired cooperatively in a controlled environment in which the whole face is precisely captured and represented. These methods mainly focussed on face recognition in the presence of expression variations, reporting very high accuracy on benchmark databases like the *Face Recognition Grand Challenge* (FRGC version 2.0) [6]. Recent studies also exploit ethnicity, gender and age to improve the accuracy of 3D face recognition [7,8]. Solutions enabling face recognition in uncooperative scenarios are now attracting an increasing interest. In such a case, probe scans are acquired in unconstrained conditions that may lead to *missing parts* (non-frontal pose of the face) or to *occlusions* due to hair, glasses, scarves, hand gestures, etc. These difficulties are further sharpened by the recent advent of 4D scanners (3D plus time) [9–11], capable of acquiring temporal sequences of 3D scans. In fact, the dynamics of facial movements captured by these devices can be useful for many applications [12,13], but also increases the acquisition noise and the variability in subjects' pose. In summary, techniques supporting *3D partial face matching* are gaining

☆To comment on this article, please join the discussion on the Collage Authoring Environment Google Group https://groups.google.com/group/collage-authoring-environment.

* Corresponding author. Tel.: +39 55 4796415; fax: +39 55 4796493.
E-mail addresses: berretti@dsi.unifi.it, stefano.berretti@unifi.it (S. Berretti).

0097-8493/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2013.04.001

importance in making 3D face recognition techniques deployable in more general contexts and, in perspective, in scenarios where 3D dynamic acquisition is performed. However, the research in this context is still preliminary also due to the limited number of face databases that also comprise partial acquisitions of 3D faces [14–16].

1.1. Related work

Below, we review the most recent methods for 3D face recognition, by limiting our analysis to the works that also propose and evaluate solutions supporting partial match of 3D facial scans. In particular, we focus on methods that were also evaluated on scans acquired from non-frontal views of the face for which the recognition problem is further complicated by artifacts that alter the geometry of the acquired 3D surface in correspondence to the borders of the missing regions, rather than to solutions that just cropped 3D full face scans to simulate missing parts. In general, existing solutions can be grouped as *global* and *local*; *Multimodal* approaches that combine together 2D and 3D methods are also possible.

Global 3D face representations for partial face matching have been proposed in a limited number of works. The first solutions appeared in this category used the Iterative Closest Point (ICP) algorithm [17]. The method proposed in [18], was global and multimodal trying to combine 3D shape and 2D texture to perform surface and appearance-based matching. The surface matching component was based on a coarse to fine alignment between a 2.5D probe and a fully 3D face model (obtained by the fusion of five 2.5D scans). In the coarse step, first three manually labeled generic points were used to calculate the rigid transformation that aligns the 2.5D scan with the 3D model, then specific feature points are identified by finding correspondence between shape index values of two scans. These feature points are then used to define a grid of control points around them. In the fine alignment step, a modified ICP algorithm is applied on the grid of control points to refine the alignment between 2.5D probes and 3D models. Good results were reported for neutral, expressive and partial scans of a proprietary database of 200 individuals, though the computational cost does not scale to large datasets. Following a similar idea, 3D face matching between 2.5D probe scans and fully 3D models is proposed in [19]. Also in this case, a coarse alignment is first performed based on the manual labeling of three generic points in the two matching scans, then ICP fine alignment is performed and the registration error is used to evaluate the similarity between the two matching scans. Separate results for scans acquired with moderate expressions, illumination changes and left/right pose variations were reported on a database of 50 subjects. The main limitations of the approach are in the scalability of ICP, and the manual labeling required by the initial coarse alignment. A canonical representation of the face is proposed in [20], where the isometry invariance of the face surface is exploited to manage missing data obtained by randomly removing areas from frontal face scans. However, no side scans were used for recognition. In [21], results on partial face matching removing quadrants of the FRGC v2.0 probes and using face crops around the nose tip are reported. This approach relies on the symmetry of the 3D face scans in order to identify the nose tip and register depth maps so as to derive a Pure Shape Difference Map (PSDM) between pairs of matching scans. Unfortunately, the symmetry hypothesis used for the registration and fiducial points detection is often violated when side views of the face are acquired in uncooperative scenarios. Instead, the experiments are conducted by just removing parts of the face after the preprocessing has been performed on the entire scans. The fact that the same part of the face is removed from both probe and gallery scans in order to generate the PSDM also reduces the concrete applicability of the approach.

The approaches above provide a *global* modeling of both gallery and probe scans, but more successful and scalable solutions use *local* representations of the face. A possible way to solve locally the problem of missing data in 3D face acquisition is to detect the absence of regions of the face and use the existing data to reconstruct the missing parts. The reconstructed scan can then be used as an input to conventional 3D face recognition methods that assume that the entire scan is available. This approach is followed in [22], focusing on face occlusions induced by glasses, scarves, caps, or by the subject's hand. A generic facial model and thresholding on facial surface distances are used to detect occlusions. In this way, the occluded areas are detected and the missing regions are restored using information from the non-occluded parts. However, face recognition accuracy was not evaluated. In [23,24], an inter-pose face recognition solution is proposed which exploits the hypothesis of facial symmetry to recover missing data in facial scans with large pose variations. First, an automatic face landmarks detector is used to identify the pose of the facial scan by marking regions of missing data and roughly registering the facial scan with an Annotated Face Model (AFM) [25]. Then, the AFM is fitted using a deformable model framework that exploits facial symmetry where data are missing. Wavelet coefficients extracted from a geometry image derived from the fitted AFM are used for the match. Experiments have been performed using the *University of Notre Dame* (UND) database [15], with the FRGC v2.0 gallery scans and side scans with 45° and 60° rotation angles respectively as probes. Since it is based on the left/right facial symmetry, this solution can work as long as half of the face with respect to the yaw axis is visible in the scan.

Tackling the problem from an opposite perspective, some methods divide the face into regions and try to restrict the match to uncorrupted parts of the face. Following this idea, the approach in [26] accurately identifies the nose tip in order to extract multiple overlapping regions around it. These regions are matched using the ICP algorithm and the respective scores are combined together in order to evaluate face similarity. This method is extended in [27] by using a set of 38 regions that densely cover the face, and selecting the best-performing subset of 28 regions to perform matching using the ICP algorithm. A recognition experiment accounting for partial match is reported that uses the left and right parts of the FRGC v2.0 probes. However, the experiments only account for the case in which some of the extracted regions are missing, rather than considering the more general case where also parts of the regions can miss. A part-based 3D face recognition method is proposed in [28], which operates in the presence of both expression variations and occlusions. The approach is based on the use of Average Region Models (ARMs) for registration: The facial area is manually divided into several meaningful components, such as eye, mouth, cheek and chin regions, and registration of faces is carried out by separate dense alignment of the regions with respect to the corresponding ARMs. The dissimilarities between gallery and probe scans obtained for individual regions are then combined to determine the final dissimilarity score. Under variations, like those caused by occlusions, the method can determine noisy regions and discard them. The performance of this approach is tested on the *Bosphorus*3D face database [16] that includes facial expressions, pose differences and occlusions. However, a strong limitation of this solution is the use of manually annotated landmarks that are required for both face alignment and regions segmentation. Instead of using extended regions, in [29] the face is represented by a collection of radial curves originating from the nose tip. Face comparison is obtained by elastic matching of the curves. A quality control allows the exclusion of corrupted radial curves from the match, thus enabling the recognition also in the case of missing data. Results of partial matching are given for the side scans of the *Gavab* database [14].

Methods that perform face recognition based on regions, use some landmarks of the face to identify the regions of interest for the match. However, facial landmarks are difficult to detect when the pose significantly deviates from the frontal one. In addition, since parts of the regions can be missing or occluded, the extraction of effective region descriptors is hindered, so that regions comparison is mostly performed using rigid (ICP) or elastic registration (*deformable models*). Approaches that use keypoints of the face promise to solve some of these limitations. Rather than relying on the detection of specific regions of the face that can fail in the presence of occlusions and missing parts, they assume that detection of keypoints on the face surface and description of these keypoints yield robust yet accurate representation of facial traits, also in the presence of occlusions and missing parts. In doing so, the number of keypoints is supposed to be sufficiently high. In this perspective, the use of keypoints instead of facial landmarks is advantageous. In fact, just few facial landmarks can be accurately detected in an automatic way – from three to ten are at most reported [30] – and detection of a larger number of landmarks is difficult even through partial manual assistance. In the case of partial face scans, up to half of these points are typically not detectable, so that description of such points and of their relationships is of limited effectiveness for face recognition. Differently, a much larger number of keypoints are typically detected – from tens to hundreds of keypoints can be easily derived – and their distribution is rather sparse, not being constrained to specific locations of the face. This makes keypoints more robust than landmarks to missing parts and also allows the extraction of a large number of local descriptors of the face. A first approach that exploits keypoints of the face has been reported in [31], where a 3D keypoints detector and descriptor inspired by the Scale Invariant Feature Transform (SIFT) [32] have been designed. This detector/descriptor has been used to perform 3D face recognition through a multi-modal 2D+3D approach that also uses the SIFT detector/descriptor to index 2D texture face images. However, results do not account for scans with pose variations and missing parts. The 3D keypoints detector defined in [31] was further generalized to the match of generic objects in [33]. Use of keypoints for partial face matching has been recently reported in [34,35]. In this approach, Multi-Scale Local Binary Patterns (MS-LBP) and Shape Index (SI) are applied to face depth images, and the scalar values obtained at each pixel are used to create an MS-LBP map and an SI map. On both these maps, the SIFT detector and descriptor are used to represent local variations of the features extracted from the face. Finally, the matching scheme accounts for local and global face features by combining local matches between SIFT features, with global constraints originated by facial components. Partial face matching results are presented for the FRGC v2.0 scans where parts of the face are masked to simulate missing parts. However, as pointed out by the authors, the approach can deal automatically just with nearly frontal face data as those included in the FRGC v2.0 dataset. In the case of missing parts of the face due to large pose variations the approach is likely to fail. Methods in [36,37] use keypoints detection for the purpose of partial face matching, resulting the best performing approaches in the track on *3D Face Models Retrieval* of the SHREC'11 competition [38]. In particular, in [36] an extension of SIFT and index map based SIFT matching [34] is proposed. First, feature points are detected on each 3D face scan using *mesh* SIFT [39]; then, the quasi-daisy local shape descriptor [40] of each feature point is obtained using multiple order histograms of differential quantities extracted from the surface; Finally, these local descriptors are matched by computing their orientation angles. The number of matched points is used as similarity between two face scans. In [37], first a PCA based shape model is learned by registering a set of training scans to a reference template model (using 12 manually annotated landmarks) and subsequently warping the template on the training scans using a non-rigid registration based on variational implicit functions. The learned model is then fitted to probe and gallery scans to generate model-based descriptions used to evaluate scans similarity. In this approach, *mesh* SIFT is used to detect keypoints whose correspondences in different scans permit to initialize the pose of probe and gallery scans with respect to the model (anyway, a manual initialization is required for about 2.5% of the scans). After pose initialization, the model is fitted following a Bayesian strategy with outliers detection and estimation. The result is an EM alike optimization, where the model updates are alternated with outlier updates, iteratively.

1.2. Contribution and organization

In this work, we propose an original 3D face recognition approach which is also capable to perform recognition in the case parts of the face scans are missing. We rely on the observation that describing the face with local geometric information extracted at the neighbors of keypoints allows partial face matching in which no particular assumption about the number or locations of the keypoints is necessary to perform sparse keypoints matching. In so doing, the size of the support used to compute the local descriptor at keypoint locations becomes crucial: small supports reduce the effectiveness of the descriptor and large supports are more sensible to missing parts that can alter the support itself. In addition, discriminant facial features are not only related to local characteristics of the face surface in the proximity of a set of keypoints, but also to mutual relationships among the position of the keypoints on the face.

Based on these premises, we propose a 3D face description approach that relies on the detection of 3D keypoints on the face surface and the description of the surface in correspondence to these keypoints. In contrast to solutions where keypoints correspond to meaningful face landmarks, such as the *eyebrows*, *eyes*, *nose*, *cheek* and *mouth* [30], we do not exploit any particular assumption about the position of the keypoints on the face surface. Rather, we expect the position of keypoints to be influenced by the specific morphological traits of the face of each subject. In particular, we exploit the assumption of *within subject keypoints repeatability*: the position of the most stable keypoints – detected at the coarsest scales – do not change substantially across facial scans of the same subject. According to this, we propose an adaptation of the meshDOG [41,42] algorithm to the specific case of 3D faces as 3D keypoints detector. In fact, meshDOG has been introduced as 3D extrema detector for the case of generic 3D objects, proving its effectiveness. However, to the best of our knowledge, it has never been applied before to the case of 3D face matching. Then distinguishing traits of a face scan are captured by local descriptors at the detected keypoints. In particular, we experiment the meshHOG descriptor [41], and also propose and experiment two different local descriptors, namely the *histogram of orientations* (SHOT) and the *geometric histogram* (GH), which exploit local properties of the mesh in different ways. We point out that all the processing required to detect keypoints and extract their local descriptors is performed on 3D meshes without requiring any pose normalization or landmark detection. In the comparison of two faces, local descriptors at the 3D keypoints are matched in order to determine the keypoints correspondences. Spatial constraints using RANSAC [43] are also imposed to avoid outlier matches.

Our approach has been experimentally evaluated with a two-fold objective. On the one hand, we verified the accuracy of recognition on two datasets that include probes with extreme variations in terms of facial expressions (*The Binghamton University 3D facial expression dataset* (BU-3DFE) [44]), and probes with up to half of the face missing due to acquisitions with large pose variations (the *2D/3D Florence Face database* (UF-3D) [45]). On the other, we experimented our solution on three largely used benchmark datasets (namely *Bosphorus*, *Gavab* and *UND/FRGC v2.0*) which allow the comparison of our solution with respect to state of the art approaches.

The contribution of our approach and its novelty over existing solutions using a similar framework, including keypoints extraction, local description and keypoints matching [31,36,39,46], can be summarized as follows:

- *Method*—An original adaptation of the meshDOG detector to the case of face meshes; The adaptation and comparison of three mesh descriptors to the case of 3D faces and their use as local representation at the keypoints; Proposal of the *multi-ring GH* as the local descriptor at the keypoints, and its identification as the most suitable descriptor to be combined with meshDOG keypoints, providing accurate recognition both in the presence of expression variations and large missing parts of the face; A 3D keypoints matching that also encompasses outliers removal using RANSAC.
- *Experiments*—This work contributes an original experimental validation on the new UF-3D face dataset that has never been used before for the purpose of 3D face recognition. Results reported by our work on this dataset can be regarded as a reference evaluation for future works aiming to test 3D face recognition approaches on challenging scans with missing parts; A thorough experimentation on the large and extreme facial expressions included in the BU-3DFE; A comprehensive comparative evaluation that includes the *Bosphorus*, *Gavab* and *UND/FRGC v2.0* datasets.

The remaining content of the paper is organized as follows: In Section 2, we present the adaptation of the meshDOG detector to the case of 3D faces, and we motivate and discuss the relevance of detected keypoints; Local descriptors computed at the keypoints are reported in Section 3; The way local keypoint descriptors are matched in two scans under comparison, so as to permit identity recognition is detailed in Section 4; A thorough experimental validation and comparison are reported in Section 5; Finally, results are discussed and future research directions are outlined in Section 6.

2. 3D keypoints

Several keypoint detectors capable to identify salient points on 3D meshes have been recently proposed. For a thorough comparative evaluation the reader can refer to the recent report at the SHREC'11 contest [47] (track on “robust feature detection and description benchmark”) and to the performance evaluation reported in [48]. Among these methods, the meshDOG detector [41,42] has been proved to be superior, in terms of both repeatability of the detection and accuracy of the matching, to other 3D keypoint detectors/descriptors, like the Harris 3D [49], meshSIFT [39,46] and Shape MSER [50] (see the results in [47] for a comparative analysis, and also the comparison provided in [48]). In particular, the meshDOG detector is proposed to perform feature detection, while the meshHOG descriptor is used for the purpose of mesh matching between generic 3D meshes. However, in the work of Zaharescu et al. [41], the 3D keypoints (extrema) were used for matching generic objects, like 3D reconstruction of the human body, reconstructed and synthetic 3D objects, using photometric surface information to extract the object descriptors using meshHOG. To the best of our knowledge, the meshDOG detector has never been used before for the purpose of 3D face analysis. In the following, we present the adaptation of the method so as to make it appropriate for extracting keypoints of 3D face meshes.

2.1. meshDOG of face meshes

The keypoints detection starts by defining and computing a scalar function f on a 3D mesh S. In principle, the function f can be any scalar function $f(v): S \rightarrow R$ that for any vertex $v \in S$ returns a scalar value. This can comprise functions computed according to the chromatic appearance of the mesh surface as well as functions that consider properties of the surface like the mean or Gaussian curvatures. In our case, we used the *mean* curvature at vertex v as value of the function $f(v)$. Though such function is not completely intrinsic, and therefore not completely invariant to local isometric deformations, in practice the keypoints detected using mean curvature turned out to be more stable on 3D face data than keypoints obtained using Gaussian curvature. One motivation for this can be the average operation, which has the advantage to smooth the noise effect that can be present in the computation of principal curvatures. The choice of the mean curvature is also supported in the recent survey on the evaluation of 3D keypoint detectors by Salti et al. [48], where the mean curvature is reported to provide better results than Gaussian curvature when combined with the meshDOG detector. The same conclusion was also reported by the authors of the meshSIFT approach [39,46], where the mean curvature was used in the construction of their scale-space extrema. According to [51], the mean curvature is computed by first rotating the local neighborhood of a vertex so that the normal of the current vertex is aligned with the Z-axis, and the neighborhood can be described by XY only, instead of XYZ. Then, a least-squares quadratic patch is fitted to the local neighborhood of a vertex $h(x,y) = ax^2 + by^2 + cxy + dx + ey + g$, and the eigenvectors and eigenvalues of the Hessian matrix are used to calculate the principal and mean curvature of the vertex.

Once the function f (mean curvature) is computed for every vertex of the mesh, the keypoints selection proceeds by processing the values of the function f through three subsequent steps. In the first step, the extrema of the Laplacian's function (DOG) across scales are found using a one-ring neighborhood of each vertex. Then, the extrema are sorted and thresholded based on a percentage value of the overall number of extrema. Finally, in the third step, only the extrema with some degree of cornerness are retained, thus removing unstable extrema. Details of these steps are given in the following.

Extrema of the scale-space. As first step, a scale-space representation of the scalar function f defined on the mesh is constructed. At every scale, the function f is convolved with a Gaussian kernel (see Eq. (A.3) for the definition of the convolution on the mesh)

$$g_\sigma(x) = \frac{\exp(-x^2/2\sigma^2)}{\sigma\sqrt{2\pi}}, \tag{1}$$

where σ is the standard deviation of the Gaussian (set equal to $\sigma = 2^{1/3} e_{avg}$ in our experiments, being e_{avg} the average edge length); and, at a vertex v_i, x is the distance between neighboring vertices to the vertex v_i, that is $\|\mathbf{v}_j - \mathbf{v}_i\|$.

The scale-space of f is built incrementally on N+1 levels, so that $f_0 = f$, $f_1 = f_0 * g_\sigma$, $f_2 = f_1 * g_\sigma, \ldots, f_N = f_{N-1} * g_\sigma$. The *N Difference of Gaussian* (DOG) are then obtained by subtracting adjacent scales, e.g., $DOG_1 = f_1 - f_0$, $DOG_2 = f_2 - f_1, \ldots, DOG_N = f_N - f_{N-1}$. In so doing, it is relevant to note that in building the scale space, the geometry of the face does not change, but the different scalar functions f_k and DOG_k defined on the mesh. A total of 96 convolutions (i.e., scales) have been used in our work. Once the scale-space is computed, the feature points are selected as the maxima of the *DOG* across scales. In particular, a vertex is an extremum at a given scale k if its DOG_k

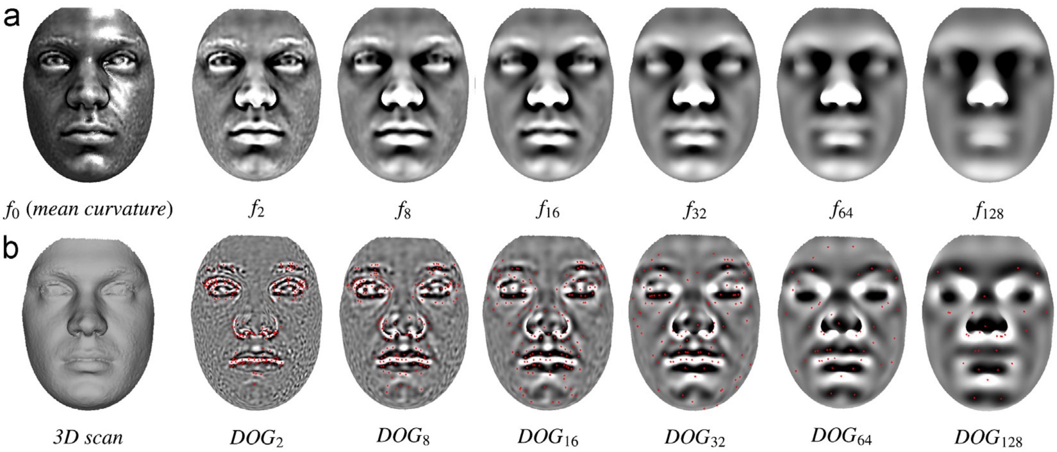


Fig. 1. (a) Face scans are colored according to the values of function f_k at different scales (f_0 being the mean curvature). (b) The 3D frontal acquisition (*subject001* of the UF-3D database) is reported, with the DOG_k values at different scales, and the 3D keypoints detected at that scale (in red). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

value is the maximum with respect to the DOG_k values in the 1-ring neighborhood at the same scale.

Percentage threshold. The extrema of the scale space obtained at the previous step are then sorted according to their magnitude. Only the top 1% of the sorted vertices are retained as extrema in our setting.

Cornerness. The last step, aims to remove unstable extrema, by retaining the features that exhibit corner characteristics. Following [32], this can be done by computing the Hessian at each vertex v of the mesh

$$H(v) = \begin{bmatrix} d_{xx}(v) & d_{xy}(v) \\ d_{yx}(v) & d_{yy}(v) \end{bmatrix}, \tag{2}$$

where d_{xx}, d_{xy}, d_{yx} and d_{yy} are the second partial derivative computed along the x and y directions. In particular, partial derivatives are estimated by applying the definition of directional derivatives given in Eq. (A.1) twice, e.g., $d_{xy} = \nabla_S D_{\overrightarrow{x}} f(v) \cdot \overrightarrow{y}$, where the gradient is computed using Eq. (A.2). In this context, the directions $\overrightarrow{x}$ and $\overrightarrow{y}$ represent a local coordinate system in the tangent plane of the vertex v, typically the gradient direction for $\overrightarrow{x}$ and its orthogonal direction for $\overrightarrow{y}$. The ratio between the largest λ_{max} and the lowest λ_{min} eigenvalues of the Hessian matrix is a good indication of a corner response, which is independent of the local coordinate frame. We typically use $\lambda_{max}/\lambda_{min} = 4$ as a minimum value to threshold responses.

An example of the scale-space construction is reported in Fig. 1. In (a), a sample face scan is colored according to the values of function f_k at different scales (f_0 being the mean curvature). In (b), gray levels are used to represent the DOG values at different scales (i.e., scales 2, 8, 16, 32, 64 and 128 are reported). The Experiment 1 Code Item 2 can Experiment 1 Data Item 1, in order to detect the 3D keypoints and generate the DOG_k images.

2.2. Keypoints distribution

According to an agreed classification [48], meshDOG is an *adaptive-scale* detector, in contrast to *fixed-scale* detectors which find distinctive keypoints at a specific constant scale, given as a parameter to the algorithm. The derivation of multiple DOG scales, allows the identification of more stable keypoints, which are typically located at highest scales, whereas keypoints detected in the first DOG scales are likely to be unstable and more affected by noise. As an example, the keypoints detected at some DOG scales for a sample face scan are highlighted in red in Fig. 1(b). At the first level of the scale-space (see DOG2 in Fig. 1(b)), the keypoints are mainly localized in the mouth and eyes regions (these regions are quite unstable with expressions) and around the nose and the eyebrows (more stable regions under expression changes). As the scale increases, keypoints are extracted by progressively smoothing the mean curvature function, and they tend to be more distributed on the face (see for example DOG64 and DOG128 in Fig. 1(b)). At these latter scales, some keypoints are located in the *forehead*, *cheekbone* and *chin*, with some keypoints close to the *pronasal* and *nasion* (thus, these keypoints are located in regions of the face that are much less affected by expression variations). Some keypoints can be also detected at multiple different scales; in such case, the keypoint occurring at the highest scale is retained. In Fig. 2, two further examples of keypoints detected at different scales are reported.

In general, meshDOG keypoints are located around areas characterized by high local curvature, this being true throughout the different scales. So, their semantic is related to the local curvature properties of the mesh. Our idea is that the robustness of the proposed approach comes from the combination of the presence of many keypoints detected at different scales, with the descriptiveness of local surface features (as discussed in Section 3). The fact that the keypoints are many increases the possibility to have a consistent number of matches also in the case of partial scans. The fact that the keypoints are extracted at different scales increases the probability to have keypoints detected in regions of the face that are not affected by facial expressions so that their descriptors are likely to be not altered in different scans of a same subject. Differently, keypoints detected in noisy regions or regions which are largely affected by expression changes are likely to not match due to their different descriptors. So, our idea is that though individual descriptors are not expression invariant, the overall matching schema can cope with expression variations thanks to the presence of keypoints that are located in regions of the face that are less affected by facial expressions. For the same reason, the approach can cope with missing parts and also occlusions, provided that a sufficient number of matches can be determined between probes and gallery scans. These considerations, motivated us to use the keypoints detected in the last levels of the scale-space. In particular, we considered for the purpose of local descriptor computation only the keypoints that are detected in the last 64 DOG scales (out of the 96 total scales used in the experiments), thus discarding those keypoints that have been detected only in the first 32 scales.

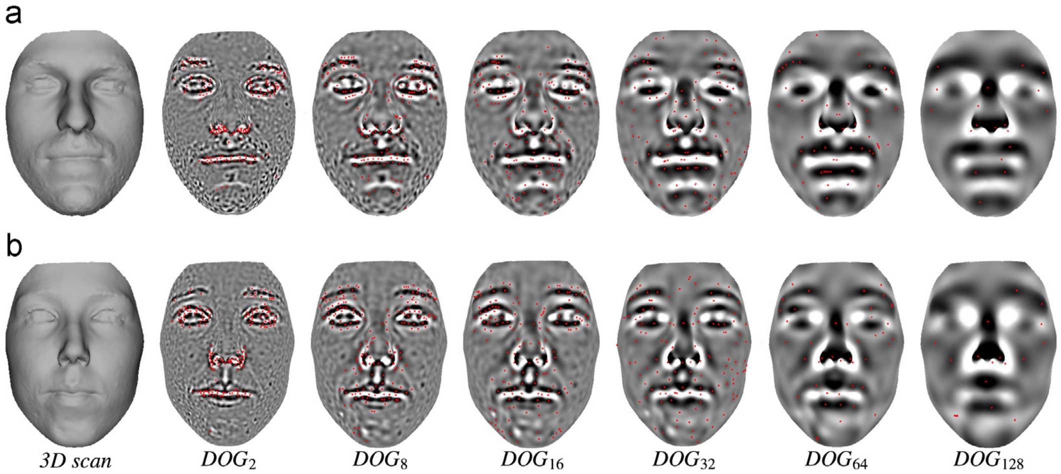


Fig. 2. DOG_k values at different scales and the 3D keypoints detected at that scale for a male subject in (a) and a female subject in (b). (a) subject002 and (b) subject003.

3. Local face descriptors

In order to support face matching, we assume that distinguishing traits of the face can be captured by describing the local morphology of the face in regions centered at 3D keypoints. This approach falls into the category of *signature descriptors* that represent the 3D surface using the neighborhood (called the *support*) of a given keypoint. A common problem faced by these solutions is the need for an invariant local reference frame in order to encode one or more geometric measurements computed individually for each point (vertex) of the support. Typically, the support is a spherical region whose radius determines the level of locality of the descriptor. Small values of the radius yield very local descriptors that capture the shape of the face in small regions around keypoints. By progressively increasing the value of the radius, the descriptor becomes more discriminant, although the probability of including regions of the face affected by undesired artifacts – such as missing parts or deformations caused by facial expressions – increases as well.

Based on these considerations, in the following we propose three different signatures to locally describe the 3D face at the keypoints, namely the *Histogram of Gradients* (HOG) (Section 3.1), the *Histogram of Orientations* (SHOT) (Section 3.2), and the *Geometric Histogram* (GH) (Section 3.3).

3.1. Histogram of gradients

The histogram of gradients descriptor [41] for a vertex extremum v is computed using a support region constituted by the vertices that belong to the neighborhood ring of size r. For each vertex from the neighborhood $v_i \in N_r(v)$, the gradient information $\nabla_S f(v_i)$ is computed using Eq. (A.2). As a first step, a local coordinate system is chosen, in order to make the descriptor invariant to rotation. Then, a histogram of gradient is computed, both spatially, at a coarse level, in order to maintain a certain high-level spatial ordering, and using orientations, at a finer level. Since the gradient vectors are three-dimensional, the histograms are computed in 3D. Since for this descriptor we followed the work of Zaharescu et al. [41], the reader is referred to that work for further implementation details.

3.2. Histogram of orientations

A description of the local shape of the 3D face is accomplished by developing on the idea of the 3D shape context descriptor proposed in [52] and on the work of [53]. The derivation of this signature first requires the definition of a local reference frame capable to make the extracted signature independent from translation and rotation of the mesh.

Local reference frame. In order to guarantee translation and rotation invariance of 3D face description and matching, each local descriptor is computed with respect to a local reference frame determined based on the local morphology of the face. For this purpose, the method proposed in [54] is considered. This avoids the descriptor computation over multiple rotations on different azimuth directions by determining a repeatable normal axis and an unique pair of directions lying on the tangent plane.

Given a keypoint located at vertex v, and a spherical neighborhood of radius R centered on v, a weighted covariance matrix $\mathbf{C}$ of the vertices within the neighborhood is computed as

$$\mathbf{C} = \frac{1}{K} \sum_{i:d_i \le R} (R - d_i)(\mathbf{v}_i - \mathbf{v})(\mathbf{v}_i - \mathbf{v})^T, \tag{3}$$

where $d_i = \|\mathbf{v}_i - \mathbf{v}\|$, and K is a normalization factor computed as

$$K = \sum_{i:d_i \le R} (R - d_i). \tag{4}$$

With respect to the usual computation of the covariance matrix, in Eq. (3) a smaller weight is assigned to distant vertices, and the centroid computation is replaced by the keypoint vertex v. A *total least squares* estimation of the normal direction is obtained by eigenvalue decomposition of the covariance matrix $\mathbf{C}$ of the vertex coordinates within the support. The eigenvectors of $\mathbf{C}$ define repeatable orthogonal directions in the presence of noise and clutter. Eigenvectors of Eq. (3) need to be disambiguated to yield a repeatable local reference frame. The idea is to orient each eigenvector so that its sign is coherent with the majority of the vectors it represents. If the three eigenvectors, given in decreasing eigenvalue order, are indicated as $\mathbf{x}^+$, $\mathbf{y}^+$, and $\mathbf{z}^+$ (and their opposite vectors with $\mathbf{x}^-$, $\mathbf{y}^-$, and $\mathbf{z}^-$), the disambiguated $\mathbf{x}$-axis is defined as

$$\begin{aligned} S_{x^+} &= \{i : d_i \le R \quad \text{and} \quad (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{x}^+ \ge 0\} \\ S_{x^-} &= \{i : d_i \le R \quad \text{and} \quad (\mathbf{p}_i - \mathbf{p}) \cdot \mathbf{x}^- > 0\} \\ \mathbf{x} &= \begin{cases} \mathbf{x}^+, & |S_x^+| \ge |S_x^-| \\ \mathbf{x}^-, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

The same procedure is used to disambiguate the $\mathbf{z}$-axis, whereas the $\mathbf{y}$-axis is obtained as the vector product $\mathbf{z} \times \mathbf{x}$.

Local signature. Once the local reference frame is identified, a spherical support around each keypoint v is considered and the vertices of the mesh included in this spherical region contribute to the computation of the local descriptor. The radial extent of this

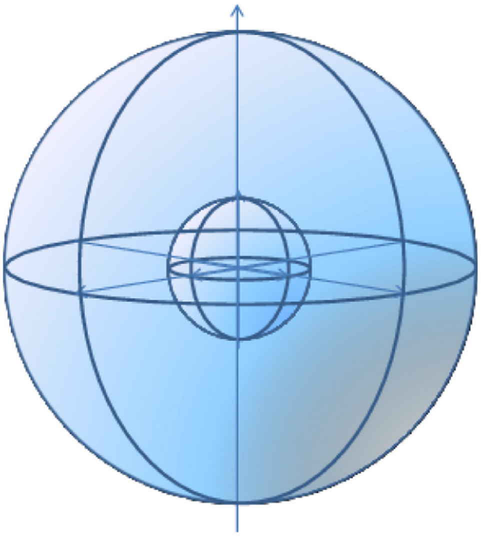


Fig. 3. Spherical local support around a keypoint. The volumetric partition of the sphere along the radial, azimuthal and elevation dimensions is reported.

sphere can be chosen independently from the radius R used for the computation of the local reference frame, but in our solution we considered the spherical support as having the same radius R used for the computation of the reference frame (i.e., 15 mm in our setting). This spherical volume is then divided along three dimensions: *radial*, *azimuthal* and *altitude*.

Along the radial dimension, the sphere is divided into concentric shells. To avoid the quadratic growth of the shell volumes with the shell index, a logarithmic parametrization of the shell radii is used

$$r_i = \frac{1}{s}\log_a\left(a^s\frac{i}{s}\right), \qquad (6)$$

where r_i is the radius of the shell of index i, s is the number of shells, and a is a parametrization coefficient that controls the growth of the shell radius (e.g., for $a=1$ the growth is linear, whereas with $a=2$ the volume of the shell is kept constant at different radius). The shells are then divided in the azimuthal plane using sectors of constant angular width, and along the elevation. In the experiments reported in Section 5, we used $a=2$, with three shells, four azimuthal sectors and two divisions along the elevation angle, resulting in a coarse partition of the volume around the keypoint into 24 spatial regions. Fig. 3 shows the idea of the volumetric partitioning of the spherical space around a keypoint (for the clarity of the plot just two shells are reported).

Once the local support is partitioned into volumetric regions (based on the unique 3D local reference frame), the histogram of the normals of the mesh vertices within the support is used as local descriptor (called SHOT in [53]). This histogram based representation provides the filtering effect required to achieve robustness to noise, and enhances the discriminative power of the descriptor by introducing geometric information about the location of the vertices within the support. As final step, all the local histograms are grouped together to form the signature which describes the mesh at the keypoint.

For each of the local histograms, mesh vertices contribute to bins according to a function of the angle θ_i, formed by the normal at each vertex within a volume of the support partition, $\mathbf{n}_{v_i}$, and the normal at the keypoint, $\mathbf{n}_u$. The $\cos\theta_i$ function is used, in that it can be computed efficiently using the dot product (i.e., $\cos\theta_i = \mathbf{n}_u \cdot \mathbf{n}_{v_i}$), and equally spaced binning on $\cos\theta_i$ is equivalent to a spatially varying binning on θ_i. This latter property results in a coarser binning for directions close to the reference normal direction and a finer one for orthogonal directions. In this way, small differences in orthogonal directions to the normal that are the most informative ones, cause a vertex to be accumulated in different bins and thus leading to different histograms. Instead, in the presence of quasi-planar regions this choice limits histogram differences due to noise by concentrating the contributions of the vertices in a fewer number of bins. In our experiments, we used 10 bins for each local histogram that combined with the partition into 24 volumetric regions, that results in a 240-dimensional signature for the keypoint.

To avoid boundary effects in the local histograms due to small differences of the spatial subdivision of the support, or to perturbations of the local reference frame, each vertex contributes to four histogram bins according to a quadrilinear interpolation between neighbors bins. In particular, the neighbor bins are represented by the neighboring bin in the local histogram and the bins having the same index in the local histograms of the neighboring volumes of the spatial partition. In doing so, each vertex contributes to neighbors bins by the weight $1-d$, where for the local histogram, d is the distance of the current entry from the central value of the bin; for elevation and azimuth dimensions, d is the angular distance of the entry from the central value of the closer volume along the dimension; for the radial dimension, d is the Euclidean distance of the entry from the central value of the closer volume along the radial dimension. Along each dimension, d is normalized by the distance between two neighbor bins or volumes. Finally, to achieve robustness to variations of the vertex density, all the local histograms are concatenated into a whole descriptor (signature) which is further normalized to sum up to 1, so as to retain the local differences as a source of discriminative information.

The local signature at a generic keypoint is expressed through a normalized histogram $G=(g_1,\ldots,g_N)$ where the size N of the signature depends on the size of the local histograms and on the number of volumes of the partition (i.e., the quantization along the radial, azimuthal and elevation dimension) of the local reference frame ($N=240$ in our case). Given two signatures $G=(g_1,\ldots,g_N)$ and $H=(h_1,\ldots,h_N)$ extracted at two keypoints, their dissimilarity is measured through the *Chi-square* distance χ^2, given by

$$\chi^2(G,H) = \frac{1}{2}\sum_{n=1}^{N}\frac{[g_n-h_n]^2}{g_n+h_n}. \qquad (7)$$

The Experiment 2 Code Item 2 can be executed on the Experiment 2 Data Item 1, in order to generate the SHOT signature of a 3D face scan.

3.3. Multi-ring geometric histogram

The geometric histogram (GH) is a local geometric descriptor proposed by Ashbrook et al. [55] and employed in surface alignment and matching. Basically, it is a 2D accumulator, or frequency table that counts the frequencies of two geometrical measurements, namely the angle and the distance between pairs of facets in a given neighborhood of a keypoint. In the following, we propose and describe a variation of the GH, which resulted more suited to our framework. This variant, develops on the idea of constructing the GH descriptor at a given keypoint in an incremental way, by accounting for an ordered sequence of rings defined around the keypoint. This idea is illustrated through the two steps involved in the computation: Derivation of the *ordered ring facets* in the neighborhood of the keypoint; Construction of the *discrete distributions* in each ring. In doing so, it is relevant to note that the GH descriptor is robust to translations and rotations also avoiding the computation of a reference frame.

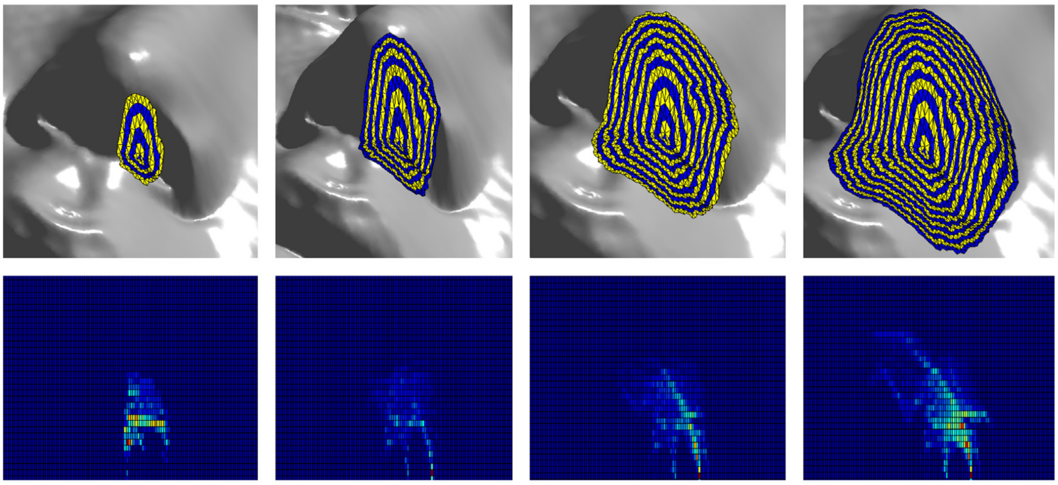
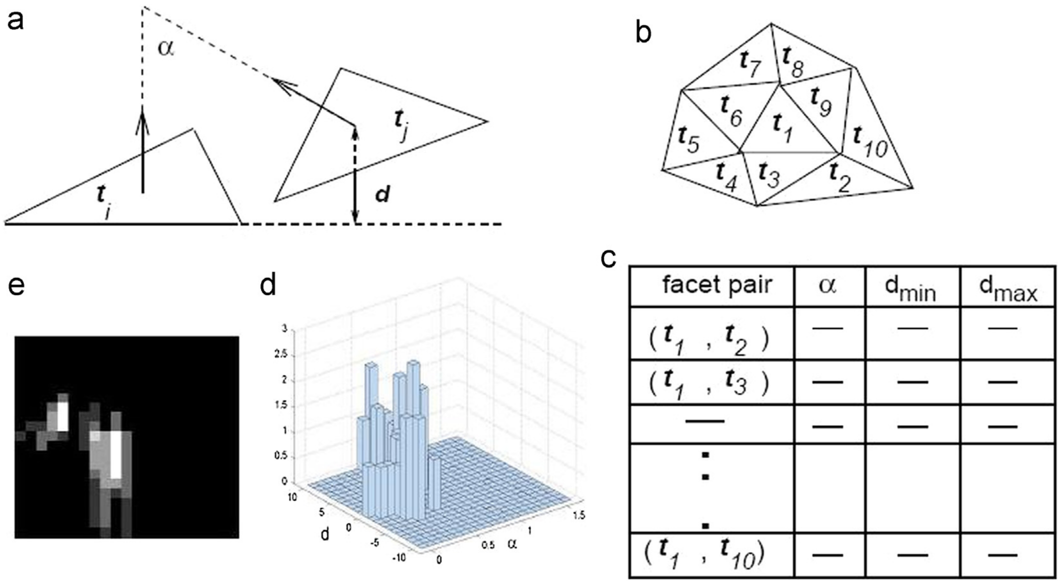


Fig. 4. ORF neighborhoods with different sizes constructed at a facial keypoints near to the nose, and their corresponding GHs.



facet pair	α	d_{min}	d_{max}
(t_1, t_2)	—	—	—
(t_1, t_3)	—	—	—
—	—	—	—
⋮			
(t_1, t_{10})	—	—	—

Fig. 5. (a) The geometric measurements used to characterize the relationship between two facets t_i and t_j. (b) A facet t_1 and its neighbor facets. (c) For each pair (t_1, t_s), $s = 1, \ldots, 10$, the angle α between the two facets' normals, the minimal and the maximal of the perpendicular distance from the plane of t_1 to the facet t_s are computed. (d) The pairs (α, d) derived from these measurements are entered in a 2D accumulator, obtaining thus a geometric distribution that characterizes the relationship between the facet t_1 and its neighbors. (e) The geometric distribution can be visualized with a gray level mapping.

Ordered ring facets. The Ordered Ring Facets (ORF) [56] is the method used to identify the facets of the mesh which are comprised in the neighborhood of a keypoint. In this approach, the neighborhood construction around a central facet t_c is performed through a sequence of concentric rings of facets emanating from a root facet (i. e., t_c). The facets are arranged circular-wise within each ring. The size of the neighborhood is simply controlled by the number of rings. This mechanism allows an easy analysis of the GH variability, and thus of the local geometry evolution, as the size of the neighborhood increases. When the triangular mesh is regular and the facets are nearly equilateral, the ORF rings form an approximation of iso-geodesic rings around the central facet t_c. The ORF construction has a linear complexity. Fig. 4 depicts examples of ORF's with increasing number of rings and their related GH's. In the experiments reported in Section 5, we obtained good results by using 8 ORF as neighborhood of the keypoints.

Discrete distribution. Consider a triangular mesh approximation $\hat{S} = \{t_1, \ldots, t_M\}$ of an object surface. The discrete geometric distribution is constructed for each triangular facet t_i in a given mesh which describes its pairwise relationship with each of the other surrounding facets within a predefined neighborhood. The range of the neighborhood controls the degree to which the representation is a local description of shape. Here, we choose a neighborhood range that encompasses the facets that share one or two vertices with the central triangular facet (Fig. 5(b)). The distribution is defined such that it encodes the surrounding shape geometry in a manner which is invariant to rigid transformations of the surface data and which is stable in the presence of surface clutter and missing surface data.

Fig. 5(a) shows the measurements used to characterize the relationship between facet t_i and one of its neighboring facets t_j. These measurements are the relative angle, α, between the facet normals, and the range of perpendicular algebraic distances, d, from the plane in which facet t_i lies to all points on the facet t_j. The range of perpendicular algebraic distances is defined by $[d_{min}, d_{max}]$, where d_{min} and d_{max} are the minimal and the maximal of the distance from the plane, respectively, in which t_i lies to the facet t_j. These extreme entities are simply obtained by calculating the distances to three vertices of the facet t_j and then selecting the minimal and the maximal distances.

Since the distance measurement is a range rather than a single value, from each measurement $(\alpha, d_{min}, d_{max})$ can be derived a number of measurements (α, d) $(d_{min} \leq d \leq d_{max})$. This number depends on the amplitude of the range $[d_{min}, d_{max}]$ and the resolution adopted for the distance parameter d. The group of pairs (α, d), extracted from the measurements related to a given facet and its neighbors (Fig. 5(b) and (c)), are entered to a 2D discrete frequency accumulator that encodes the perpendicular distance d and the angle α (Fig. 5(d)). This accumulator has size $N \times M$, where N and M are the number of bins in the axis α and d, respectively. The values of the accumulated matrix are also normalized so as to sum up to 1. The accumulator can be visualized in a 2D plotting using a gray level colormap (Fig. 5(e)), and stored in a matrix for subsequent processing. This representation only depends upon the surface shape and not on the placement of facets over the surface. This independence on the placement of the facets is important as it guarantees the invariance of the correspondence with respect to geometric transformations. A possible variant of the geometric histogram is obtained by considering all the pairs of facets within N_{t_c}, i.e., the set $\{(t_i, t_j),\ t_i \in N_{t_c},\ t_j \in N_{t_c}\}$. The construction of this variant is computationally more demanding as the number of histogram entries evolves quadratically with respect to the number of facets in the neighborhood. Due to this, in our experiment we considered the computation referred to the central facet t_c, using $N=8$ and $M=20$.

With respect to the computation of the central GH, we introduced a variant which is related to the ORF definition. In particular, in our approach, a GH is constructed on each of the rings that constitute the ORF of a keypoint: This means that the GH descriptor is actually given by a set of GH, constructed on the sequence of rings which surround the keypoint. This improves the descriptiveness of GH by capturing information on how the local characteristic of the surface changes when the distance from the keypoint increases. This multi-ring structure is also exploited during the match. In particular, the normalized GH can be viewed as a probability density function, and thus can be adapted to probabilistic matching paradigms. To this end, the Bhattacharyya distance (d_B) is used as metric for evaluating the similarity between GHs at each ring. According to this, given two GHs in the form of 1D arrays of $K = N \times M$ elements, $A(l) = \{a_1, \ldots, a_K\}$ and $B(l) = \{b_1, \ldots, b_K\}$, their distance at ring-l is computed as

$$d_B(A(l), B(l)) = \sqrt{1 - \sum_{k=1}^{K} \sqrt{(a_k \cdot b_k)}}. \tag{8}$$

The overall distance between two multi-ring GH, computed on L rings is then obtained by accumulating the distances between the GHs at different rings, that is

$$d(A, B) = \sum_{l=1}^{L} d_B(A(l), B(l)). \tag{9}$$

The Experiment 2 Code Item 2 can be executed on the Experiment 2 Data Item 1, in order to generate the GH descriptor of a 3D face scan.

4. Face matching

Given two face scans, their comparison is performed by matching the local shape descriptors at corresponding keypoints under the constraint that a consistent spatial transformation exists between inliers pairs of matching keypoints. To this end, local shape descriptors at the keypoints detected in probe and gallery scans are compared so that for each keypoint in the probe, a candidate corresponding keypoint in the gallery is identified. In particular, a keypoint k_p in the probe is assigned to a keypoint k_g in the gallery, if they match each other among all keypoints, that is, if and only if k_p is closer to k_g than to any other keypoint in the gallery and k_g is closer to k_p than to any other keypoint in the probe. For this purpose, distance between keypoints descriptors is measured through the distances presented for the three local descriptor HOG, SHOT and GH, discussed, respectively, in Sections 3.1–3.3. Finally, the candidate matches for which the second best match is significantly worse are accepted (i.e., a match is accepted if the ratio between the distance of the best match and the second best match is lower than 0.7).

This analysis of proximity of keypoint descriptors results in the identification of a candidate set of keypoint correspondences. Identification of the actual set of keypoint correspondences must pass a final constraint targeting the consistent spatial transformation between corresponding keypoints in the probe and gallery scans. The RANSAC algorithm [43,57] is used to identify outliers in the candidate set of keypoint correspondences. This involves generating transformation hypotheses using a minimal number of correspondences and then evaluating each hypothesis based on the number of inliers among all features under that hypothesis. In our case, we modeled the problem of establishing correspondences between sets of keypoints detected on two matching scans as that of identifying points in $\Re^3$ that are related via a rotation, scaling and translation transformation (RST transformation). According to this, at each iteration, the RANSAC algorithm validates sampled pairs of matching keypoints under the current RST transformation hypothesis, updating at the same time the RST transformation according to the sampled points. In this way, corresponding keypoints whose RST transformation is different from the final RST hypothesis are regarded as outliers and are removed from the match. Examples of the application of RANSAC are reported in Fig. 6. In the figure, detected keypoints are highlighted with a "+" symbol (in blue); corresponding keypoints based on descriptors matching are connected by green lines; finally, the inliers matching which pass the RANSAC algorithm are shown with a red line connection. It can be observed as by applying the RANSAC algorithm just the matches that show a coherent RST transformation among each other is retained. This avoids matches of keypoints that are located in different parts of the face of two scans. Cases in (a) and (b), respectively, report the match between two scans of the same subject and of different subjects. In Fig. 7, we also report the case in which scans of the same subject with large missing parts (a) and with expression (b) are matched against a full neutral gallery scan. It can be observed as the number of inliers is still high compared to that of different subjects, despite the large missing parts and expression.

Once the set of inlier keypoints is established, the distance between their descriptors is accumulated and averaged. Given a probe and a gallery, the correspondences identified by the spatial transformation hypothesis is a function $\xi : \aleph \mapsto \aleph$ that associates with a keypoint descriptor $\mathcal{C}_k^{(p)}$ in the probe, its corresponding keypoint descriptor $\mathcal{C}_{\xi(k)}^{(g)}$ in the gallery. For each keypoint descriptor in the probe $\mathcal{C}_k^{(p)}$ the distance to the corresponding keypoint descriptor $\mathcal{C}_{\xi(k)}^{(g)}$ in the gallery is evaluated (using Eq. (7) for SHOT or Eq. (9) for GH), and these distances are finally averaged on the total number of inlier matches N_i

$$D = \frac{1}{N_i} \sum_{k=1}^{N_i} \mathcal{D}(\mathcal{C}_k^{(p)}, \mathcal{C}_{\xi(k)}^{(g)}). \tag{10}$$

In this way, the distance between two face scans is regarded as a pair $\langle N_i, D \rangle$. The number of matching inliers is used as measure of distance. In the case two scans have the same number of inliers, the distance D serves as disambiguation value.

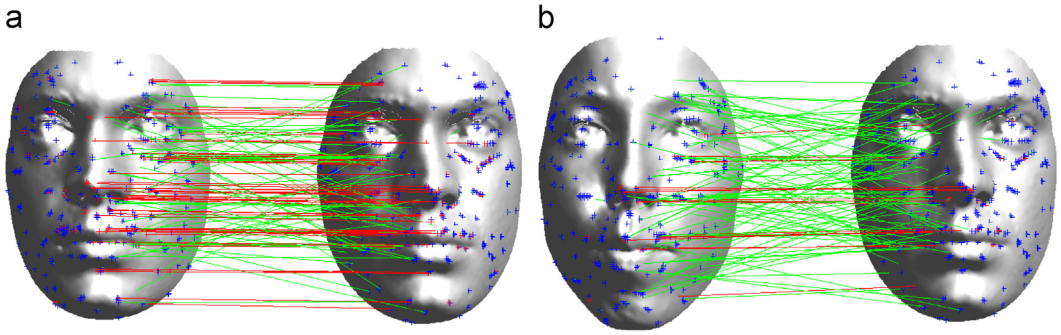


Fig. 6. Matching of scans of same and different subjects are reported in (a) and (b), respectively. All the detected keypoints are shown with "+". Lines indicate matching keypoints (in green), and inliers matching after RANSAC (in red). In the case of scans of the same subject in (a), 61 inlier matches are identified; For scans of different subjects in (b), 18 matches are detected. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

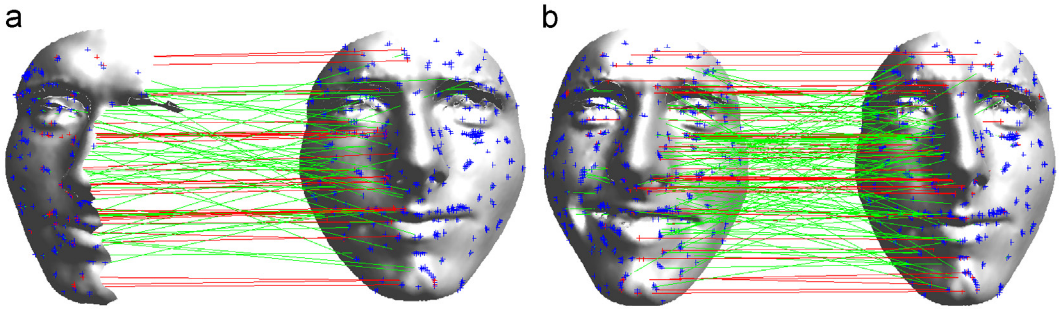


Fig. 7. (a) Partial probe vs. full gallery same subject (34 inliers). (b) Expressive probe vs. neutral gallery same subject (47 inliers). (a) same subject: missing parts and (b) same subject: expression.

The Experiment 3 Code Item 2 can be executed on the Experiment 3 Data Item 1 and Experiment 3 Data Item 2, in order to compute the match between two 3D face scans using local descriptors and RANSAC. An image showing the keypoints matching is also generated.

About the computational complexity of the proposed matching approach, it depends on two main cost factors: the matching of local descriptors and the execution of the RANSAC algorithm. The first term resulted the main source of cost, growing quadratically with the number of keypoints in the two scans. All the three descriptors presented in Section 3 are histogram based and so the complexity in computing their match depends on the distance measure and on the number of histogram bins.

5. Experimental results

The performance of the proposed approach has been evaluated in a comprehensive set of experiments. For the sake of the presentation and discussion, experiments have been divided and organized into two parts:

1. The goal of the first session of experiments was to evaluate the robustness of our 3D face recognition solution to probes showing large facial expressions (from moderate to exaggerated), and extreme pose variations (side rotations of 90°). To this end, experiments were carried out on two datasets that are specifically designed for investigating 3D face recognition in the presence of facial expressions, *The Binghamton University 3D Facial Expression database* (BU-3DFE) [44], and missing parts, *The 2D/3D Florence Face dataset* (UF-3D) [45]. In addition, we provide an in depth investigation on the keypoints detection and repeatability, using the same datasets. Results of this first session of experiments are reported in Section 5.1.
2. In the second session of experiments, the proposed approach is evaluated on a variety of benchmark datasets that differ in the number of scans, acquisition modalities and characteristics of the scans in terms of missing parts, occlusions, and expressions. The used databases are the *Bosphorus* [28], *Gavab* [14] and *UND/FRGC v2.0* [6]. These datasets have been used by many of the existing 3D face recognition works, thus permitting a direct comparison of our approach with state of the art solutions. Section 5.2 reports results of this evaluation.

The datasets listed above largely differ in the scanners used during acquisition (i.e., either laser or structured light scanners), so that both 2.5D (only one z-value is possible at a given xy location) and 3D acquisitions are involved (multiple z-values at the same xy location are allowed). According to this, in the perspective of not to restrict the proposed approach to any particular scenario, in the experimentation we do not make any assumption about the type of scans available in the probe or gallery sets (i.e., they can be either 2.5D or 3D).

5.1. Performance evaluation

The objective of the results reported in this section is to verify the performance of the proposed approach in the case of probes with very large facial expressions (Section 5.1.1), and extreme side rotations (Section 5.1.2). In so doing, we devised an *identification scenario* where the effectiveness of recognition is measured through the rank-k *recognition rate* (RR): a rank-k recognition experiment is successful if the gallery face representing the same individual of the current probe is ranked within the first k positions of the ranked list. The rank-1 value has been reported in our experiments.

5.1.1. The BU-3DFE database

The BU-3DFE database was recently constructed at Binghamton University [44]. It has been designed to provide 3D facial scans of a large population of different subjects each showing a set of facial expressions at various levels of intensity. There are a total of 100 subjects in the database, divided between female (56 subjects) and male (44 subjects). The subjects are well distributed across different ethnic groups or racial ancestries, including *White*, *Black*, *East-Asian*, *Middle-East Asian*, *Hispanic-Latino*, and others. During the acquisition, each subject was asked to perform the neutral facial expression as well as the six basic facial expressions defined by Ekman [58], namely *anger*, *disgust*, *fear*, *happiness*, *sadness*, and *surprise*. Each facial expression has four levels of intensity, respectively *low*, *middle*, *high* and *highest*, except the neutral facial expression that has only one intensity level. Thus, there are 25 3D facial expression scans for each subject, resulting in 2500 3D facial expression scans in the database. As an example, Fig. 8 shows the 3D scans of a sample subject showing the six basic facial expressions at the *low* and *medium* levels of intensity.

Face recognition results. The BU-3DFE dataset has been used to investigate the robustness of the proposed approach with respect to facial expressions in a wide range of intensity variations, from low to exaggerated. This allowed us to infer some evidence of the facial variations that mostly affect face recognition. So far, the BU-3DFE database has been used mainly to test facial expression recognition methods, rather than the robustness of face recognition methods in the presence of expression variations. Actually, face recognition experiments on the BU-3DFE were conducted in [59,60], but only cumulated results were reported in these works, without a detailed analysis for each expression/intensity. As a consequence, for the large part of the methods reported in the literature, there is no insight of the effect induced by different expressions.

In our experiments, we randomly partitioned the dataset into a training and a testing set. The scans of 20 subjects have been included in the train set and have been used for tuning the parameters of the 3D keypoints detector (i.e., the number of DOG scales, the percentage and cornerness thresholds, see Section 2) and the local descriptors (i.e., number of histogram bins for HOG, SHOT and GH descriptors, see Section 3). A classic grid search approach has been used to this end (this phase is mainly important for keypoints detection, since the percentage and cornerness thresholds largely influence the number of detected keypoints, which can vary of an order of magnitude or so). These parameters have been used in the experiments carried out on this dataset, on the UF-3D database (as reported in the next section) and on the three databases used in Section 5.2. The scans of the remaining 80 subjects have been included in the test set. In particular, we considered the neutral scan of each subject as a reference scan and included it in the gallery set (gallery with 80 neutral scans in total). The probe set is composed of 24 expressive scans for each subject, including for each expression the scans with *low*, *medium high* and *highest* intensity level (see Fig. 8). With this selection, the probe set includes 1920 expressive probe scans. The scans classified as showing a *low* and *medium* expression intensity have moderate and natural expressions, similar to those that are likely to occur in a real context. Instead, scans classified in the BU-3DFE as having *high* and *highest* expression intensity, present quite exaggerated expressions for the large part of the subjects, and are more suited to verify the performance of the approach in very difficult situations.

Using these probe and gallery sets, we performed recognition experiments based on keypoints matching with each of the three local descriptors presented in Section 3. Rank-1 recognition accuracies are reported in Table 1, separately for the six expressions, and for the low and medium intensity level (L1 & L2), and the high and highest level (L3 & L4). From the table, it can be observed that, as the overall performance is concerned, the SHOT descriptor provides the best results among the three local descriptors. Looking in to the performance of the SHOT descriptor, it results that the expression that makes the recognition more difficult is the *surprise* one at L1 & L2. This is confirmed also using the HOG and GH descriptors. This is mainly due to the open mouth

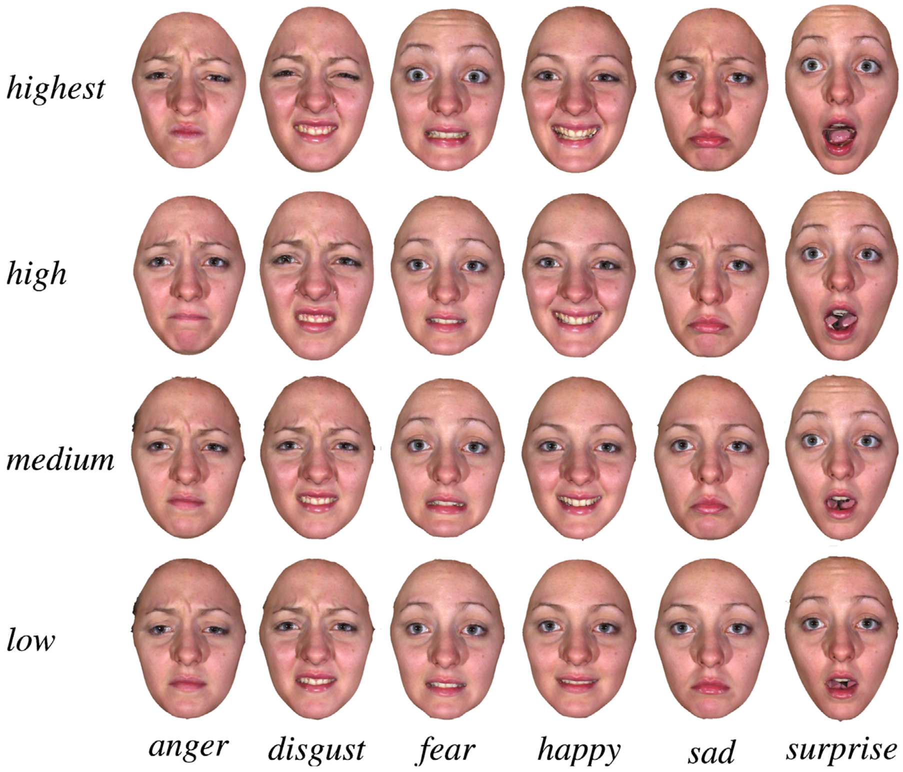


Fig. 8. BU-3DFE: 3D face scans (with texture) of a sample subject showing the six basic facial expressions at the *low*, *medium*, *high* and *highest* level of intensity.

Table 1
BU-3DFE: rank-1 recognition rate (RR) for different expressive scans. Results are reported separately for the HOG, SHOT and GH descriptors. For each descriptor, the average for the *low* and *medium* expression intensity (L1 & L2), and for the *high* and *highest* intensity level (L3 & L4) are reported, together with the average on all the intensity (*All* column).

Expression	Rank-1 RR								
	HOG			**SHOT**			**GH**		
	L1 & L2 (%)	L3 & L4 (%)	All (%)	L1 & L2 (%)	L3 & L4 (%)	All (%)	L1 & L2 (%)	L3 & L4 (%)	All (%)
Angry	90.0	81.3	85.6	93.8	87.5	90.6	90.6	86.3	88.4
Disgust	87.5	75.6	81.6	90.6	78.8	84.7	85.0	79.4	82.2
Fear	88.8	78.8	83.8	91.9	85.6	88.8	84.4	80.0	82.2
Happy	88.1	80.6	84.4	90.0	79.4	84.7	85.6	79.4	82.5
Sad	90.6	82.5	86.6	94.4	90.0	92.2	90.6	85.0	87.8
Surprise	85.0	76.9	80.9	88.8	79.4	84.1	82.5	78.8	80.6
Overall	88.3	79.3	83.8	91.6	83.4	87.5	86.5	81.5	84.0

that appears in the large part of subjects with this expression. The effect of this is a modification of both the location of the detected keypoints with respect to the neutral case, as well as a change of the local descriptors. At L3 & L4 also faces with *disgusted* expression become difficult to be recognized. Furthermore, from this analysis also results that the performance with the GH descriptor seems to degrade more gracefully than for the other descriptors when passing from L1 & L2 to L3 & L4.

5.1.2. The 2D/3D Florence face dataset

The 2D/3D Florence face dataset (UF-2D/3D)[1] has been constructed at the Media Integration and Communication Center of the University of Florence [45]. The dataset consists of high-resolution 3D scans of human faces along with several video sequences of varying resolution and zoom level. This dataset is designed to simulate, in a controlled fashion, realistic surveillance conditions and to test the efficacy of exploiting 3D models in real scenarios. In this work, we used the 3D part of the dataset (UF-3D) that currently includes 53 subjects (14 females and 39 males, numbered from *subject001* to *subject053*) of Caucasian ethnicity. The age of the subjects ranges from 20 to 60, with the majority of the subjects (28) being student at the School of Engineering of the University of Florence, aged between 20 and 30 years. The 3D scans of each subject are acquired in the same session and include two frontal scans with neutral expression (named as *frontal1* and *frontal2*), and two scans where the subject is rotated of 90° on the left and right sides (named *left* and *right*, respectively). In all the acquisitions, the subjects are required to assume a neutral expression, though some scans exhibit moderate, involuntary, facial expressions. The *3dMD face system* [10] scanner has been used in the acquisition, which produces one continuous point cloud from two stereo cameras with a capture speed of about 1.5 ms at the highest resolution, and a geometry accuracy lower than 0.2 mm RMS. As an example, Fig. 9 reports the 3D face scans of two sample subjects.

Face recognition results. The UF-3D dataset allows us to evaluate the recognition accuracy of the proposed solution in the case of frontal neutral probes as well as for probes with extreme yaw rotations. In particular, the left and right probes in this dataset have been acquired with side rotation of 90°, which results in scans with half of the face missing, with consequent very challenging recognition conditions. One neutral scan ("frontal1") has been selected as reference for each subject and included in the gallery. The other neutral scan of each subject ("frontal2") has been used as probe in the "neutral vs. neutral" experiment. The left/right scans have been used in two separate experiments aiming to test the robustness of the proposed approach to partial face matching, where large parts of the face are missing. It is relevant to note that being the proposed approach based completely on 3D processing, both keypoints detection and local description extraction can be performed without the need of costly pose normalization solutions that are required by other existing methods [23,24,29,35].

Results of this evaluation are reported in Table 2. It can be observed that the proposed solution achieves a very high accuracy in matching neutral frontal scans, with each of the three experimented descriptors showing a similar behavior (in this case the SHOT descriptor achieves the best results). For side scans, the accuracy drops significantly with similar results obtained for the left and right scans. The GH descriptor evidences the highest accuracy in this experiment. To the best of our knowledge, the only two other works reporting results on probes with yaw rotations of 90° are those in [36,46], though these two approaches were experimented on the Bosphorus database. Direct comparison of our solution with respect to [36,46] on the Bosphorus database is given in Section 5.2.1.

Fig. 10 shows two examples of wrong recognition for probes with large missing parts. In both the cases, the number of inliers resulted too low to allow rank-1 recognition. For the case on the left, this can be motivated by the presence of a facial expression (see the open mouth) which is combined with a large part of the face missing. In the case on the right, the main problem was originated by the preprocessing operation, which closes holes in the face scans. Due to the large extent of the hole, the hole filling procedure fails in producing a consistent closing, thus altering the face geometry and the keypoints extraction and description.

5.1.3. Localization and repeatability of 3D keypoints

The idea of representing the face by a sparse and adaptive set of automatically detected keypoints relies on the assumption of *intra-subject keypoints repeatability*: Keypoints extracted from different facial scans of the same individual are expected to be located approximately in the same positions of the face. Since keypoints detection only depends on the geometry of the face surface through its mean curvature (see Section 2), these keypoints are not guaranteed to correspond to specific meaningful landmarks of the face. For the same reason, the detection of keypoints on two face scans of the same individual should yield to the identification of the same points of the face, unless the shape of the face is altered by major occlusions or non-neutral facial expressions.

To test the repeatability of keypoints detection, we used the 3D scans of the BU-3DFE database selected for the experiments reported in Section 5.1.1. We followed the approach proposed in [31], and measured the correspondence of the location of keypoints detected in two face scans by performing ICP registration. Accordingly, the 3D faces belonging to the same individual are automatically registered and the errors between the nearest neighbors of their keypoints (one from each face) are recorded. Fig. 11 shows the results of our keypoint repeatability experiment, by reporting the cumulative rate of repeatability as a function of increasing values of the distance. The repeatability reaches a value of 90% for frontal faces with neutral and non-neutral expressions at a distance error of 5 mm (with an average number of 360 keypoints detected per scan). We remark that these results, and those reported in the following about the number of detected keypoints, have been obtained by computing 96 DOG scales, and retaining the unique keypoints that are detected in the last 64 DOG scales (see also Section 2).

[1] The database is publicly available and can be accessed upon request from the following address: http://www.micc.unifi.it/masi/research/ffd/. The dataset is also released within the Elsevier Collage Authoring Environment.

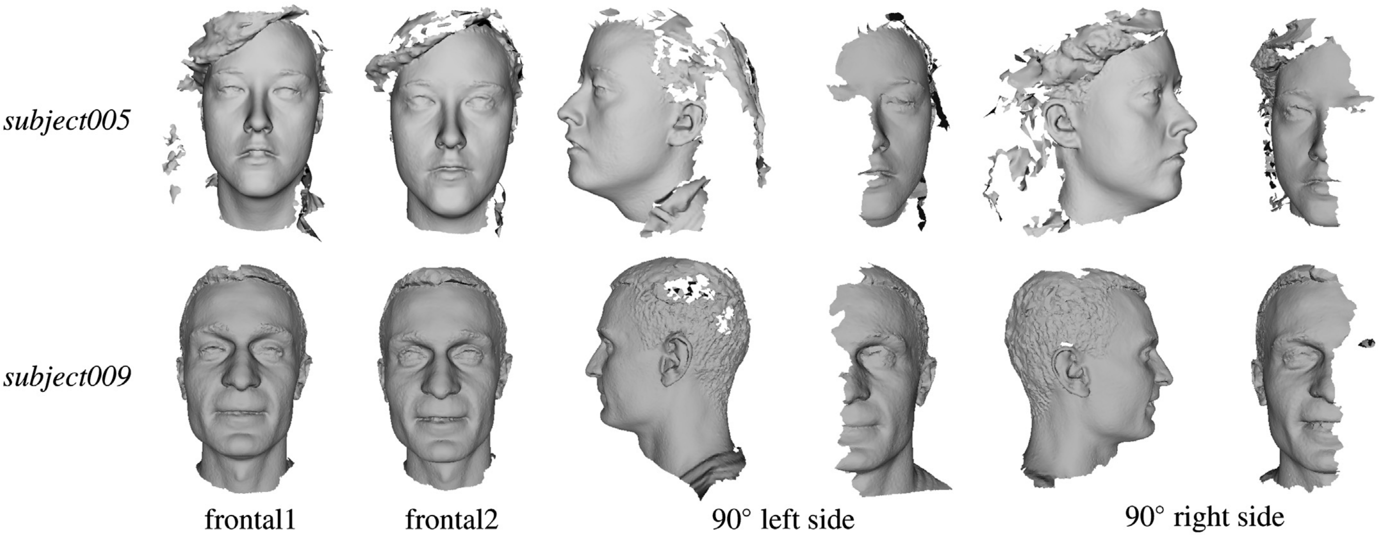


Fig. 9. UF-3D: 3D face scans of two sample subjects. For the left and right cases, the acquired scan is shown as well as its frontal view so has to evidence the missing amount of the facial surface.

Table 2
UF-3D: rank-1 RR for frontal neutral and left/right probes.

Local descriptor	Rank-1 RR	Rank-1 RR		
	Overall (%)	Frontal (%)	Left (%)	Right (%)
HOG	64.8	92.5	49.1	52.8
SHOT	69.2	96.2	54.7	56.6
GH	71.1	94.3	58.5	60.4

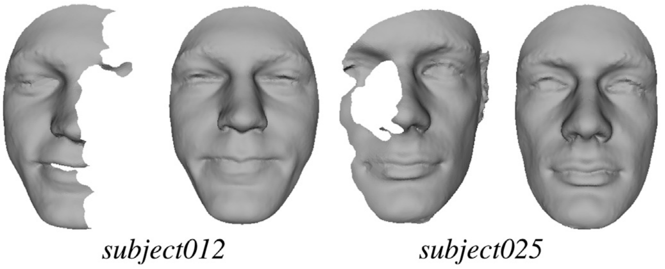


Fig. 10. UF-3D: Example of scans with missing parts that are not recognized when matched against corresponding full gallery scans.

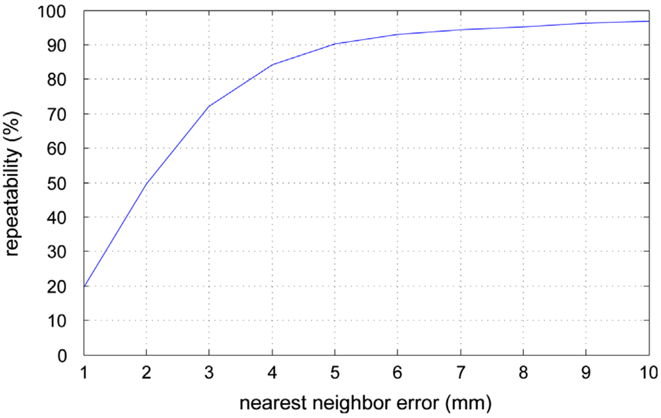


Fig. 11. Repeatability of keypoints.

Table 3
Number of detected keypoints per scan (average, min and max).

Dataset		*Number of keypoints*		
Name	Scans	Avg	Min	Max
UF-3D *frontal*	106	445	346	572
UF-3D *left/right*	106	205	130	396
UF-3D *total*	212	325	130	572
BU-3DFE *neutral*	80	327	265	402
BU-3DFE *expressive*	1920	361	292	464
BU-3DFE *total*	2000	360	265	464

Table 3 also reports the number of keypoints detected on the face scans of the BU-3DFE and the UF-3D datasets. In particular, separate values are given for the *average*, *minimum* and *maximum* number of keypoints. As expected, it can be observed that the largest number of keypoints is detected in the gallery and frontal probes with neutral expression, whereas the number of detected keypoints decreases for side scans. No remarkable differences are observed for the number of keypoints detected on left or right probes. Non-neutral expressions have a small impact on the number of detected keypoints, which remains comparable to that obtained for frontal neutral scans (in some cases, an increase in the number of keypoints is observed).

From Table 3, it results that the number of detected keypoints is quite large. In fact, an important trait of a keypoints detector is the amount of repeatable keypoints it can provide to the subsequent modules of an application. Detecting a small number of keypoints cannot be enough to apply geometrical verification or outliers removal steps, whereas too many may waste computational resources [48]. In the case of meshDOG, the number of detected keypoints is the result of the thresholds involved in the detection algorithm (see Section 2). Of course, making these thresholds more selective, the number of keypoints can be reduced. In our experiments, the number of keypoints reported in Table 3 represented a good compromise between computational cost and accuracy of recognition. A number of detected 3D keypoints on 3D face scans of the order of hundreds are also reported for the 3D keypoints detector defined by Mian et al. [31], and for the meshSIFT detector [39,46]. These results seem to support our findings. For example, in the meshSIFT, an average number of about 560 keypoints is reported by the authors, with a number of matching at rank-1 of about 97. The recent survey on the evaluation of 3D keypoint detectors [48], also reported that meshDOG tends to extract a high number of keypoints, that accumulate around areas characterized by high local curvature.

5.2. Comparative evaluation

In this section, the proposed approach is evaluated and compared to state of the art solutions on three benchmark databases: Bosphorus, Gavab and UND/FRGC v2.0. Based on the analysis of Section 5.1, in the following we provide results of our approach only for the GH descriptor. In fact, we found that the GH descriptor provides a good balance of recognition performance between the cases of probes with missing parts and probes with large facial expressions.

5.2.1. The Bosphorus 3D face database

The Bosphorus database has been collected at the Boğaziçi University and made available during 2008 [16]. It consists of 3D facial scans and images of 105 subjects acquired under different expressions and various poses and occlusion conditions. Occlusions are given by hair, eyeglasses or predefined hand gestures covering one eye or the mouth. Many of the male subjects have also beard and moustache. The majority of the subjects are Caucasian aged between 25 and 35, with a total of 60 males and 45 females. The database includes a total of 4666 face scans, with the subjects categorized as follows:

- About 34 subjects with up to 31 scans per subject (including 10 expressions, 13 poses, 4 occlusions and 4 neutral).
- About 71 subjects with up to 54 different face scans. Each scan is intended to cover one pose and/or one expression type, and most of the subjects have only one neutral face, though some of them have two. Totally, there are 34 expressions, 13 poses, 4 occlusions and one or two neutral faces. In this set, 29 subjects are professional actors/actresses, which provide more realistic and pronounced expressions.

Face recognition results and comparative evaluation. In our experiments, we used the same experimental protocol proposed in [36,46], thus allowing a direct comparison of the results. For each subject, the first neutral scan was included in the gallery, whereas the probe scans have been organized in different classes as reported in Table 4 (the number of probes per class is also indicated). The first class groups probe according to their facial expression, distinguishing between neutral probes and expressive probes categorized according to the six expressions defined by Ekman [58], plus some not-classified probes. Probes where subjects exhibit face action units are accounted in the second class, by considering scans with Lower Face Action Unit (LFAU), Upper Face Action Unit (UFAU), and Combined Action Unit (CAU). Finally, the last class reports probes with missing parts due to Yaw Rotation (YR), Pitch Rotation (PR) and Cross Rotation (CR), plus probes with Occlusions (O). For the methods in [36,46] we provide the rank-1 RR accuracy as reported in the respective publications.

Table 4
Bosphorus: rank-1 RR for different probe classes. Results of our approach are compared with those reported in [36,46].

Probes (#)	Li et al. [36] % rank-1RR	Smeets et al. [46] % rank-1RR	**This work** rank-1RR
Neutral (194)	100.0	–	97.9
Anger (71)	88.7	–	85.9
Disgust (69)	76.8	–	81.2
Fear (70)	92.9	–	90.0
Happy (106)	95.3	–	92.5
Sad (66)	95.5	–	93.9
Surprise (71)	98.6	–	91.5
Other (18)	–	–	100.0
LFAU (1549)	97.2	–	96.5
UFAU (432)	99.1	–	98.4
CAU (169)	98.8	–	95.6
YR (735)	78.0	–	81.6
PR (419)	98.8	–	98.3
CR (211)	94.3	–	93.4
O (381)	99.2	–	93.2
All (4561)	94.1	93.7	93.4

From the table, we first note that the approach by Li et al. [36] reports a detailed analysis for the different probe categories, whereas in Smeets et al. [46] results are presented in a cumulative way. Results show that our approach has overall performance which are very close to state of the art solutions, and for some category are even better. In particular, our solution performs particularly well in recognizing scans with missing parts (see for example the YR category). More in detail, our approach achieves an accuracy of 45.7% on scans with ±90° left/right yaw rotations. Results for these scans are not reported directly in [46]. However, authors also reported the overall recognition in the case the ±90° scans are removed. So, it is possible to derive the accuracy of [46] on ±90° scans to be around 25%.

We guess the lower performance achieved in [46] on scans with very large missing parts are mainly due to the way local descriptors are computed. In fact, in [46] the local support used for the computation of the meshSIFT feature is quite large and increases with the scale at which keypoints are detected. As a result, keypoints detected at the highest scales, which in principle are the most stable, have local descriptors which span a large part of the face. This reduces the robustness of the descriptor to missing parts. In our case instead, the local support is quite small thanks to the descriptive capability of the multi-ring GH descriptor, thus making our representation quite robust to missing parts of the face.

5.2.2. Gavab database

The Gavab database [14] comprises facial scans with large pose and expression variations, and noisy acquisitions. It includes 3D face scans of 61 adult Caucasian individuals (45 males and 16 females). For each individual, nine scans are taken that differ in the acquisition viewpoint and facial expressions, resulting in a total of 549 facial scans. In particular, for each individual, there are two frontal face scans with neutral expression, two face scans where the subject is acquired with a rotated posture of the face (around ±35° looking-up or looking-down) and neutral facial expression, and three frontal scans in which the person laughs, smiles, or shows a random expression. Finally, there are also two side scans nominally acquired with a rotation of ±90° left and right. In our experiments, we used all the probes and compared them against the gallery scans. The gallery includes, for each subject, the scan named “frontal1” according to the experimental protocol of this dataset.

Face recognition results and comparative evaluation. On this dataset, our results are compared with those reported in [29,35] that used a similar experimental setup. Table 5 summarizes the evaluation using rank-1 RR. Results demonstrate that our approach is capable of achieving or improving state of the art performance for all the classes of scans. As a general behavior, a quite large difference in recognizing left and right side scans can be noted for this dataset (about 10%, 14% and 16% decrease, respectively, for our work and the approaches in [29,35]). Measuring the yaw rotation for the left and right side scans, we obtained an average angle of about 50° and 70°, respectively. These rotation angles are lower than the nominal values reported in the database description, and the difference of around 20° between left and right rotations motivate the different recognition accuracy in the two cases.

Table 5
Gavab dataset: Comparison between methods reporting partial face matching results on left/right scans. The rank-1 RR is reported (highest RR values are evidenced in bold for each class).

Dataset		Rank-1 RR		
Name	Scans	Drira et al. [29] (%)	Huang et al. [35] (%)	**This work (%)**
Frontal neutral	61	**100.0**	**100.0**	**100.0**
Frontal expressive	183	**94.5**	94.0	94.0
Neutral + expressive	244	94.7	**95.5**	95.1
Looking-down	61	**100.0**	96.7	95.1
Looking-up	61	**98.4**	96.7	96.7
Left side	61	86.9	**93.4**	**93.4**
Right side	61	70.5	78.7	**83.6**

5.2.3. UND/FRGC v2.0 database

We performed experiments on the side facial scans of the ear database from the *University of Notre Dame* (UND) [15], collections F and G. This database was created for ear recognition purposes and contains side scans with yaw rotations of 45°, 60° and 90°. Similarly to [23], we used the 45° side scans (119 subjects, with 119 left and 119 right scans) and the 60° side scans (88 subjects, with 88 left and 88 rights scans). As noted in [23], even if these side scans are marked as 45° and 60° by the creators of the database, the measured average yaw angle of rotation is 65° and 80°, respectively. There is a partial overlap between subjects in the UND and in the FRGC v2.0 databases, but not all subjects exist in both the UND and FRGC v2.0. In fact, the number of common subjects between the gallery scans (i.e., frontal scans in the FRGC v2.0) and the 45° side scans is 39, and between the gallery scans and the 60° side scans is 33. According to the partition of the probes used in [23], in our experiments we considered the following test datasets:

- DB45F: Gallery set has one frontal scan for each of the 466 subjects of the FRGC v2.0; Probe set has 45° left/right side scans for each of the 39 subjects.
- DB60F: Gallery set has one frontal scan for each of the 466 subjects of the FRGC v2.0; Probe set has 60° left/right side scans for each of the 33 subjects.

In both the cases, there is only one gallery scan per subject (466 scans in total), and the gallery coincides with that of the FRGC v2.0 dataset. In addition, all the subjects included in the probe set are also present in the gallery set (the opposite is not always true). In the following, we will also use UND45 left/right and UND60 left/right to refer to the probe sets constituted by the 45° left/right side scans and by the 60° left/right side scans, respectively.

Face recognition results and comparative evaluation. In the following, we compare the proposed solution with the approaches in [23] (*automatic* and *manual*) and [24] that have been evaluated on the UND/FRGC v2.0 following the same experimental setup and protocol. Results of the comparative evaluation are summarized in Table 6 using rank-1 RR. Results are organized in three parts:

- UND45 left/right: At rank-1 the approach in [23] (*manual*) results the most effective. We point out that the solution in [23] can use both automatically and manually detected facial landmarks in order to identify face regions used for face alignment and recognition. Quite interestingly, the accuracy of our solution is very close to the accuracy of the solution relying on manual annotation [23], and higher than the accuracy of the solution relying on automatic detection.

Table 6
UND dataset: Comparison between methods reporting partial face match results on the left and right scans of the UND probes. The RR at rank-1 is reported, with values for individual experiments and their average (*avg*). The highest RR values for each dataset are reported in bold.

Dataset		Rank-1 RR			
		Perakis et al. [23]		Passalis et al. [24] (%)	**This work (%)**
Name	Scans	*Manual* (%)	*Automatic* (%)		
UND45 *left*	39	**92.3**	74.4	–	87.2
UND45 *right*	39	**82.1**	64.1	–	**82.1**
UND45 *avg*	78	**87.2**	69.2	–	84.6
UND60 *left*	33	42.4	42.4	–	**66.7**
UND60 *right*	33	42.4	45.5	–	**69.7**
UND60 *avg*	66	42.4	43.9	–	**68.2**
UND *left avg*	72	69.4	59.7	74.6	**77.8**
UND *right avg*	72	63.9	55.6	**78.9**	76.4
UND *total avg*	144	66.7	57.6	76.8	**77.1**

- UND60 left/right: These results evidence the large improvement in the recognition accuracy (more than 20% at rank-1) that our approach achieves with respect to the other solutions.
- UND left/right (45° plus 60°), UND total: Overall, at rank-1, our approach is competitive with the state of the art solution recently reported in [24].

The comparative evaluation evidences that our solution is capable of achieving and in some cases improve state of the art results in the recognition of partial face scans. This is obtained with a completely automatic solution and at a reasonable computational cost. We also evidence that, unlike the solution in [24], our approach does not rely on any assumption of symmetry of the face to reconstruct its global geometry, but only relies on the match of descriptors extracted at detected keypoints of existing parts of the face. This makes our solution more generally applicable.

6. Discussion and conclusions

In this work, we have proposed an original approach to 3D face recognition based on the idea of capturing local information of the face surface around a set of 3D keypoints detected at multiple scales according to differential surface measurements. The approach, first detects 3D keypoints of the face mesh, then local descriptors are extracted at each keypoint and used to find keypoint correspondences during the match. The approach makes no assumption about the correspondence of detected keypoints to specific landmarks on the face, and therefore it can support the comparison of probe and gallery scans even in the case probe scans represent just a part of the face. To improve the accuracy of keypoints correspondences, a spatial constraint is introduced using the RANSAC algorithm.

A preliminary evaluation carried out on the BU-3DFE and the UF-3D datasets showed the viability of the approach in managing moderate as well as exaggerated facial expressions and extreme rotations of the scans, with consequent absence of large parts of the face. This first round of experiments suggested us to use the multi-ring GH descriptor in the subsequent comparative evaluation that has been extended to the Bosphorus, Gavab and UND/FRGC v2.0 databases. Results of this comparison showed that our solution can compete with state of the art works evidencing a clear advantage in the case of probes with large missing parts. In summary, our view is that the proposed approach presents some interesting solutions in the perspective to make 3D face recognition deployable in real non-cooperative context of use: The approach is fully-3D, reducing to the minimum the need for preprocessing operations, not requiring any costly normalization or alignment; The meshDOG keypoints combined with the multi-ring GH descriptor as proposed in this work, provide a good compromise between robustness to expression changes and missing parts of the face; The inclusion of a statistical technique for outlier removal of matching keypoints largely improves the recognition results.

In perspective, the proposed approach could be further improved by fusing together the local descriptors proposed in this work so as to exploit and combine their strengths. Furthermore, the proposed framework can be easily adapted to include texture information of the face surface, so as to define a multi-modal solution that can combine together in a *native* way (i.e., at the level of the function used for meshDOG detection) 2D and 3D data.

Acknowledgments

The authors thank Iacopo Masi at the University of Firenze for making available the 2D/3D Florence face database, and Emiliano Mazzoncini at the University of Firenze for writing part of the code for meshDOG/meshHOG keypoint detection and description.

Appendix A. Operations on the mesh

In order to make this work self-comprehensive, in the following we summarize the main operations performed on the mesh surface that we used in the paper (according to the analysis in [41]). In so doing, we consider uniformly sampled triangulated meshes S, that is meshes whose facets are triangles of approximately the same area and whose vertices have a valence close to 6 (the vertex's valence being defined as the number of edges incident on it). Simple mesh operations can be applied to transform a non-uniform mesh into a uniform one [61].

A mesh S is viewed as a pair $\langle V, E\rangle$, where $V=\{v_i\}_{i=1,\ldots,N}$ is the set of mesh vertices (with $\mathbf{v_i}$ we indicate the 3D point associated to the vertex v_i, i.e., $\mathbf{v}_i \in \mathcal{R}^3$), and $E=\{e_{ij}\}$ is the set of mesh edges between adjacent vertices. The ring of a vertex $ring(v_i,n)$ is the set of vertices that are at distance n from v_i on S, where the distance n is the minimum number of edges between two vertices. Thus $ring(v_i,0)$ is the vertex v_i itself, and $ring(v_i,1)$ is the set of direct neighbors of v_i. According to this, the neighborhood $N_n(v_i)$ is the set of rings $\{ring(v_i,k)\}_{k=0,\ldots,n}$. We further denote $\overrightarrow{\mathbf{n}}_{v_i}$ the unit vector normal to the surface S at vertex v_i, computed as the average direction of the normals of the triangles incident to v_i.

Given a scalar function f defined on the vertices of a mesh S, that is $f: S\rightarrow\mathcal{R}$, the operations of *directional derivative*, *gradient* and *convolution* of f on the *discrete domain* of the vertices of S can be computed as reported in the following.

Discrete directional derivative. The discrete directional derivative of f on S along the direction of the edge $\overrightarrow{e_{ij}}$ (i.e., the direction of the vector $\overrightarrow{v_i v_j}$ originating in v_i and oriented from v_i to v_j) is defined as

$$D_{\overrightarrow{e_{ij}}} f(\mathbf{v}_i) = \frac{1}{\|\mathbf{v}_i - \mathbf{v}_j\|} \cdot (f(\mathbf{v}_j) - f(\mathbf{v}_i)), \tag{A.1}$$

with $v_j \in ring(v_i,1)$, and using the fact that up to the first order $f(\mathbf{v}_j) - f(\mathbf{v}_i) = \nabla_S f(\mathbf{v}_i)\cdot(\mathbf{v}_j - \mathbf{v}_i)$ around v_i.

Discrete gradient. The gradient operator $\nabla_S f(\mathbf{v}_i)$ of f at vertex $\mathbf{v}_i \in S$ is defined as (based on the directional derivatives on v_i)

$$\nabla_S f(\mathbf{v}_i) = \sum_{v_j \in ring(v_i,1)} (w_{ij} \cdot D_{\overrightarrow{e_{ij}}} f(\mathbf{v}_i)) \cdot \overrightarrow{u_{ij}}, \tag{A.2}$$

where w_{ij} weights the contribution of $D_{\overrightarrow{e_{ij}}}$ and $\overrightarrow{u_{ij}}$ is the normalized projected direction of $\overrightarrow{v_{ij}}$ in the tangent plane at v_i. Assuming that S is uniformly sampled and thus that neighbors around v_i are equally spaced we get: $w_{ij}=1/val(v_i)$ where $val(v_i)$ is the valence of v_i (i.e., the number of edges incident on it). For non-uniformly sampled meshes, the weights are a function of the angles between the directions $\overrightarrow{u_{ij}}$ around v_i in the tangent plane at v_i.

Discrete convolution. The convolution of the function f with a kernel h on S is defined as

$$(f*h)(v_i) = \frac{1}{H} \cdot \sum_{v_j \in N_n(v_i)} h(\|\mathbf{v}_i - \mathbf{v}_j\|) \cdot f(\mathbf{v}_j), \tag{A.3}$$

where the kernel weighs the neighboring vertices v_j as a function of their distances from vertex v_i, and $H = \sum_{v_j \in N_n(v_i)} h(\|\mathbf{v}_i - \mathbf{v}_j\|)$ is a normalization factor. Notice that, as for the discrete gradient, a uniformly sampled mesh is assumed. As a consequence, contributions of neighboring vertices v_j in the above expression are equally weighted with respect to their spatial arrangements. In this work, we used the above definition with the first ring only (i.e., $n=1$, so that the vertex v_i and the vertices in its $ring(v_i,1)$ are considered).

References

[1] Phillips PJ, Scruggs WT, O'Toole AJ, Flynn PJ, Bowyer KW, Schott CL, Sharpe M. FRVT 2006 and ICE 2006 large-scale results. In: Technical Report, NISTIR 7408, National Institute of Standards and Technology; 2007.

[2] Bowyer KW, Chang KI, Flynn PJ. A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition. Comput Vision Image Understanding 2006;101(1):1–15.

[3] Berretti S, Del Bimbo A, Pala P. 3D face recognition using iso-geodesic stripes. IEEE Trans Pattern Anal Mach Intell 2010;32(12):2162–77.

[4] Mian AS, Bennamoun M, Owens R. An efficient multimodal 2D-3D hybrid approach to automatic face recognition. IEEE Trans Pattern Anal Mach Intell 2007;29(11):1927–43.

[5] Wang Y, Liu J, Tang X. Robust 3D face recognition by local shape difference boosting. IEEE Trans Pattern Anal Mach Intell 2010;32(12):1858–70.

[6] Phillips PJ, Flynn PJ, Scruggs T, Bowyer KW, Chang J, Hoffman K, Marques J, Min J, Worek W. Overview of the face recognition grand challenge. In: Proceedings of the IEEE workshop on face recognition grand challenge experiments. San Diego, CA; 2005. p. 947–54.

[7] Berretti S, Del Bimbo A, Pala P. Distinguishing facial features for ethnicity-based 3D face recognition. ACM Trans Intell Syst Technol 2012;3(3):1–20.

[8] Ballihi L, Ben Amor B, Daoudi M, Srivastava A, Aboutajdine D. Boosting 3D-geometric features for efficient face recognition and gender classification. IEEE Trans Inf Forensics Secur 2012;7(6):1766–79.

[9] Artec. ⟨http://www.artec3d.com⟩.

[10] 3dMD. ⟨http://www.3dmd.com⟩.

[11] Kinect. ⟨http://www.xbox.com⟩.

[12] Berretti S, Del Bimbo A, Pala P. Superfaces: a super-resolution model for 3D faces. In: Proceedings of the workshop on non-rigid shape analysis and deformable image alignment (NORDIA'12). Firenze, Italy; 2012. p. 73–82.

[13] Sandbach G, Zafeiriou S, Pantic M, Rueckert D. Recognition of 3D facial expression dynamics. Image Vision Comput 2012;30(10):762–73.

[14] Moreno AB, Sánchez Á. Gavabdb: A 3D face database. In: Proceedings of the workshop on biometrics on the Internet. Vigo, Spain; 2004. p. 75–80.

[15] University of Notre Dame, University of Notre Dame biometrics database; 2008. ⟨http://www.nd.edu/@cvrl/UNDBiometricsDatabase.html⟩.

[16] Savran A, Alyüz N, Dibeklioğlu H, Çeliktutan O, Gökberk B, Sankur B, Akarun L. Bosphorus database for 3D face analysis. In: Proceedings of the first COST 2101 workshop on biometrics and identity management; 2008.

[17] Besl PJ, Mc Kay ND. A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 1992;14(2):239–56.

[18] Lu X, Jain AK, Colbry D. Matching 2.5D face scans to 3D models. IEEE Trans Pattern Anal Mach Intell 2006;28(1):31–43.

[19] Ben Amor B, Ardabilian M, Chen L. New experiments on ICP-based 3D face recognition and authentication. In: Proceedings of the international conference on pattern recognition (ICPR'06), vol. 3. Hong Kong; 2006. p. 1195–9.

[20] Bronstein AM, Bronstein MM, Kimmel R. Robust expression-invariant face recognition from partially missing data. In: Proceedings of the European conference on computer vision. Gratz, Austria; 2006. p. 396–408.

[21] Wang Y, Tang X, Liu J, Pan G, Xiao R. 3D face recognition by local shape difference boosting. In: Proceedings of the European conference on computer vision, vol. 1. Marseille, France; 2008. p. 603–16.

[22] Colombo A, Cusano C, Schettini R. Gappy pca classification for occlusion tolerant 3D face detection. J Math Imaging Vision 2009;35(3):193–207.

[23] Perakis P, Passalis G, Theoharis T, Toderici G, Kakadiaris IA. Partial matching of interpose 3D facial data for face recognition. In: Proceedings of the international conference on biometrics: theory, applications, and systems. Washington, DC; 2009. p. 1–8.

[24] Passalis G, Perakis P, Theoharis T, Kakadiaris IA. Using facial symmetry to handle pose variations in real-world 3D face recognition. IEEE Trans Pattern Anal Mach Intell 2011;33(10):1938–51.

[25] Kakadiaris IA, Passalis G, Toderici G, Murtuza N, Lu Y, Karampatziakis N, Theoharis T. Three-dimensional face recognition in the presence of facial expressions: an annotated deformable approach. IEEE Trans Pattern Anal Mach Intell 2007;29(4):640–9.

[26] Chang KI, Bowyer KW, Flynn PJ. Multiple nose region matching for 3D face recognition under varying facial expression. IEEE Trans Pattern Anal Mach Intell 2006;28(6):1695–700.

[27] Faltemier TC, Bowyer KW, Flynn PJ. A region ensemble for 3D face recognition. IEEE Trans Inf Forensics Secur 2008;3(1):62–73.

[28] Alyüz N, Gökberk B, Akarun L. 3D face recognition system for expression and occlusion invariance. In: Proceedings of the IEEE international conference on biometrics: theory, applications, and systems. Washington, DC, USA; 2008. p. 1–7.

[29] Drira H, Ben Amor B, Daoudi M. A. Srivastava, Pose and expression-invariant 3D face recognition using elastic radial curves. In: Proceedings of the British machine vision conference. Aberystwyth, UK; 2010. p. 1–11.

[30] Gupta S, Markey MK, Bovik AC. Anthropometric 3D face recognition. Int J Comput Vision 2010;90(3):331–49.

[31] Mian AS, Bennamoun M, Owens R. Keypoint detection and local feature matching for textured 3D face recognition. Int J Comput Vision 2008;79(1):1–12.

[32] Lowe D. Distinctive image features from scale-invariant key points. Int J Comput Vision 2004;60(2):91–110.

[33] Mian AS, Bennamoun M, Owens R. On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. Int J Comput Vision 2010;89(2–3):348–61.

[34] Huang D, Zhang G, Ardabilian M, Wang Y, Chen L. 3D Face Recognition using distinctiveness enhanced facial representations and local feature hybrid matching. In: Proceedings of the IEEE international conference on biometrics: theory, applications and systems (BTAS'10). Washington DC, USA; 2010. p. 1–7.

[35] Huang D, Ardabilian M, Wang Y, Chen L. 3D face recognition using eLBP-based facial representation and local feature hybrid matching. IEEE Trans Inf Forensics Secur 2012;7(5):1551–64.

[36] Li H, Huang D, Lemaire P, Morvan J-M, Chen L. Expression robust 3D face recognition via mesh-based histograms of multiple order surface differential quantities. In: Proceedings of the IEEE international conference on image processing (ICIP'11); 2011. p. 3053–6.

[37] Claes P, Smeets D, Hermans J, Vandermeulen D, Suetens P. SHREC'11 track: robust fitting of statistical model. In: Proceedings of the eurographics workshop on 3D object retrieval. Llandudno, UK,;2011. p. 89–95.

[38] Veltkamp R, van Jole S, Drira H, Ben Amor B, Daoudi M, Li H, Chen L, Claes P, Smeets D, Hermans J, Vandermeulen D, Suetens P. SHREC'11 track: 3D face models retrieval. In: Proceedings of the eurographics workshop on 3D object retrieval. Llandudno, UK; 2011. p. 89–95.

[39] Maes C, Fabry T, Keustermans J, Smeets D, Suetens P, Vandermeulen D. Feature detection on 3D face surfaces for pose normalisation and recognition. In: Fourth IEEE international conference on biometrics: theory applications and systems (BTAS); 2010. p. 1–6.

[40] Tola E, Lepetit V, Fua P. A fast local descriptor for dense matching. In: Proceedings of the international conference on computer vision and pattern recognition. Anchorage, AK; 2008. p. 1–8.

[41] Zaharescu A, Boyer E, Varanasi K, Horaud R. Surface feature detection and description with applications to mesh matching. In: Proceedings of the IEEE international conference on computer vision and pattern recognition. Miami Beach, FL; 2009. p. 373–80.

[42] Zaharescu A, Boyer E, Horaud R. Keypoints and local descriptors of scalar functions on 2D manifolds. Int J Comput Vision 2012;100(1):78–98.

[43] Zuliani M, Kenney CS, Manjunath BS. The multiransac algorithm and its application to detect planar homographies. In: Proceedings of the IEEE international conference on image processing; 2005. p. 153–6.

[44] Yin L, Wei X, Sun Y, Wang J, Rosato M. A 3D facial expression database for facial behavior research. In: Proceedings of the IEEE international conference on automatic face and gesture recognition. Southampton, UK; 2006. p. 211–6.

[45] Bagdanov AD, Del Bimbo A, Masi I. The Florence 2D/3D hybrid face dataset. In: Proceedings of the joint ACM workshop on human gesture and behavior understanding (J-HGBU'11). Arizona, USA; 2011. p. 79–80.

[46] Smeets D, Keustermans J, Vandermeulen D, Suetens P. meshSIFT: Local surface features for 3D face recognition under expression variations and partial data. Comput Vision Image Understanding 2013;117(2):158–69.

[47] Boyer E, Bronstein AM, Bronstein MM, Bustos B, Darom T, Horaud R, Hotz I, Keller Y, Keustermans J, Kovnatsky A, Litman R, Reininghaus J, Sipiran I, Smeets D, Suetens P, Vandermeulen D, Zaharescu A, Zobel V. SHREC 2011: robust feature detection and description benchmark. In: Proceedings of the eurographics workshop on 3D object retrieval (3DOR 2011). Llandudno, UK; 2011.

[48] Salti S, Tombari F, Di Stefano L. Performance evaluation of 3D keypoint detectors. Int J Comput Vision 2013;102(2–3):198–220.

[49] Sipiran I, Bustos B. A robust 3D interest points detector based on Harris operator. In: Proceedings of the eurographics workshop on 3D object retrieval, eurographics association. Norrköping, Sweden; 2010. p. 7–14.

[50] Litman R, Bronstein AM, Bronstein MM. Diffusion-geometric maximally stable component detection in deformable shapes. Comput Graph 2011;35(3):549–60.

[51] Peyre G. Toolbox graph. In: MATLAB central file exchange select; 2009.

[52] Frome A, Huber D, Kolluri R, Bülow T, Malik J. Recognizing objects in range data using regional point descriptors. In: Proceedings of the European conference on computer vision, vol. 3. Prague, Czech Republic; 2004. p. 224–37.

[53] Tombari F, Salti S, Di Stefano L. Unique signature of histograms for local surface description. In: European conference on computer vision, vol. III. Heraklion, Crete, Greece; 2010. p. 347–60.

[54] Tombari F, Salti S, Di Stefano L. Unique shape context for 3D data description. In: Proceedings of the ACM workshop on 3D object retrieval. Firenze, Italy; 2010. p. 57–62.

[55] Ashbrook A, Fisher R, Robertson C, Werghi N. Finding surface correspondence for object recognition and registration using pairwise geometric histograms. In: Proceedings of the European conference on computer vision. Friburg, Germany; 1998. p. 674–86.

[56] Werghi N, Rahayem M, Kjellander J. An ordered topological representation of 3D triangular mesh facial surface: concept and applications. EURASIP J Adv Signal Process 2012;2012(144):1–20.

[57] Fischler MA, Bolles RC. Random sample consensus. Commun ACM 1981;24(6):381–95.

[58] Ekman P. Universals and cultural differences in facial expressions of emotion. In: Proceedings of the of the Nebraska symposium on motivation, vol. 19. Lincoln, NE; 1972. p. 207–83.

[59] Mpiperis I, Malassiotis S, Strintzis M. Bilinear models for 3-D face and facial expression recognition. IEEE Trans Inf Forensics Secur 2008;3(3):498–511.

[60] Ocegueda O, Passalis G, Theoharis T, Shah S, Kakadiaris I. UR3D-C: linear dimensionality reduction for efficient 3d face recognition. In: International joint conference on biometrics (IJCB'11). Washington DC, USA; 2011. p. 1–6.

[61] Kobbelt L, Bareuther T, Seidel H-P. Multiresolution shape deformations for meshes with dynamic vertex connectivity. Eurographics 2000;19(3):1–11.

Boosting 3D LBP-Based Face Recognition by Fusing Shape and Texture Descriptors on the Mesh

Naoufel Werghi, *Senior Member, IEEE*, Claudio Tortorici, Stefano Berretti, *Member, IEEE*, and Alberto Del Bimbo, *Senior Member, IEEE*

***Abstract*— In this paper, we present a novel approach for fusing shape and texture local binary patterns (LBPs) on a mesh for 3D face recognition. Using a recently proposed framework, we compute LBP directly on the face mesh surface, then we construct a grid of the regions on the facial surface that can accommodate global and partial descriptions. Compared with its depth-image counterpart, our approach is distinguished by the following features: 1) inherits the intrinsic advantages of mesh surface (e.g., preservation of the full geometry); 2) does not require normalization; and 3) can accommodate partial matching. In addition, it allows early level fusion of texture and shape modalities. Through experiments conducted on the BU-3DFE and Bosphorus databases, we assess different variants of our approach with regard to facial expressions and missing data, also in comparison to the state-of-the-art solutions.**

***Index Terms*— Mesh-LBP, feature and score fusion, 3D face recognition.**

I. Introduction

THE LAST decade has seen an extensive investigation of 3D face image usage for human identification. Adding to shape information the intrinsic features characterizing facial image, such as universal acceptance and non-invasiveness, 3D face image has emerged as promising modality addressing the limitations of its 2D counterpart, such as pose and luminance variation, while opening-up new horizons for enhancing the reliability of face-based identification systems. This trend has been further fueled by the advances in 3D scanning technology, which provides now 3D textured scans encompassing aligned shape and photometric data.

Since their introduction in the mid '90, Local Binary Patterns (LBP) [2] have been extensively used in 2D face description and representation, and rapidly have been extended to the 3D modality. 3D-LBP approaches advanced the state of the art, and proved to be competitive with other classes of methods. However, their application is hindered by the intrinsic limitations of the 2D image support. Indeed, most if not all 3D-LBP approaches operate on depth images, in which depth is mapped to a gray level via 2D projection. As such, depth images require normalization to accommodate with pose variation. Yet, they still remain vulnerable to self-occlusions (caused for instance by the nostrils).

To address these problems, we propose a novel LBP-based face representation that can be constructed over triangular mesh manifolds. This representation, which is based on the recently proposed mesh-LBP concept [1], preserves the full 3D geometry of the shape, thus relieving the recognition process from the need for pose normalization (i.e., since mesh-LBP descriptors are computed on the 3D mesh triangulation, they are intrinsically independent from the mesh orientation in the 3D space). In another hand, given the consensus on the advantageous aspects of multi-modal face recognition [3], LBP construction on the mesh allows boosting recognition by offering an elegant framework for fusing, over a mesh support, texture and shape information at data and feature level, in addition to score and decision level, noticeably. To the best of our knowledge, this work is the first one to propose texture and shape fusion for face recognition using LBP constructed on the mesh. We also point out that our contribution in this work focuses mostly on the aspect related to the face description and, as a matter of fact, we are employing a very basic minimum distance classifier in the recognition pipeline. In the remaining of this Section, we first summarize the works that are most related to our solution (Sect. I-A), then we outline the proposed approach and its main contributions (Sect. I-B).

A. Related Work

Many 3D face recognition approaches have been proposed in the literature, and going through all of them is out of the scope of this summary. Instead, in the following, we will focus on existing methods that are relevant for the proposed solution, which can be categorized according to three different aspects: *a)* Methods that use local representations of the face, and thus are capable of supporting partial face matching, as can occur in the case of expression variations or missing parts (many recent methods achive this goal relying on *fiducial points* of the face); *b)* Approaches featuring face recognition by extending the LBP framework to depth images and 3D modalities; *c) Multi-modal* solutions that fuse together the 3D geometry and the 2D photometric appearance of the face to improve recognition. A more general and comprehensive review of 3D face recognition can be found in [3]–[6].

Manuscript received August 20, 2015; revised November 27, 2015; accepted December 21, 2015. Date of publication January 6, 2016; date of current version February 24, 2016. This work was supported by the National Research Foundation, United Arab Emirates, under Grant UIRCA 2013-24877. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. P. C. Yuen.

N. Werghi and C. Tortorici are with the Electrical and Computer Engineering Department, Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: naoufel.werghi@kustar.ac.ae; claudio.tortorici@kustar.ac.ae).

S. Berretti and A. Del Bimbo are with the Department of Information Engineering, University of Florence, Florence 50139, Italy (e-mail: stefano.berretti@unifi.it; alberto.delbimbo@unifi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2016.2515505

1) Local Methods Based on Fiducial Points: At a very broad level, solutions for 3D face recognition can be grouped as *global*, performing face matching based on the whole face, and *local* that partition the face surface into regions and extract appropriate descriptors for each of them [7]. Methods in this latter category have recently gained an increasing credit, mainly thanks to their capability of natively supporting partial face match, as occurring in the case of scans with missing parts or occlusions (the case of facial expressions is often managed in a similar way, by excluding from the match the parts of the face that are most affected by expression variations). Among local approaches, effective results have been reported by methods that detect *fiducial points* of the face (being them either anthropometric landmarks, points of a predefined grid, or sparse keypoints), and compute local descriptors of surface patches centered at the fiducial points. One of the first approaches following this framework was proposed by Mian et al. [8], which designed a 3D keypoints detector and descriptor inspired by SIFT [9]. This detector/descriptor was used to perform 3D face recognition through a multi-modal 2D+3D approach that also used SIFT to index 2D images of the face. However, results reported for the method did not account for face scans with pose variations and missing parts. In [10] and [11], the framework of SIFT keypoints detector has been reformulated to operate on 3D face meshes by defining the mesh-SIFT detector and local descriptor. A scale-space analysis of the mesh is first performed through subsequent smoothing of the 3D geometry, then 3D keypoints are identified as the local extrema of the mean curvature extracted from the smoothed versions of the original mesh through the scales. Local descriptors are defined at the keypoints using nine local regions (arranged according to a daisy-like pattern), and computing for each of them a pair of histograms (the shape-index and the angle between surface normal descriptors are used). Effective local solutions based on fiducial points have been recently reported also in [12]–[14]. In [12], Lin et al. used mesh-SIFT to detect feature points on 3D face scans; Then, the quasi-daisy local shape descriptor [15] at each feature point was obtained using multiple order histograms of differential quantities extracted from the surface; Finally, these local descriptors were matched by computing their orientation angles. The same authors extended this work in [13], by boosting the keypoints matching with the Sparse Representation based Classifier (SRC) [16]. In [14], Berretti et al. used a similar paradigm by considering different varieties of histogram descriptors computed at mesh-DOG 3D keypoints [17]. The keypoints matching was also improved using the RANSAC algorithm.

2) LBP-Based Solutions: Since the seminal work of Ahonen et al. [18], [19], LBP-based solutions have shown their effectiveness in face recognition from 2D still images [20]. Inspired by these works, the idea of extending LBP to the 3D geometry of the face has been explored in several studies. Most, if not all, the LBP-based face recognition methods in the literature operate on depth images. This format allowed a straightforward application of the 2D-LBP operator as it was demonstrated in the pioneering work of Li et al. [21]. Later, Huang et al. [22], [23] proposed the multi-scale extended LBP (eLBP), which consists of several LBP codes in multiple layers accounting for the exact gray value differences between the central pixel and its neighbors. Sandbach et al. [24] introduced the local normal binary pattern (LNBP), which used the angle between normals at two points, rather than the depth value to obtain the local binary code. This novel LNBP concept has been adopted in subsequent works in different variants. Li et al. [25] extracted surface normals in 3D, then the values of the normal components along the direction of the three coordinate axes are interpreted as depth values, and LBP is computed on these depth maps reporting the values of the normal components. In a further extension, Sandbach et al. [26] constructed images of azimuthal projection distance. The azimuthal equidistant projection is able to project normals onto points in an Euclidean space according to the direction. Though the projected information is not the depth, depending on the normals of the 3D surface, 2D LBP are still computed on the projection images. The 3D-LBP method proposed in [27] computed the difference of the depth value or the angle between the normal of a central vertex and the eight neighboring vertices on a mesh. Using this descriptor, a region based representation of the face similar to the one developed in [19] for 2D face recognition is derived. This work includes the idea of using normals computed on the mesh, but the mesh requires an elaborated preprocessing in order to extract LBP constrained to the eight vertices near to a central one. Also, the circular ordering procedure of these vertices, necessary to perform LBP computation is not revealed. In addition, multi-resolution LBP is not supported, and the partitioning of the face into regions is defined based on a set of 48 landmarks manually annotated. More recently, Bayramoglu et al. [28] combined a central symmetric variant (CS-3DLBP) pattern, and a set of geometrical features in a decision-level fusion using a robust random forest classifier. This method operates on depth images and adopted also surface normal orientation as a shape function. All the aforementioned LBP-based methods, except [27], operate on depth images, and therefore when dealing with a mesh model as input have to convert it into a depth image via assiduous normalization procedures. This makes handling incomplete face scan resulting, for instance, from pose variation and occlusion, quite problematic for these methods. Finally, while the method of Tang et al. [27] constructs LBP patterns on the mesh, it requires intense mesh preprocessing and lacks the multi-resolution aspect of the original LBP.

3) Multi-Modal 2D-3D Solutions: Multi-modal methods try to combine multiple processing paths (typically in 2D and 3D) into a coherent architecture to solve critical aspects of individual methods. In [29], Chang et al. proposed applying PCA to face depth images and 2D face images separately, and then fusing the results together. In the work of Lu et al. [30], ICP registration of the 3D face models was combined with LDA applied to 2D face images to improve the robustness of 2D face matching in the presence of pose and illumination variations. Beumier et al. [31] extracted central and lateral profiles of the face and compared them in both 3D and 2D. In the approach of Hüsken et al. [32], landmark positions used to define the face regions were also detected on 2D texture images obtained with the 3D face scan. Mian et al. [33] assembled a fully automated system performing: pose correction, automatic region segmentation to account for local variations of the face geometry, quick filtering of distant faces using SIFT and 3D Spherical Face Representation, and matching of the remaining faces applying a modified ICP to a few regions of the face (eyes, forehead, and nose) that are less sensitive to face expressions. The similarity scores provided by the two matching engines were fused into a single similarity measure. An in-depth study of fusion strategies for 3D face recognition was carried out by Gökberk et al. [34] that discussed and compared various techniques for classifier combination, such as fixed rules, voting- and rank-based fusion schemes, by fusing several off-the-shelf 3D and 2D features. Soltana et al. [35] through extensive experimentation show that individual 2D and 3D features are far from being distinctive for discriminating human faces. They propose an adaptive score level fusion strategy for multi-modal 2D-3D face recognition. The strategy consists of an offline and an online weight learning process, which automatically selects the most relevant weights of all the scores for each probe face in each modality.

B. Contribution and Paper Organization

From the above analysis, it emerges that solutions locally describing the face around fiducial points can perform 3D face recognition in difficult conditions, thanks to their intrinsic capability of managing partial match. On another side, there is evidence that LBP is an effective descriptor of the face capable of capturing local information. Last, multi-modal solutions that fuse together shape and photometric information can be used to boost further the recognition. In light of these considerations, we propose in this work a method capable of supporting recognition in the presence of missing parts, occlusions and expressions. Our method encompasses the following stages: 1) Computation of LBP descriptors using both shape and photometric information of the face mesh surface; 2) Construction of a grid of points on the face surface to obtain an ordered set of regions (equivalent to blocks in the 2D case); 3) Computing a histogram at each region, then concatenating the regional histograms into a structure encoding either a global or partial description of the face; 4) Performing the face matching by exploiting different fusion modalities. Our work presents the following innovative aspects:

- We introduce an LBP-based face representation constructed over triangular mesh manifolds;
- Our method relieves the recognition process from face pose normalization, while preserving the full geometry of the facial shape;
- Operating on the mesh, with our approach the photometric appearance is processed directly attached to the mesh and not on a separated planar image as in other multimodal methods, thus allowing an early level-fusion of the texture and shape information;
- Our method uses a fixed set of fiducial points based on a sampling grid of the face. The points of the grid are obtained according to a predefined arrangement with respect to three reference facial landmarks. This avoids the need for elaborated processing required by keypoints detectors.

The results obtained on the BU-3DFE and Bosphorus datasets show the proposed method competes, and in some cases overcomes, the state of the art solutions.

The rest of the paper is organized as follows: In Sect. II, we give an overview on the mesh-LBP concept, focussing on the descriptor computation and its properties; In Sect. III, we present our face representation based on mesh-LBP; The fusion modalities used to combine geometric and photometric descriptors attached to the mesh are discussed in Sect. IV; Experimental evaluation in comparison to state of the art methods with results on two datasets is reported in Sect. V; Finally, we discuss the main positive aspects of our framework together with its current limitations in Sect. VI, where we also draw possible directions for future work.

II. LBP Descriptors on the Mesh

LBP construction on triangular mesh manifolds is a recent concept introduced by Werghi et al. [1], [36]. Before describing it, let us briefly remind about the standard LBP construction. In its simplest form, an LBP is an 8-bit binary code obtained by comparing a pixel's value (e.g., gray level, depth) with each pixel's value in its 3×3 neighbour. The outcome of this comparison is 1 if the difference between the central pixel's value and its neighbour pixel's counterpart is less or equal than a certain threshold, and 0 otherwise. The so obtained local description can be refined and extended at different scales by adopting circular neighbourhoods at different radii and using pixel sub-sampling.

Werghi et al. [36] elegantly extended the LBP concept to 2D-mesh manifolds by proposing a simple yet efficient technique for constructing sequences of facets ordered in a circular fashion around a central facet. The principle of the approach consists in categorizing the facets on the contour defined by a central facet's edges in two categories, namely, the $Fout$ facet and the $Fgap$ facets. An $Fout$ facet (respectively, an $Fgap$ facet) shares an edge (respectively, a single vertex) with a central facet (referred by f_c in Fig. 1).

Starting with three—clockwise or anticlockwise—ordered $Fout$ facets ($fout_1$, $fout_2$, and $fout_3$ in Fig. 1), the construction algorithm iteratively extracts the $Fgap$ facets located between each pair of consecutive $Fout$ facets following the same order in which the $Fout$ facets have been initially arranged, and closing the loop at the pair composed by the last $Fout$ facet (the third one) and the first one. The outcome of this procedure is a ring of ordered facets arranged clockwise or anticlockwise around the central facet. From this ring, a new sequence of ordered $Fout$ facets located on the ring's outer-contour can be extracted, thus allowing the ring construction procedure to be iterated, and to generate a sequence of concentric rings around the central facet (see the illustrations on the bottom of Fig. 1).

Algorithms 1 and 2 summarize the computation of ordered rings of facets.

The so obtained structure of ordered and concentric rings around a central facet forms an adequate support for computing LBP operators (referred as mesh-LBP in [36]) at different

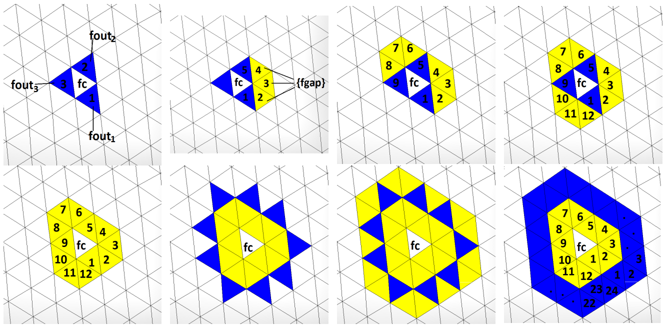


Fig. 1. Ordered ring construction: From the initial $Fout$ facets formed by the three ordered facets $fout_1$, $fout_2$, and $fout_3$ that are adjacent to the central facet f_c, a sequence of $Fgap$ facets located between each pair $\langle fout_1, fout_2\rangle$, $\langle fout_2, fout_3\rangle$, and $\langle fout_3, fout_1\rangle$ are extracted. The $Fgap$ facets have exactly one vertex on the initial 3-edge contour of the central facet f_c, and they are dubbed so because they look like filling the gap between the $Fout$ facets. This procedure produces a ring of facets ordered in a circular fashion around the central facet f_c. By iterating this procedure, using as new set of $Fout$ facets the sequence of facets that share an edge on the outer contour of the current ring, a sequence of rings of ordered facets can be generated.

Algorithm 1 Bridge

```
Require: $fout_i$, $fout_{i+1}$ two consecutive $Fout$ facets sharing
  a vertex; $fin_i$ facet that shares an edge with $fout_i$
Ensure: $Fgap_i$ set of consecutive $fgap$ facets bridging the
  gap between $fout_i$ and $fout_{i+1}$

  procedure BRIDGE($fout_i, fout_{i+1}, fin_i$)
    $Fgap_i = [\ ]$
    $v \leftarrow$ vertex shared by $\langle fout_i,\ fout_{i+1}\rangle$
    $gf \leftarrow$ facet adjacent to $fout_i$, different from $fin_i$
        and containing $v$
    $prev \leftarrow fout_i$
    while $gf \neq fout_{i+1}$ do
      append $gf$ to $Fgap_i$
      $new\_gf \leftarrow$ facet adjacent to $gf$, different from $prev$
          and containing $v$
      $prev \leftarrow gf$
      $gf \leftarrow new\_gf$
    end while
  end procedure
```

Algorithm 2 GetRing

```
Require: $Fout$, set of $n$ ordered facets, $fout_1$, $fout_2, \ldots,$
  $fout_n$, lying on a convex contour; $Fin$, set of $n$ ordered
  facets, $fin_1, fin_2, \ldots, fin_n$, one-to-one adjacent to the
  $Fout$ facets and located inside the region delimited by
  the convex contour (depending on the contour, $Fin$ might
  include duplicates)
Ensure: $Ring$, ring of ordered facets

  procedure GETRING($Fout, Fin$)
    $Ring = [\ ]$
    for all $\langle fout_i, fout_{i\%n+1}\rangle$, $i \leftarrow 1, \ldots, n$ do
      append $fout_i$ to $Ring$
      $Fgap_i \leftarrow$ BRIDGE($fout_i, fout_{i\%n+1}, fin_i$)
      append $Fgap_i$ to $Ring$
    end for
  end procedure
```

radial and azimuthal resolutions, while preserving the simplicity of the original LBP. Let $h(f)$ be a scalar function defined on the mesh, incarnating either a geometric (e.g., curvature) or photometric (e.g., color or gray level) information. The mesh-LBP operator at the facet f_c is defined as follows [36]:

$$meshLBP_m^r(f_c) = \sum_{k=0}^{m-1} s(h(f_k^r) - h(f_c)) \cdot \alpha(k),$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0, \end{cases} \tag{1}$$

where r is the ring number, and m is the number of facets uniformly spaced on the ring. The parameters r and m control, respectively, the radial resolution and the azimuthal quantization. In practice, in our implementation, we used a number of rings going from 1 to 7, with 12 facets per ring for computing mesh-LBP descriptors. This choice reflects the fact we have 12 facets in the first ring (regular mesh), and we keep this number of samples in any subsequent ring of the facet's support. The discrete function $\alpha(k)$ is introduced for the purpose of deriving different LBP variants. In this work, we will consider two variants of $\alpha(k)$: for $\alpha(k) = 2^k$, we obtain the mesh counterpart of the basic LBP operator firstly suggested by Ojala et al. [2]; for $\alpha(k) = 1$, we obtain the sum of the digits equal to 1 composing the binary pattern. In the experiments, we will refer to these two functions by α_2 and α_1, respectively. For the discrete surface function $h(f)$, in this work we experimented the *mean curvature* (H), the *curvedness* (C), the *Gaussian curvature* (K) and the *shape index* (SI), as shape descriptors, plus the *gray level* value (GL) as photometric characteristic of the facets.

With reference to the computation of mesh-LBP, it is relevant to note that the facets of the first ring can be ordered in three different ways, depending on which of the three $Fout$ facets adjacent to the central facet f_c is considered as the initial one. To solve this ambiguity, the closest facet to the center of mass of the f_c's neighbourhood is elected as the initial facet of the ring. Subsequent rings inherit the ordering of the facets from that established for the first ring. It can be also observed that, by construction, patterns computed with the α_1 function do not depend on the choice of the initial facet of the ring (i.e., the pattern value is determined just from the number of digits set to 1, rather than from their position as instead occurs for α_2). In the ideal case of a regular mesh, the number of facets ν at ring i is computed according to the arithmetic progression $\nu_{i+1} = \nu_i + 12$ $(\nu_0 = 0)$. In the real case, to cope with mesh tessellation irregularities as produced by 3D scanner acquisitions, the scalar function $h(f)$ is interpolated and subsampled across each ring, allowing thus to maintain a constant azimuthal quantization. The authors in [36] showed that this technique copes to a large extent with mesh irregularity.

A. Constructing and Comparing Mesh-LBP Descriptors

As for their 2D counterpart, the outputs of mesh-LBP operators of Eq. (1) computed across a mesh surface are not usually directly used in shape matching, but rather accumulated into a discrete histogram constructed over a given neighborhood. The size of the histogram depends on the radial and azimuthal parameters r and m, as well as on the discrete function α. For example, with $r = 7$ and $m = 12$ we will obtain the histogram encompassing 7×13 and 7×4096 bins, for the α_1 and α_2, respectively. In fact, in the first case, 13 different values of the patterns are possible, being them coincident with the possible number of digits set to 1 in the binary code (i.e., the number of bit from 0 to 12 that to 1, which is also equal to the sum of the bit values); in the case of α_2, each digit in the pattern is weighted according to its position, so that 4096 different binary codes are possible (i.e., from 0 to 4095). The obtained histogram bins can be arranged in a 1-D or 2-D accumulator, and compared using χ^2 distance:

$$d(H_1, H_2) = \frac{1}{2} \cdot \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}, \qquad (2)$$

where H_1 and H_2 are two normalized histogram descriptors. Good results have been obtained also using the *cosine* distance, especially for the α_1 variant.

Considering histograms obtained with the α_1 and α_2 functions, it is evident the different size of the respective descriptors. In particular, with an azimuthal quantization $m = 12$, 4096 mesh-LBP patterns are possible for α_2, compared to the 13 different patterns for α_1. This aspect has been investigated in [36], showing that the majority of the α_2 patterns have a number of 0-1 transitions below 4. These patterns have been called “uniform” following a similar property noticed first by Ojala et al. [37] for 2D-LBP (in that case, for patterns of eight bits, the uniformity was assumed for a number of 0-1 transitions not greater than 2).

In this work, we re-investigated the presence of uniform patterns on face scan samples from the Bosphorus database, using the *mean curvature*, *curvedness* and the *gray level* as scalar functions. Again, we found that the mesh-LBP with a number of 0-1 transitions less or equal than 4 form more than 95% of the total number of patterns across seven rings. The detailed statistics are reported in Fig. 2, whereby we can see the frequencies of the different 0-1 transitions in the mesh-LBP patterns and the percentage of the transitions below or equal to 4, across all the rings. In the bottom of Fig. 2, we also visualize the facets corresponding to non-uniform patterns. It is evident, there are a few non-uniform patterns, and they are located mostly in non-rigid parts of the face, which change with facial expressions. These results seem to suggest that considering uniform patterns is sufficient. Thus, considering four 0-1 transitions as the threshold for uniform patterns, it results in exactly 1124 uniform patterns against 2972 non-uniform ones. Following the same partition scheme of [37], where all the non-uniform patterns are grouped into a single label, whereas a separate label is assigned to each non-uniform pattern, the number of histogram bins (or classes) is reduced to 1125 for our mesh-LBP. We will adopt this partition in the rest of the paper for the α_2 function. For α_1, the distinction into uniform/non-uniform patterns does not make too much sense, since in this case the sum of the number of digits set to 1 in the binary code is computed, rather than the binary value given by the polynomial expansion of the digits, as for α_2. This results in only 13 possible different patterns.

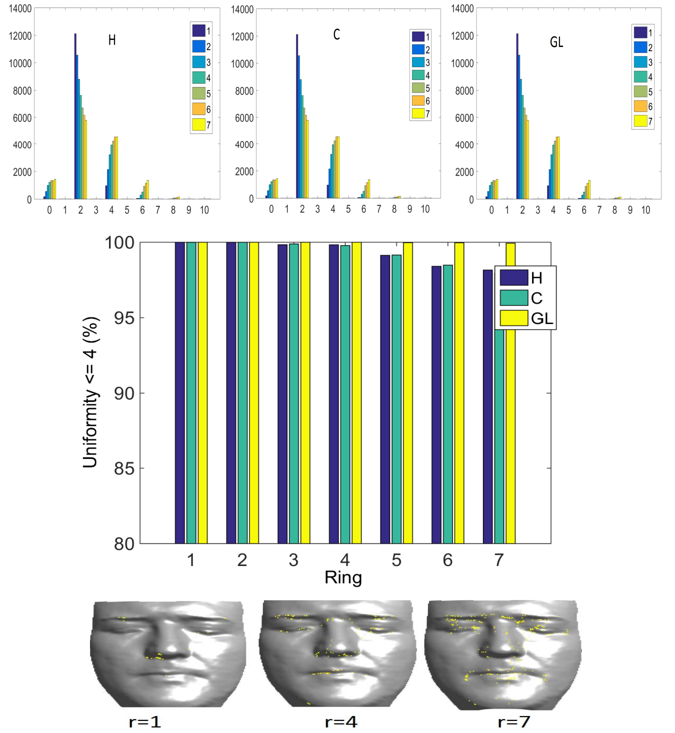


Fig. 2. Top: distribution of the number of 0-1 transitions in the mesh-LBP patterns using α_2 and the scalar functions *mean curvature* (H), *curvedness* (C) and *gray level* (GL). The mesh-LBP patterns have been computed for the seven radial resolutions $r = 1:7$ (i.e., seven rings), and for an azimuthal resolution $m = 12$ across all the rings. Note that number of odd transitions is always zero because what is counted actually is both the 0-1 and 1-0 transitions, and considering a circular arrangement of the binary digits. Middle: Percentage of the mesh-LBP patterns, in the same variants, showing a number of 0-1 transitions below or equal to four. Bottom: Facets on an example face scan having a non-uniform pattern obtained with *mean curvature*, for the radial resolutions $r = 1$, 4 and 7.

To have a visual insight on the capacity of mesh-LBP to capture and discriminate local shape information, we considered first five fundamental shapes, namely, *valley*, *ridge*, *pit*, *peak*, and *saddle* (see Fig. 3(a)), and computed their mesh-LBP histograms using the *mean curvature* as scalar surface function. Results are reported in Fig. 3(b)-(c) for the α_1 and α_2 (adopting the uniform/non-uniform pattern partition) variants, respectively. We can notice that the pairs *valley-ridge*, *pit-peak* show similar histograms, because of their symmetry relationships, while they are quite distinguishable from each other and from their *saddle* counterpart. For the facial shape, we report in Fig. 4 representative mesh-LBP variants computed with the geometric and photometric functions H and GL, at seven different radial resolutions ($r = 1, \ldots, 7$). We can easily observe, across these different variants, patterns reflecting facial features. Also we notice that, as the radial resolution increases, these patterns exhibit a fine to coarse evolution common to multi-resolution operators. In Fig. 5, we extend this analysis to the case of within and between class variation of the mesh-LBP descriptors, by reporting examples computed on two sets of four instances corresponding to a same and

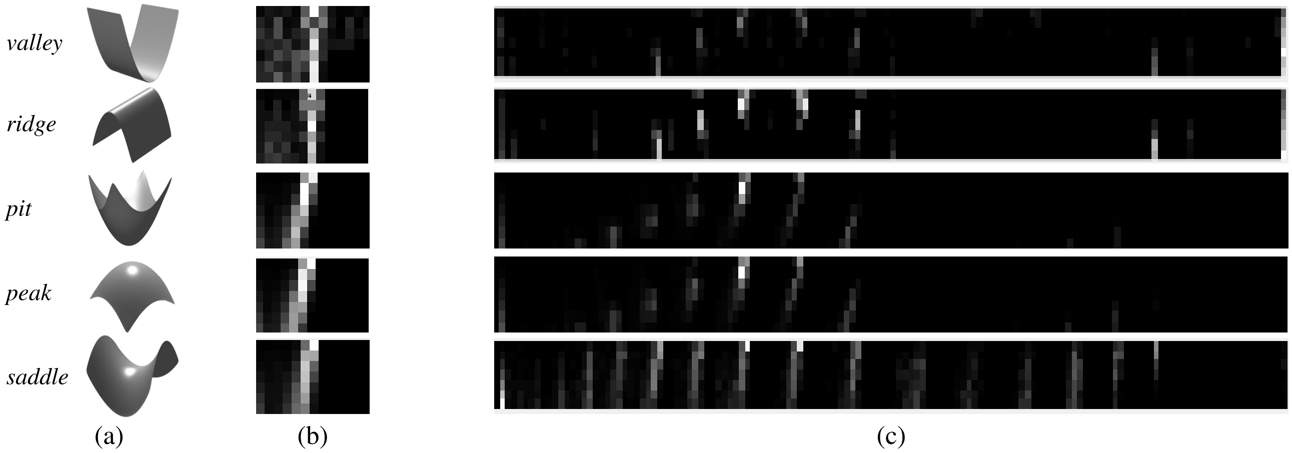


Fig. 3. Fundamental shapes (a) and their mesh-LBP histograms obtained using the mean curvature descriptor with α_1 (b), and α_2 (c).

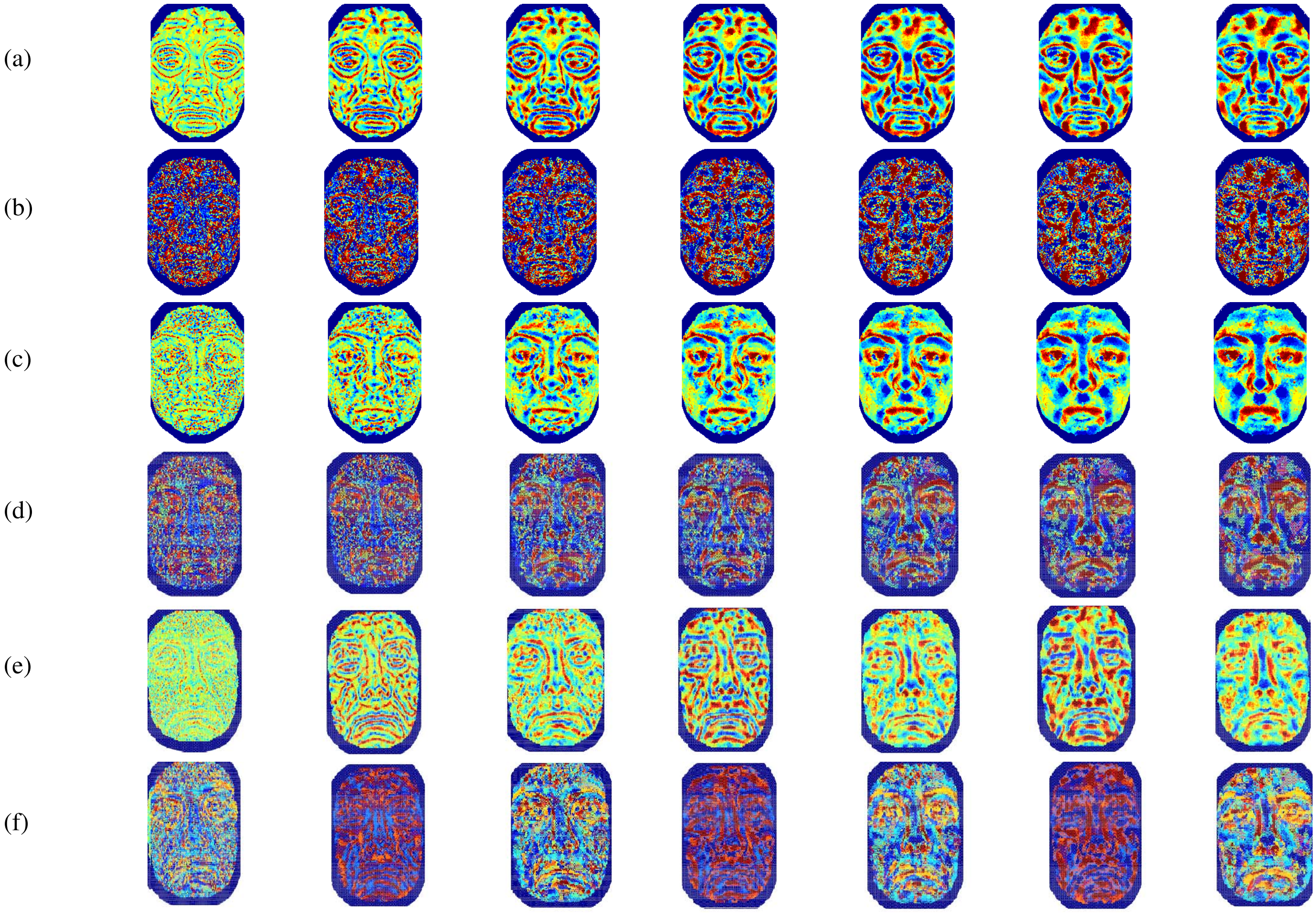


Fig. 4. Examples of mesh-LBP computed using the *mean curvature* (H), and the *gray level* (GL) in combination with α_1 and α_2: (a) $\langle H, \alpha_1 \rangle$; (b) $\langle H, \alpha_2 \rangle$; (c) $\langle GL, \alpha_1 \rangle$; (d) $\langle GL, \alpha_2 \rangle$; (e) $FF3\langle (H, GL), \alpha_1 \rangle$; (f) $FF3\langle (H, GL), \alpha_2 \rangle$. From left to right, the radial resolution r changes from 1 to 7 in each case.

different subjects. We can easily appreciate the stability of the patterns across the sibling instances as opposed to the neat variability observed across the non-related ones.

From Figs. 3, 4 and 5, both α_1 and α_2 categories exhibit great potential to be employed in facial surface description. While rotation invariance and low size properties of α_1 give it more favor than α_2, there are no prior indicators that can objectively indicate whether it can equate or outperform α_2 in terms of discriminating ability. The accentuated level of details and granularity exhibited by the examples of α_2 mesh-LBP descriptors displayed in Fig. 4 and 5 seem rather to indicate the opposite.

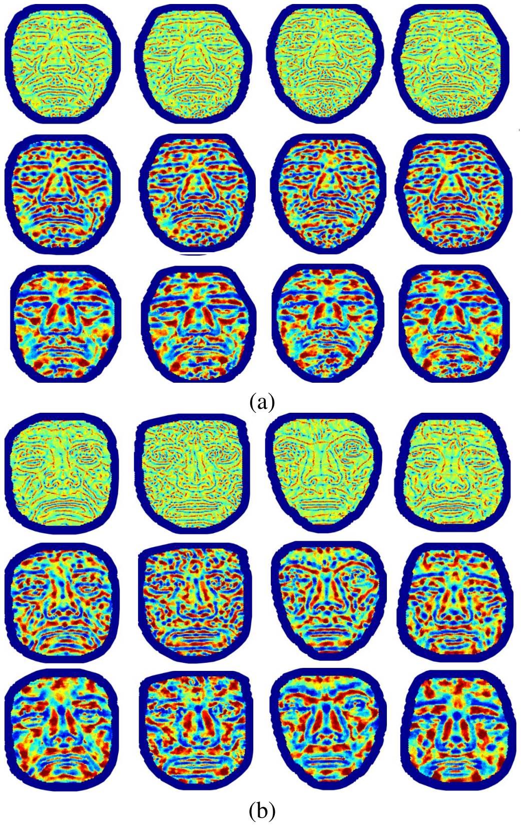


Fig. 5. Examples of mesh-LBP descriptors for four instances of the same subject (a), versus their counterparts related to four different subjects (b). The mesh-LBP used here is $\langle H, \alpha_1 \rangle$ computed at three radial resolutions $r = 1$, 3 and 7, from top to bottom of (a) and (b).

III. Face Representation

The previous analysis indicates that the mesh-LBP has some useful properties that make it attractive for capturing the shape and photometric information of a 3D surface. In order to exploit such potential for deriving a suitable face representation, we have taken inspiration from 2D face recognition methods that use standard LBP, and 3D methods based on fiducial points of the face that showed their appropriateness in supporting face recognition in the presence of facial expressions, occlusions and missing parts of the face. In particular, in the standard LBP-based face representation [19], a 2D face image is divided into a grid of rectangular blocks, then histograms of LBP descriptors are extracted from each block and concatenated afterwards to form a global description of the face. In so doing, image partitioning is performed easily due to the natural ordering of image pixels.

To extend this scheme to the face manifold, we need firs t to partition the facial surface into a grid of regions (the counterpart of the blocks in the 2D-LBP), compute their corresponding histograms, and then group them into a single structure. Since partitioning of the 2D mesh manifold is not straightforward, we rely on the idea of extracting a grid of fiducial points of the face with predefined position, and then use their neighborhood regions as local supports for computing mesh-LBP. In more details, this is performed with the following steps. First, the plane formed by the nose tip and the two eyes inner-corner landmark points is initially computed (see Fig. 6(a), left). We used these three landmarks as they are the most accurate detectable landmarks on the face, and they are also quite robust to facial expressions. From these landmarks we derive, via simple geometric calculation, an ordered and regularly spaced set of points on that plane (see Fig. 6(a), middle). Afterwards, the plane is tilted slightly, by a constant amount, to make it more aligned with the face orientation, and then we project this set of points on the face surface, along the plane's normal direction. The outcome of this procedure is an ordered grid of points, which defines an atlas for the facial regions that will divide the facial surface (see Fig. 6(a), right). To account for the effects of facial expressions, we segmented the grid points into three bands, dubbed *top* (T), *middle* (M) and *bottom* (B). The TM option allows us to neutralize to some extent the shape changes manifesting at the lower part of the face, and caused by the mouth in particular. The TMB and the TM grids contain 35 and 26 points, respectively. The three TMB bands are shown on the left of Fig. 6(b), while the points comprised by the TM bands only are shown on the right.

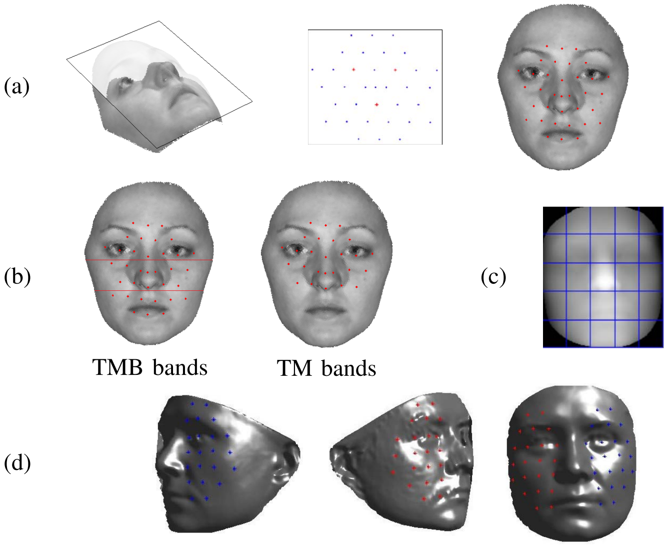


Fig. 6. (a) Construction of the face grid on the mesh; (b) On the left scan, all the grid points are shown and partitioned into three bands, namely, top (T), middle (M) and bottom (B), whereas on the right scan only the points in the top and middle bands (TM) are shown; (c) Grid partition of a depth image as used for the LBP method applied to depth images (3D-LBP); (d) Construction of the partial grid on a two rotated probe scans and a gallery scan.

For a yaw rotated pose resulting on a partial scan that does not allow the extraction of one of the two eyes inner-corner landmarks, we adopted a lateral grid, constructed upon the plane defined by one eye inner-corner, an eye outer-corner and the nose ridge. The grid covers one side of the face and contains 22 points. For the gallery scans, the TMB grid and both the left and right lateral grids are constructed (see Fig. 6(d)). Figure 6(c) instead, shows the partitioning of a depth image into a grid of 5 × 5 blocs, which is used to compare our method with the 3D-LBP counterpart operating on depth images, as detailed in Sect. V-B.

Once the grid of points has been defined, we extract a neighbourhood of facets around each point of the grid. Each neighbourhood can be defined by the set of facets confined

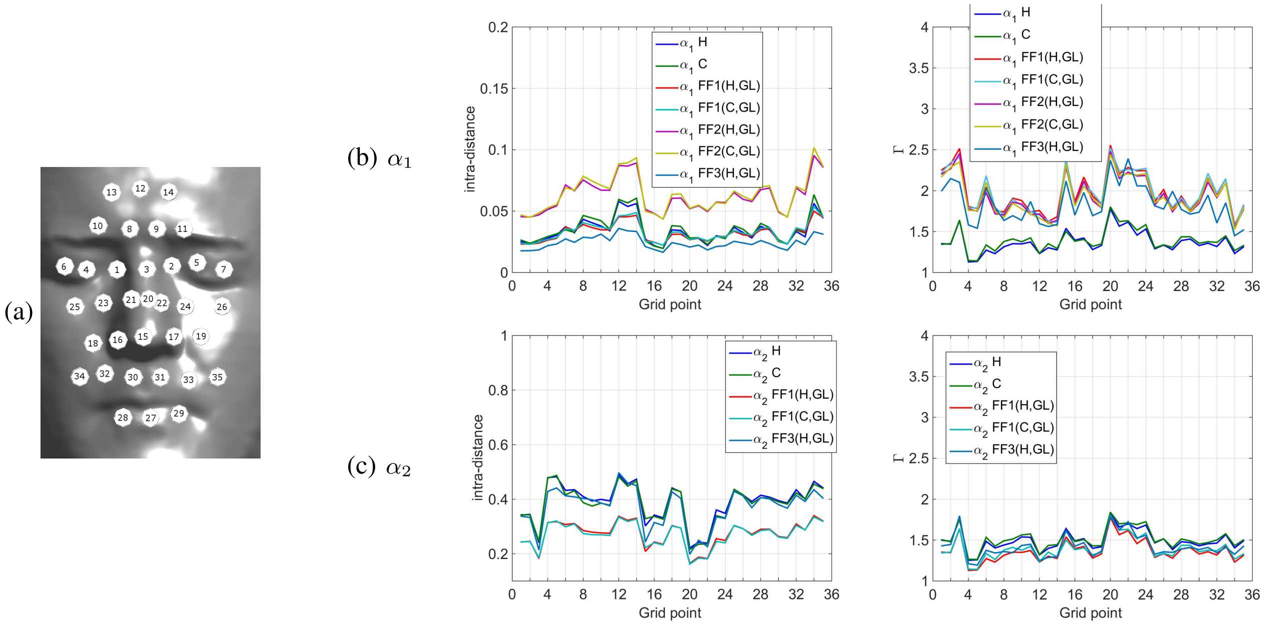


Fig. 7. (a) The numbered grid points; (b) and (c) report the *intra-class* distance and the criterion Γ computed at each grid point for the α_1 and α_2 mesh-LBP variants, respectively.

within a geodesic disc or a sphere, centered at a grid point. Then, we compute the multi-resolution mesh-LBP descriptor using Eq. (1) at each grid point region, considering both shape-valued (i.e., H, K, C, SI) and texture-valued (i.e., GL) functions. In the final step, the histograms of these descriptors are computed and integrated into a single histogram describing either the whole face or part of it (see Fig. 9(a)).

As primary assessment of the repeatability and the discrimination capacity of the different grid points in face matching we computed, for each grid point, the *inter-class* distance and the *intra-class* distance of the corresponding histogram. These two quantities have been obtained from, respectively, 35 pairs of scans, each corresponding to the same subject, and 35 scans corresponding to different subjects. Here, we adopted the *intra-class* distance and the ratio Γ=*inter-class distance/intra-class distance* as *repeatability* and *discrimination* indicators, respectively. Figure 7(b) and Fig. 7(c) depict the plot of these two indicators for each point of the grid (numbered according to Fig. 7(a)), and for the α_1 and α_2 mesh-LBP variants, respectively. Each plot compares a group of different descriptors including single and fusion variants (these will be described in Sect. IV). We can notice that the repeatability indicator shows virtually the same pattern across the different histogram descriptors. The best repeatability (i.e., lowest value) is observed at grid points around the nose and inner-eye corners (grid points {1, 2, 15, 16, 17, 22}). A similar behaviour is observed for the criterion Γ, whereby the grid points {1, 2, 3, 15, 20, 22} exhibit the most discriminative histograms (note that in this case the maximum of the curves correspond to the most discriminative points).

When we examined the distributions of the *intra-class* and the *inter-class* distances across the different grid points, we found that those in the α_2 variants exhibit more compact and separated distributions when compared to their α_1 counterparts. Figure 8 depicts some distribution examples illustrating this aspect. This suggested us that the α_2 variants have a higher discrimination, superior than α_1, as it will be confirmed in the experiments.

IV. Fusion Schemes

As a contribution of the proposed face representation, we propose the fusion of shape and photometric descriptors computed on the mesh. We further emphasize that the photometric channel is elaborated on the mesh as gray level attached to the triangles. No information is extracted from the 2D domain of gray (or depth) images of the face, but all the information is directly processed on the mesh manifold domain. Therefore, rather than being a multi-modal solution, the proposed approach can be regarded as a particular case of 3D methods, where the gray level plays an interchangeable role with standard shape surface descriptors.

In biometry applications, there are four levels of fusion considered, namely, *data*, *feature*, *score*, and *decision* [38]. As mentioned by Al-Osaimi et al. [5], it is believed that low-level fusion (data and feature) performs better than its higher level counterparts (score and decision) [39]. Looking at the spectrum of region methods fusing texture and 3D shape modalities, we found much concentration in the score-level category [21], [29], [33], [40], [41], as compared to the feature-level [8], [21], [42]. The work of Li et al. [21] in particular, fused LBP features derived from depth and texture image.

In our approach, we have investigated a score-level fusion and three variants of feature-level fusion. We have chosen the sum rule for the score-level, as it has been proven to be the optimal one [43]. In the first variant of the feature-level fusion, we concatenate the two mesh-LBP regional histograms, corresponding to the shape and the texture functions. For example, considering an azimuthal quantization $m = 12$ and α_1,

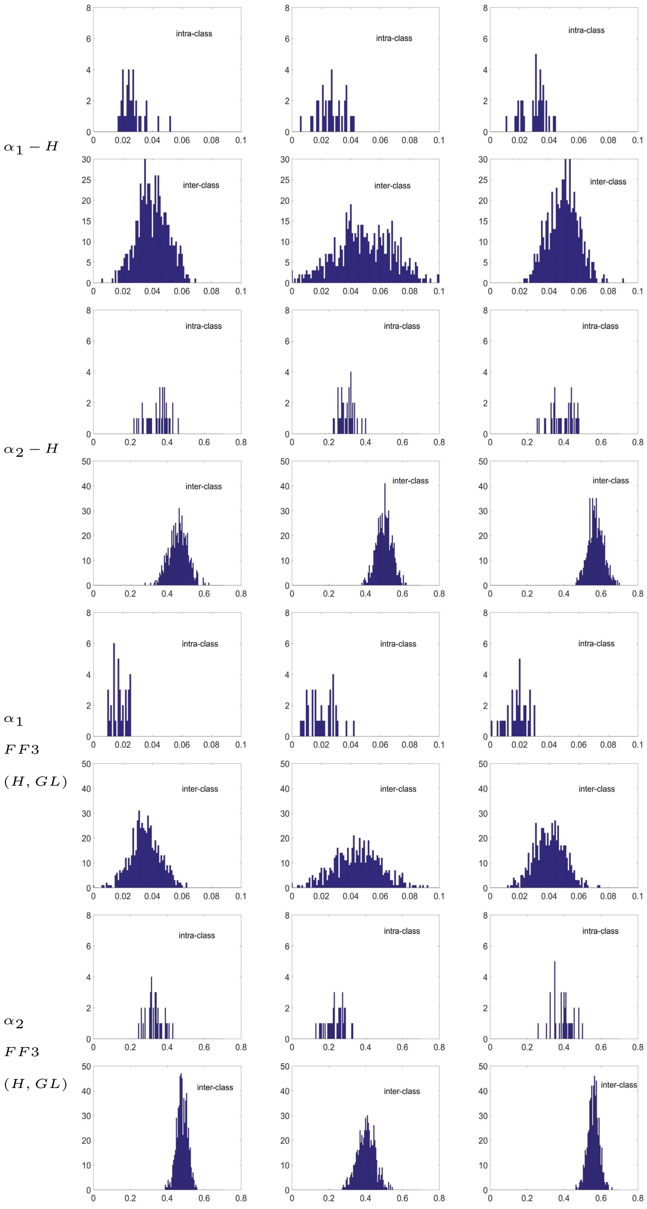


Fig. 8. Examples of *intra-class* and *inter-class* distribution computed at grid points at the nose tip (leftmost plots), right inner eye corner (middle plots), and mouth area (grid point 29 in Fig. 7, rightmost plots), for four different mesh-LBP variants. We notice that α_2 distributions exhibit more separation and compactness than their α_1 counterparts. The number of inter-class looks larger than its intra-class counterparts, as it encompasses all the pair combinations in the 35 subjects ($34 \times 35/2$).

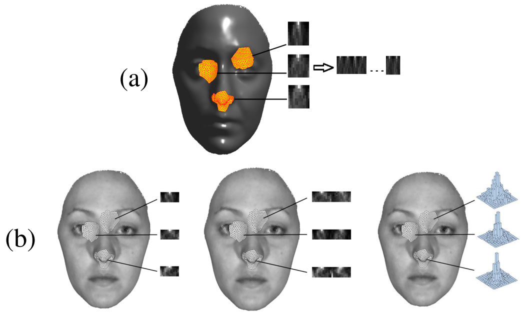


Fig. 9. (a) Global histogram construction: Region histograms are computed and then concatenated into a global histogram; (b) Examples of regional histogram variants obtained with $m = 12$ and $r = 7$ and α_1: (left) A 7×13 unimodal histogram corresponding to a shape function; (middle) A 7×26 histogram obtained by concatenating two 7×13 histograms corresponding to a shape function and a photometric function (gray level). This corresponds to the first variant of feature-level fusion ($FF1$); (right) A 2D section of a $7 \times 13 \times 13$ histogram obtained with a shape function and a photometric function. This is the second variant we used of feature-level fusion ($FF2$).

we obtain a 13-bins histogram for each function, thus leading to a one-dimensional 26-bins histogram for each radial resolution r, that is a $r \times 26$ histogram. In the second feature-level fusion variant, we used a 2-D accumulator that counts for the co-occurrences of the mesh-LBP corresponding to the shape and the texture functions. For the same aforementioned parameters' values, we obtain an $r \times 13 \times 13$ histogram (Fig. 9(b) depicts some examples). In the third variant, the fusion is performed at the LBP pattern level, rather than the histogram level, as for the first two. Here, the mesh-LBP pattern is constructed by interleaving digits from the shape mesh-LBP with a texture mesh-LBP. So, for an azimuthal quantification $m = 12$, the mesh-LBP pattern sequence is $b^s_1\, b^t_2\, b^s_3\, b^t_4\, b^s_5\, b^t_6\, b^s_7\, b^t_8\, b^s_9\, b^t_{10}\, b^s_{11}\, b^t_{12}$. The last variant has the advantage to keep the related histogram to the same size than its mono-feature counterpart. In the rest of the paper, we will refer to these first, second and third feature-level fusion variants by $FF1$, $FF2$, and $FF3$, respectively, whereas the score-level fusion will be referred by SF.

V. Experiments

We conducted a series of experiments aiming at studying the behavior and performance of our fusion framework with respect to facial expressions, missing face data resulting from pose variation and occlusion, and the extent it improves the recognition over the classic fusion performed on the depth image. Our framework is assessed in comparison with the best methods in the literature, adopting similar experimental settings.

A. BU-3DFE Database

A first series of experiments was conducted with the BU-3DFE database from Binghamton University [44]. This database contains scans of 56 males and 44 females, acquired in a neutral plus six different expressions (*anger*, *disgust*, *fear*, *happiness*, *sadness*, and *surprise*). Apart of the neutral expression, all the other facial expressions have been acquired at four levels of intensity. This combination results in a total of 2500 scans. We considered as gallery and probe the sets of neutral scans and the expression scans, respectively. Scans in this database contain both texture and shape data. Figure 10 depicts samples of the 3D facial expression instances, and a 2D image used for texture mapping in that database. The image encompasses two face sides acquired from the two stereo pods composing the face scanner used in the data collection.

The purpose of using the BU-3DFE is to assess the performance of our method, in particular our fusion schemes, with respect to facial expressions. On this dataset, we set the radial resolution r and the azimuthal quantization m used in computing mesh-LBP equal to 7 and 12, respectively.

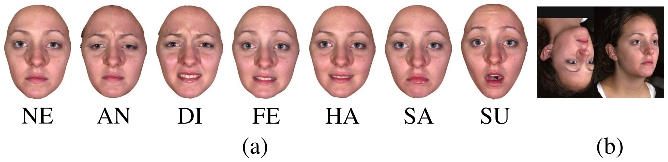


Fig. 10. BU-3DFE: (a) 3D face scans (with texture) of a sample subject showing, from left, the *neutral*, *anger*, *disgust*, *fear*, *happy*, *sad*, and *surprise* expressions (the level-1 of intensity is shown in each case); (b) The appearance image acquired by the scanner with two 45° side views of the face.

TABLE I

BU-3DFE: Rank-1 Recognition Accuracy (in Percentage) Obtained With Different Variants of Our Method for Level-1 Expression Intensity

		TMB		**TM**	
		cos	χ^2	cos	χ^2
α_1	H	90.61	88.00	92.52	89.57
	C	89.57	87.13	90.43	89.04
	SI	82.43	80.00	82.96	80.00
	GL	92.35	91.48	93.22	93.22
	$FF1$ H	95.65	95.13	**96.70**	95.65
	$FF1$ C	**96.35**	94.96	96.35	96.35
	$FF1$ SI	94.78	93.91	96.17	94.96
	$FF2$ H	96.17	95.13	**96.70**	95.65
	$FF2$ C	95.30	95.30	96.00	96.00
	$FF2$ SI	95.65	93.74	96.17	94.96
	SF H	96.00	95.13	96.00	95.13
	SF C	**96.35**	94.96	96.35	94.96
	SF SI	95.48	93.91	96.17	94.96
α_2	H	82.96	92.70	87.83	95.48
	C	85.22	93.57	89.39	95.13
	SI	70.61	82.96	66.26	78.96
	GL	75.93	90.09	78.61	92.35
	$FF1$ H	81.22	96.17	85.22	97.39
	$FF1$ C	80.35	**96.35**	84.35	**97.74**
	$FF1$ SI	78.96	94.96	83.13	95.13
	SF H	89.22	96.17	89.22	96.17
	SF C	88.87	**96.35**	88.87	96.35
	SF SI	86.61	94.96	86.43	95.13

The choice of 12 for m is justified by the fact that given a generic central facet, the number of facets in its first ring is always equal to 12 for regular meshes, regardless of the resolution, as demonstrated in [36]. Choosing this value allows then to account for all facets in the first ring. This number is used for the subsequent rings, so as to have patterns taking values in the same range. The number of rings r is related to the resolution of the mesh. The rationale behind the choice of r is to cover an area around a point of the sampling grid wide enough to capture local surface information. With the mesh of the BU-3DFE we found that $r = 7$ covers about $7mm$ around the point making a good compromise between computation efficiency and effectiveness of the description.

To account for the effects of facial expressions, we considered the grid points partition into three bands, dubbed *top* (T), *middle* (M) and *bottom* (B), as introduced in Sect. III. Then, we tested our recognition approach considering the full grid (TMB) and the top and middle bands (TM) only (see Fig. 6(b)). The TM option allows us to neutralize to some extent the shape changes manifesting at the lower part of the face, and caused by the mouth in particular. The TMB and the TM grids contain 35 and 26 points, respectively. For the choice of the local descriptors we tested, in a preliminary experimentation, a variety of descriptors that include the *mean* (H) and the *Gaussian* (K) curvatures, the *curvedness* (C), and the *shape index* (SI), in combination with the α_1 and α_2 functions. We found that the H and C descriptors perform best than the rest, so we will report results related to these descriptors, mainly.

In the first experiment, we considered two grid configurations, namely, the full grid encompassing the top, middle and bottom band (TMB), and the partial grid including the top and the middle band only (TM). The goal is to assess to what extent excluding the bottom region of the face can neutralize the facial expressions for different descriptors and fusion modes. In order to emphasize this effect, we considered only the first level of expression intensity (referred to as *level*-1) of the BU-3DFE. Table I reports the $rank$-1 recognition rates obtained for different combinations of α_1, α_2, H, C, and the gray level GL as texture function, in both a unimodal and a fusion scheme. The table shows also the recognition rate for two types of histogram distances, namely, the cosine distance (cos), and the chi-squared distance (χ^2). First, we notice that the TM grid produces better results across most of the variants. This confirms the capacity of the TM grid matching of reducing the effects of facial shape variation caused by the mouth, while ensuring an overall acceptable recognition accuracy. Looking at the combination between the operator α and the histogram distance, we observe that α_1 and α_2 are best coupled with cos and χ^2, respectively. So, in the subsequent experiments, we used each variant with its best distance (i.e., cos with α_1 and χ^2 with α_2). Regarding the fusion aspect, we can notice the improvement induced by fusing shape and texture at each instance of the aforementioned combinations. In this context, we reported also results related to SI to show the ample improvement brought by the fusion, which is illustrated, for instance, in a jump in the accuracy from 82.43% to 95.65%, and from 78.96% to 95.13% in the $\langle TMB, cos\rangle$ and $\langle TM, \chi^2\rangle$ variants, respectively. We also observe that feature-fusion variants perform better than their score-level counterparts. The variant using $\langle FF1, TM, C, \chi^2\rangle$, in particular, scored the best performance of 97.74%.

Referring to the computational cost and pattern repeatability, the α_1 variant is more appealing than α_2. This also motivated us to not include the $FF2$ fusion modality for α_2, since this would result in a high dimensionality of the fused descriptor with a consequently high computational cost. Nevertheless, α_2 takes advantage, theoretically, in its discriminative power given the wider range of its related patterns. While the results confirm the superiority of the α_2 variant overall, we notice that at some instances, α_1 performs better than α_2. While we do not have a definitive postulate explaining this consistency, we believe that the most plausible one is the intrinsic repeatability of the α_1 variant.

In Table II, the probe scans are categorized into the six different facial expressions, and recognition rates are reported for each category separately. We also included results obtained

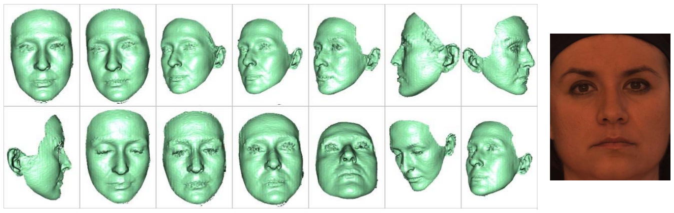
TABLE II

BU-3DFE: RANK-1 RECOGNITION RATE (IN PERCENTAGE) OBTAINED FOR THE DIFFERENT EXPRESSION SUBSETS COMPARED TO [14]

		Level 1 & Level 2 Expressions							Level 3 & Level 4 Expressions						
Descriptors		**AN**	**DI**	**FE**	**HA**	**SA**	**SU**	**All**	**AN**	**DI**	**FE**	**HA**	**SA**	**SU**	**All**
[14]	HOG	90.00	87.50	88.80	88.10	90.60	85.00	88.30	81.30	75.60	78.80	80.60	82.50	76.90	79.30
	SHOT	93.80	90.60	91.90	90.00	94.40	88.80	91.60	87.50	78.80	85.60	79.40	90.00	79.40	83.40
	GH	90.60	85.00	84.40	85.60	90.60	82.50	86.50	86.30	79.40	80.00	79.40	85.00	78.80	81.50
α_1	H	89.00	74.50	83.50	89.00	96.50	93.00	87.58	73.50	48.00	68.00	79.00	85.00	84.00	72.92
	C	88.50	68.00	79.50	86.00	93.00	91.50	84.42	69.00	43.50	65.50	73.50	82.50	83.50	69.58
	GL	83.00	72.50	83.00	83.50	86.00	81.50	81.58	67.00	49.50	71.00	69.00	76.50	71.50	67.42
	FF1 H	95.50	86.50	92.50	94.50	97.00	97.50	93.92	82.50	67.50	86.00	86.50	94.00	94.00	85.08
	FF1 C	94.50	83.00	91.00	94.50	96.50	97.50	92.83	82.00	59.50	86.00	87.00	92.00	93.50	83.33
	FF2 H	94.00	85.50	91.50	95.00	97.50	96.50	93.33	83.00	66.50	85.50	87.50	93.00	93.50	84.83
	FF2 C	94.00	83.00	90.50	92.50	97.50	97.50	92.50	82.50	62.00	85.00	86.50	93.50	93.00	83.75
	SF H	95.00	86.50	92.50	95.00	97.00	98.00	94.00	83.00	67.50	86.50	87.00	94.00	93.50	85.25
	SF C	94.50	85.50	92.50	94.50	97.00	97.50	93.58	82.50	61.00	86.50	88.00	92.50	93.50	84.00
α_2	H	96.50	90.00	95.50	**98.00**	**99.00**	**99.50**	96.42	92.50	72.50	90.50	92.50	98.00	**99.50**	90.92
	C	97.00	89.00	95.50	**98.00**	**99.00**	**99.50**	96.33	92.00	69.00	89.50	92.50	97.50	**99.50**	90.00
	GL	88.00	82.00	87.50	89.50	91.00	87.50	87.58	72.50	58.00	80.00	77.50	81.00	85.50	75.75
	FF1 H	**98.00**	**93.50**	**96.50**	**98.00**	98.50	**99.50**	**97.33**	**94.50**	**80.00**	**92.50**	**95.50**	**98.50**	**99.50**	**93.42**
	FF1 C	97.50	92.50	96.00	**98.00**	**99.00**	**99.50**	97.08	94.00	73.50	91.00	94.50	97.50	**99.50**	91.67
	SF H	**98.00**	**93.50**	**96.50**	**98.00**	98.50	**99.50**	**97.33**	**94.50**	**80.00**	**92.50**	**95.50**	**98.50**	**99.50**	**93.42**
	SF C	97.50	92.50	96.00	**98.00**	**99.00**	**99.50**	97.08	94.00	73.50	91.00	94.50	97.50	**99.50**	91.67

with three variants of the interest-points method proposed in [14] and which have been applied on the same database. Methods in [42] and [45] also used the BU-3DFE database for 3D face recognition, but they are not directly comparable with our due to the different settings. The work in [45] limited the analysis to consistently labeled scans with expression intensities 3 and 4, that do not show large variations in illumination and geometry (total of just 212 scans of 81 subjects out of 2500 scans of 100 subjects). The approach in [42] is based on training multiple SVMs, thus dividing the dataset into two halves of 1200 scans each, one used for training and the other for test. Depending on the fact the intensities 1-2 or 3-4 are used for training, the rank-1 recognition rate is 97.7% and 98.7%, respectively.

From Table II, we first notice the α_2 variant of mesh-LBP outperforms in all the cases the α_1 variant. Compared to the results of Table I, where at *level*-1 expression α_1 and α_2 score similar results. This seems to indicate a major robustness of this latter variant to large and exaggerated expressions. Secondly, we observe that our method outperforms [14] even with variants using single modality (see scores related to H, C and GL with α_2). We notice, in particular, the almost full recognition rate obtained for the *surprise* category. The *disgust* category, which is the most radical expression, exhibits the lowest rate (93.50% for lower level distortions). The distribution of the best scores, highlighted in bold, clearly indicates the recognition enhancement brought by the fusion schemes. Also, we can observe that most of the best scores have been obtained with the feature-level fusion variants, though the score level fusion $\langle \alpha_2, SF, H \rangle$ achieves similar results. This observation is confirmed in the over-all results, whereby the configurations using $\langle \alpha_2, FF1, H \rangle$ and $\langle \alpha_2, SF, H \rangle$ score the best performance.



(a) (b)

Fig. 11. Bosphorus: (a) Samples from the different categories of Bosphorus scans; (b) A sample of the 2D image obtained with the single view scanner used for this database.

B. Bosphorus Database

The Bosphorus database [46], contains 4666 scans of 105 subjects scanned in different poses, action units, and occlusion conditions. Figure 11 shows some scan instances of this database. Notice here that scans are obtained with a single-view scanner, that is one stereo-pod. In particular, the dataset is divided in multiple subsets corresponding to neutral and expressive scans (the six fundamental expressions are considered, namely, *anger*, *disgust*, *fear*, *happy*, *sad*, *surprise*), scans with Action Units (*Lower Face Action Unit* (LFAU), *Upper Face Action Unit* (UFAU), and *Combined Action Unit* (CAU)), scans with rotations (*Yaw Rotation* (YR), *Pitch Rotation* (PR), and *Cross Rotation* (CR)), and scans with *occlusions* (O). Most of the face instances are provided with a set of landmarks that also includes the inner corner landmarks and the nose tip. These three landmarks are those used to define the plane on which the sampling grid of the face is defined (see Sect. III). For the scans with rotation, the inner corner of one of the eyes can be missing. In that case, the partial grid of points is constructed (as illustrated in Fig. 6(d)).

TABLE III

BOSPHORUS: RANK-1 RECOGNITION ACCURACY OBTAINED WITH A SELECTION OF OUR METHOD VARIANTS COMPARED TO [12] AND [14], AND THE BEST TWO VARIANTS REPORTED [13]. THE MAXIMUM OBTAINED RECOGNITION RATE IN EACH SUBSET IS HIGHLIGHTED IN BOLD

	Depth		[12]	[14]	HOMQ [13]		mesh-LBP α_1		mesh-LBP α_2		
	Hist	Score			CGM	FGM	$FF2$ H+GL	SF H+GL	SF H+GL	SF C+GL	$FF3$ H+GL
Neutral	93.30	93.30	**100**	97.9	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Anger	73.24	73.24	88.73	85.9	88.73	**97.18**	**97.18**	**97.18**	94.37	92.96	91.55
Disgust	73.91	73.91	76.81	81.2	76.81	86.96	85.51	89.86	**92.75**	**92.75**	88.41
Fear	62.86	62.86	92.86	90.0	92.86	**98.57**	**98.57**	**98.57**	**98.57**	**98.57**	**98.57**
Happy	87.74	87.74	95.28	92.5	95.28	**98.11**	88.68	91.51	97.17	97.17	97.17
Sad	89.39	89.39	95.45	93.9	94.45	**100**	96.97	98.48	98.48	98.48	98.48
Surprise	56.34	56.34	98.59	91.5	98.59	98.59	97.18	98.59	**100**	**100**	**100**
LFAU	88.19	88.19	97.22	96.5	97.22	98.84	97.09	97.55	99.10	99.03	**99.16**
UFAU	91.67	91.67	99.07	98.4	99.07	**100**	99.77	99.77	**100**	**100**	**100**
CAU	84.02	84.02	98.82	95.6	98.82	**100**	**100**	**100**	**100**	**100**	**100**
Yaw	6.39	6.39	77.96	81.6	77.96	**84.08**	72.0	74.47	56.57	56.19	59.24
Pitch	48.21	48.21	98.81	98.3	98.81	**99.52**	97.85	98.09	94.51	93.32	92.84
Cross	3.32	3.32	94.31	93.4	94.31	**99.05**	92.68	90.24	80.48	80.49	85.37
Occlusion	77.43	77.43	99.21	93.2	99.21	99.21	96.29	95.68	98.76	98.77	**99.38**

Experiments on this dataset aim to test the proposed approach on a larger dataset and in the presence of action units, missing parts and occlusions, in addition to expressions. On this dataset, we can also compare our approach with respect to state of the art methods. In particular, we compared with Li et al. [12], Berretti et al. [14], and Li et al. [13], which share the idea of using keypoints matching, and use the same experimental protocol. Actually, differently from our solution, in these methods keypoints are regarded as points on the mesh-manifold, which are stable over multi-scale differentiation, and which are usually detected using the mesh-DOG operator [17]. Local descriptors constructed at these keypoints are compared in order to find the best matches. In [12], multiple order histograms of differential quantities constructed at each face keypoint and its immediate neighbourhood points are used. In [14], a similar paradigm is used by considering different variety of histogram descriptors. The keypoints matching is also improved using the RANSAC algorithm. In their second version, Li et al. [13], boosted the keypoints matching with the Sparse Representation based Classifier (SRC) [16]. The approaches of Sandbach [26] and Bayramoglu [28] used also the Bosphorus database, but their purpose and setting are different from ours. First, these works assess expression recognition; and second, they employed, respectively, AdaBoost and Random Forest classifiers, and a 10-fold cross-validation scheme, whereas our method used a simple minimum-distance classifier. Besides, they do not consider pose scans in their experiments because of the limitation of the depth images with regard to this category. Therefore, to assess our fusion paradigm on the mesh over its counterpart on the depth images, we compared our method with the 3D-LBP operating on depth images, considering the same aforementioned fusing schemes, namely, score fusion (SF) of the depth and gray-level data, the first and third feature fusion ($FF1$ and $FF3$) of Sect. IV. For the setting of the 3D-LBP face description, the LBP patterns have been computed on 5 rings (radii from 1 to 5) and with an azimuthal resolution of 8. The global histogram is constructed over a grid of 5 × 5 blocs in the depth image, as shown in Fig. 6(c).

Table III depicts the comparison results. First we notice that, despite the fusion scheme, the 3D-LBP on the depth image scores quite below the other methods, for both *histogram* and *score* fusion variants. We can notice that our method neatly outperforms [12], [14], while it competes well with [13], equating and outperforming it at several subsets, noticeably at the *Disgust* and *Surprise* for expressions, *LFAU* for action units, and at the *Occlusion* subset.

For the *Pitch*, and *Occlusion* subsets our scores are reasonably close to [13], whereas the *Cross* subset score is a bit distant. The most critical case for our solution is represented by the *Yaw* subset, where we obtain an accuracy of about 75% for the $\langle SF, H + GL \rangle$ variant of α_1. In order to investigate more this most critical case, we broken-down the *Yaw* rotation subset results, and we found that our method scores well up to 20 degrees rotation as reported in Table IV. If we exclude the 45-degrees results, we obtain an overall score of 86.66%. Interestingly, the α_1 variant resulted more robust than the α_2 for rotation angles of 30 and 45 degrees. Examining the 45-degree scans, we found that the recognition failures in this category are probably due to surface corruption noticed at many instances (Fig. 12 shows some samples). While they do affect the global facial shape, such surface corruptions alter mesh-LBP patterns, which are by principle sensitive to surface artifacts, and consequently will be reflected on the grid histograms.

For the intra-comparison side, referring to the different fusion schemes and the α_1 and α_2 variants of our approach, some considerations can be drawn. As emerged also in the experiments on the BU-3DFE, fusion techniques combined with the α_2 variant seem more robust to expressions than the corresponding α_1 variants, though with a lower gap than in Table II. This is motivated also by the lower intensity of expressions in the Bosphorus dataset. The α_2 variants also show very high accuracy, equal or very close to 100%, on the action unit subsets. The α_1 variants, instead, are neatly more competitive than their α_2 counterparts in the case of rotated scans (as also emerged for the larger rotation angles in Table IV), with the much marked progress observed for the

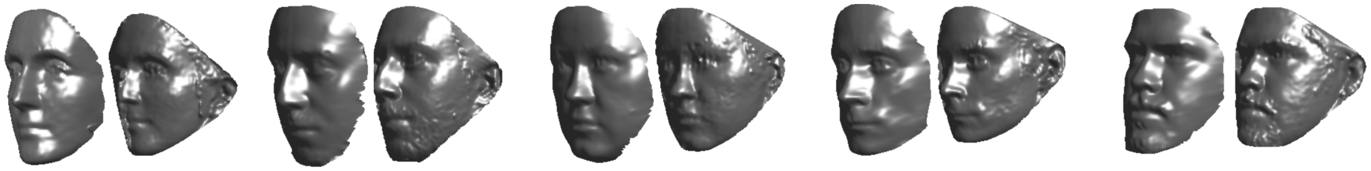


Fig. 12. Samples of the 45-yaw rotated scans failure cases. For each pair, the gallery scan is reported on the left, in the same pose of the probe scan shown on the right.

TABLE IV
BOSPHORUS: RANK-1 RECOGNITION ACCURACY OBTAINED FOR DIFFERENT *Yaw* ROTATION SUBSETS

Descriptors		10°	20°	30°	45°
α_1	FF1 H+GL	98.10	89.52	73.33	46.67
	FF2 H+GL	98.10	87.62	73.33	45.71
	SF H+GL	98.10	89.52	76.19	46.67
α_2	FF1 H+GL	99.05	93.33	64.76	12.38
	SF H+GL	99.05	93.33	64.76	13.33

TABLE V
COMPARISON OF THE ALGORITHM COMPLEXITY OF THE METHOD HQMQ FGM IN [13] AND OUR METHOD

HQMQ FGM [13]		
Stage	**Complexity**	**Details**
Smoothing	O(N)	N: number of vertices
Difference of Gaussians	O(N)	–
Keypoints detection	O(aN)	a: number of scales (3)
Keypoints description	O(bK)	K: number of keypoints (350 average) b: number of neighborhood points (400)
Keypoints matching (Orthogonal Matching Pursuit algorithm)	O(KIG) [51]	I: number of iterations G: number of subjects in the gallery

Our method		
Stage	**Complexity**	**Details**
Geometric descriptors	O(N)	N: number of vertices
mesh-LBP computation	O(cF)	F: number of mesh facets c: number of sampled facets in the mesh-LBP rings (84)
Grid construction	O(de)	d: number of grid points (35) e: number of facets per grid region (336)
mesh-LBP histogram matching minimum distance classifier	O(G)	G: number of subjects in the gallery

Yaw subset. This can be mainly due to the intrinsic rotation invariance, and thus repeatability of the patterns obtained with α_1, which is expected to be much relevant in this case. Last, for the occlusion subset, comparable performance is obtained with a slight prevalence of the α_2 variants.

Table V reports an algorithmic complexity comparison between our method and the best variant of [13] (HQMQ FGM). We can notice that up to the mesh-LBP computation (for our method) and the keypoints detection (for [13]) both methods have a same linear complexity. The keypoint description and the grid construction have both constant complexity. The last two stages, however, show some difference. For the keypoint matching in [13], assuming all galley subjects have a same number of keypoints K, and considering the descriptor size as constant, the algorithmic complexity can be approximated by $O(KIG)$, where I is the number of iterations in Orthogonal Matching Pursuit algorithm (OMP) [47], involving a non-linear minimization used in the keypoint matching, and G is the number of subjects in the gallery. Considering the typical values of $K = 350$ the algorithmic complexity can be estimated as to $O(350IG)$. Using a simple minimum distance classifier, the algorithmic complexity of our method at the mesh-LBP histogram matching is $O(G)$. This indicates that the iterative nature of the OMP algorithm, and the individual keypoint matching in the last stage of (HQMQ FGM) variant in [13] is quite computationally more demanding than its counterpart in our method. For what concerns the size of the face signature, in Li's method it is $m \times K = 261 \times 350 = 75600$, being m the keypoint descriptor size for the HQMQ variant. In ours, it is $35 \times 13 \times 7 = 3185$, and $35 \times 1125 \times 7 = 275625$ for the α_1 and α_2 variants, respectively. These figures, give advantage to Li's method when compared to our α_2 variant.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we presented an original approach for constructing a multi-modal LBP-based face representation on a triangular mesh-model. It is the first approach of its kind that integrates texture and shape information in LBP-patterns derived from a mesh support. This marriage between mesh-model and LBP-based face recognition will open-up new horizons that go quite beyond the limits imposed by the depth image constraints. We proposed a face representation that encompasses a face-centric grid to which is attached, at each point of it, LBP histograms constructed using geometric and photometric data. Contrary to its depth-image counterpart, this representation supports partial facial matching, and does not require normalization. In addition, it preserves the full geometry of the facial shape, which might be partially lost in depth images because of self-occlusion. In addition, we have showed that our framework can be easily adapted to different fusion schemes, in particular the early stage fusion.

Despite having used a basic minimum distance classifier, we showcased the performance enhancement brought by our novel 3D face representation, and demonstrated that it can compete to a reasonable extent with the best methods of the state of the art. Indeed, The experiments conducted with BU-3DFE database showcased the boosting of the recognition performance brought by our fusion framework, and its superiority with regard to the most closest approach. Results obtained on the Bosphorus database report competitive accuracy compared to the state of the art solutions, with an increment for some specific subsets.

Regarding the different variants of our method, including different shape descriptors in the mesh-LBP computation, the α_1 and α_2 weighting functions, and the varying fusion schemes, some summary comments can be drawn. Among the different surface descriptors we tested, the mean curvature (H) resulted the most suited to be combined with mesh-LBP across almost all the experiments. The mean curvature also resulted the optimal option for fusing with the gray level appearance of the surface's facets, using either low-level fusion at the feature level, or late fusion at the score level. The comparison between the α_1 and α_2 variants of mesh-LBP does not come to a univocal conclusion: the α_1 variant is intrinsically invariant to rotation and more efficient from a computational point of view; the α_2 variant, instead, takes advantage from the large gamut of possible values, which makes it more discriminative in most of the cases.

Looking at the performance of our method in the presence of facial expressions, one valid question might raise on how the methods achieve elevated scores for facial expression cases, where the facial surface might undergo significant changes compared to the neutral expression. We believe that this robustness lies first on the choice of TM grid, which discards the lower part of the face that is affected the most by deformation. Also, we think that the small size we choose for the grid regions ($r = 7$) made the representation fine enough to preserve local variability up to large extent. Discarding non-uniform-patterns for α_2 contributes further to the robustness to expressions, since these patterns are mostly located in non-rigid parts of the face.

For what concerns the matching procedure, our method has been employed in a global way, that is considering all the grid points in the matching, without assessing the plausibility of individual pairs of corresponding grid-points. Such procedure is a fundamental part of the methods in [12]–[14], where the plausibility of a pair of potential matching keypoints is evaluated by comparing their related local descriptors. In fact, the boosting of the performance in Li et al. method [13] as compared to their first work in [12] is due to the Sparse Representation based Classifier employed in keypoints matching. However this is without compromising the computation cost, as we have demonstrated it in the algorithmic complexity comparison.

As future work, there are several aspects worth to explore. First, the feature fusion methods we employed used two descriptors, while the numerous descriptors we can derive from the mesh, in addition to the texture, are appealing for investigating a multiple-descriptor fusion. However, we think that this needs to go beyond the standard concatenation and co-occurrence schemes, and that a sound theoretical framework would be necessary for accommodating such a fusion. Second, we believe that integrating a robust mechanism, rather than the simple minimum distance classifier, at the level of grid point matching would considerably boost our method's performance. Third, investigating keypoints framework with mesh-LBP as local descriptors would be novel blending worthwhile to investigate. Finally, optimizing further the size of our face signature, while keeping its discrimination power, noticeably for the α_2 variant.

Overall, we think that our contribution will pave the way for applying the other techniques and methods developed within the LBP-based face recognition directory.

Acknowledgment

Preliminary ideas and experiments of the face mesh-LBP based face recognition approach appeared in [49]. The mesh-LBP code is publicly available at Matlab Central: http://www.mathworks.com/matlabcentral/fileexchange/48875-mesh-lbp.

References

[1] N. Werghi, S. Berretti, A. del Bimbo, and P. Pala, "The mesh-LBP: Computing local binary patterns on discrete manifolds," in *Proc. ICCV Int. Work. 3D Represent. Recognit.*, Sydney, NSW, Australia, Dec. 2013, pp. 562–569.

[2] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[3] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition," *Comput. Vis. Image Understand.*, vol. 101, no. 1, pp. 1–15, Nov. 2006.

[4] S. Berretti, A. del Bimbo, and P. Pala, "3D face recognition using isogeodesic stripes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2162–2177, Dec. 2010.

[5] F. R. Al-Osaimi, M. Bennamoun, and A. Mian, "Spatially optimized data-level fusion of texture and shape for face recognition," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 859–872, Feb. 2012.

[6] H. Drira, B. Ben Amor, M. Daoudi, A. Srivastava, and R. Slama, "3D face recognition under expressions, occlusions, and pose variations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2270–2283, Sep. 2013.

[7] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003.

[8] A. S. Mian, M. Bennamoun, and R. Owens, "Keypoint detection and local feature matching for textured 3D face recognition," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 1–12, Aug. 2008.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[10] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens, and D. Vandermeulen, "Feature detection on 3D face surfaces for pose normalisation and recognition," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Washington, DC, USA, Sep. 2010, pp. 1–6.

[11] D. Smeets, J. Keustermans, D. Vandermeulen, and P. Suetens, "meshSIFT: Local surface features for 3D face recognition under expression variations and partial data," *Comput. Vis. Image Understand.*, vol. 117, no. 2, pp. 158–169, Feb. 2013.

[12] H. Li, D. Huang, P. Lemaire, J.-M. Morvan, and L. Chen, "Expression robust 3D face recognition via mesh-based histograms of multiple order surface differential quantities," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3053–3056.

[13] H. Li, L. Chen, D. Huang, Y. Wang, and J.-M. Morvan, "Towards 3D face recognition in the real: A registration-free approach using fine-grained matching of 3D keypoint descriptors," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 128–142, Jun. 2015.

[14] S. Berretti, N. Werghi, A. del Bimbo, and P. Pala, "Matching 3D face scans using interest points and local histogram descriptors," *Comput. Graph.*, vol. 37, no. 5, pp. 509–525, 2013.

[15] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[17] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 373–380.

[18] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Proc. Eur. Conf. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 469–481.

[19] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[20] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 765–781, Nov. 2011.

[21] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3D+2D based face recognition at both feature and decision levels," in *Proc. Int. Work. Anal. Modeling Faces Gestures*, Beijing, China, Oct. 2005, pp. 44–54.

[22] Y. Huang, Y. Wang, and T. Tan, "Combining statistics of geometrical and correlative features for 3D face recognition," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, Scotland, Sep. 2006, pp. 879–888.

[23] D. Huang, M. Ardabilian, Y. Wang, and L. Chen, "3-D face recognition using eLBP-based facial description and local feature hybrid matching," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1551–1565, Oct. 2012.

[24] G. Sandbach, S. Zafeiriou, and M. Pantic, "Local normal binary patterns for 3D facial action unit detection," in *Proc. IEEE Int. Conf. Image Process.*, Orlando, FL, USA, Sep. 2012, pp. 1813–1816.

[25] H. Li, L. Chen, D. Huang, Y. Wang, and J. Morvan, "3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns," in *Proc. Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 2577–2580.

[26] G. Sandbach, S. Zafeiriou, and M. Pantic, "Binary pattern analysis for 3D facial action unit detection," in *Proc. Brit. Mach. Vis. Conf.*, Guildford, U.K., Sep. 2012, pp. 1–12.

[27] H. Tang, B. Yin, Y. Sun, and Y. Hu, "3D face recognition using local binary patterns," *Signal Process.*, vol. 93, no. 8, pp. 2190–2198, Aug. 2013.

[28] N. Bayramoglu, G. Zhao, and M. Pietikäinen, "CS-3DLBP and geometry based person independent 3d facial action unit detection," in *Proc. Int. Conf. Biometrics (ICB)*, Madrid, Spain, Jun. 2013, pp. 1–6.

[29] K. I. Chang, K. W. Bowyer, and P. J. Flynn, "An evaluation of multimodal 2D+3D face biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 619–624, Apr. 2005.

[30] X. Lu and A. K. Jain, "Deformation modeling for robust 3D face matching," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 1377–1383.

[31] C. Beumier and M. Acheroy, "Face verification from 3D and grey level clues," *Pattern Recognit. Lett.*, vol. 22, no. 12, pp. 1321–1329, Oct. 2001.

[32] M. Hüsken, M. Brauckmann, S. Gehlen, and C. Von der Malsburg, "Strategies and benefits of fusion of 2D and 3D face recognition," in *Proc. IEEE Workshop Face Recognit. Grand Challenge*, San Diego, CA, USA, Jun. 2005.

[33] A. S. Mian, M. Bennamoun, and R. Owens, "An efficient multimodal 2D-3D hybrid approach to automatic face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1927–1943, Nov. 2007.

[34] B. Gökberk, H. Dutağaci, A. Ulas, L. Akarun, and B. Sankur, "Representation plurality and fusion for 3-D face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 155–173, Feb. 2008.

[35] W. B. Soltana, D. Huang, M. Ardabilian, L. Chen, and C. B. Amar, "Comparison of 2D/3D features and their adaptive score level fusion for 3D face recognition," in *Proc. 3D Data Process., Visualizat. Transmiss. (3DPVT)*, Paris, France, May 2010, pp. 1–8.

[36] N. Werghi, S. Berretti, and A. del Bimbo, "The mesh-LBP: A framework for extracting local binary patterns from discrete manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 220–235, Jan. 2015.

[37] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[38] A. Ross and A. Jain, "Information fusion in biometrics," *Pattern Recognit. Lett.*, vol. 24, no. 13, pp. 2115–2125, 2003.

[39] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[40] T. Maurer *et al.*, "Performance of geometrix ActiveID 3D face recognition engine on the FRGC data," in *Proc. IEEE CVPR Workshop Face Recognit. Grand Challenge Experim.*, Jun. 2005, p. 154.

[41] K. I. Chang, K. W. Bowyer, and P. J. Flynn, "Multimodal 2D and 3D biometrics for face recognition," in *Proc. IEEE Int. Workshop Anal. Modeling Faces Gestures (AMFG)*, Oct. 2003, pp. 187–194.

[42] Y. Lei, M. Bennamoun, and A. A. El-Sallam, "An efficient 3D face recognition approach based on the fusion of novel local low-level features," *Pattern Recognit.*, vol. 46, no. 1, pp. 24–37, Jan. 2013.

[43] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[44] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. IEEE 7th Int. Conf. Autom. Face Gesture Recognit.*, Southampton, U.K., Apr. 2006, pp. 211–216.

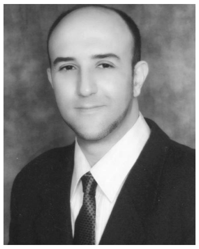
[45] Y. V. Venkatesh, A. A. Kassim, J. Yuan, and T. D. Nguyen, "On the simultaneous recognition of identity and expression from BU-3DFE datasets," *Pattern Recognit. Lett.*, vol. 33, no. 13, pp. 1785–1793, Oct. 2012.

[46] N. Alyüz, B. Gökberk, and L. Akarun, "A 3D face recognition system for expression and occlusion invariance," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl., Syst.*, Washington, DC, USA, Sep./Oct. 2008, pp. 1–7.

[47] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Pacific Grove, CA, USA, Nov. 1993, pp. 40–44.

[48] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[49] N. Werghi, C. Tortorici, S. Berretti, and A. Del Bimbo, "Representing 3D texture on mesh manifolds for retrieval and recognition applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2521–2530.



Naoufel Werghi received the Ph.D. degree in computer vision from the University of Strasbourg. He was a Visiting Professor with the Department of Electrical and Computer Engineering, University of Louisville, in 2002, and the Media Integration and Communication Center, University of Florence, in 2012. He has been a Research Fellow with the Division of Informatics, University of Edinburgh, and a Lecturer with the Department of Computer Sciences, University of Glasgow. He is currently an Associate Professor with the Electrical and Computer Engineering Department, Khalifa University of Science, Technology and Research, UAE. He published over 100 journal and conference papers. His main research area is 2-D/3-D image analysis and interpretation, where he has been leading several funded projects in the areas of biometrics, medical imaging, geometrical reverse engineering, and intelligent systems.

Claudio Tortorici received the bachelor's and master's degrees in computer engineering from the University of Florence, in 2011 and 2013, respectively. He was with the Media Integration and Communication Center as an Internee from 2013 to 2014. He is currently a Research Associate with the Visual Signal Analysis Processing Center, Khalifa University of Science, Technology and Research, UAE. He has been working on research projects related to multimedia information retrieval and 3-D face recognition. He is a Reviewer of the *Scientific World Journal*.

Stefano Berretti received the Ph.D. degree in information and telecommunications engineering from the University of Florence, Italy, in 2001. He has been a Visiting Researcher with IIT Mumbai, India, and a Visiting Professor with the Institute TELECOM, TELECOM Lille 1, Lille, France, and Khalifa University, Sharjah, UAE. He is currently an Associate Professor with the Department of Information Engineering and the Media Integration and Communication Center, University of Florence. He has authored over 100 papers in conference proceedings and international journals in the area of pattern recognition, computer vision, and multimedia. His main research interests focus on 3-D object retrieval and partitioning, face recognition and facial expression recognition from 3-D and 4-D data, and 3-D face superresolution, human action recognition from 3-D data. He is on the program committee of several international conferences and serves as a frequent reviewer of many international journals. He was the Cochair of the Fifth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (2012) in conjunction with ECCV 2012.

Alberto Del Bimbo was the Deputy Rector for Research and Innovation Transfer with the University of Florence from 2000 to 2006. He is a Full Professor of Computer Engineering, the Director of the Master in Multimedia, and the Director of the Media Integration and Communication Center with the University of Florence. His scientific interests are multimedia information retrieval, pattern recognition, image and video analysis, and natural human–computer interaction. From 1996 to 2000, he was the President of the IAPR Italian Chapter and the Member-at-Large of the IEEE Publication Board from 1998 to 2000. He was the General Chair of IAPR ICIAP97, the International Conference on Image Analysis and Processing, IEEE ICMCS99, the International Conference on Multimedia Computing and Systems and Program Cochair of ACM Multimedia 2008. He was the General Cochair of ACM Multimedia 2010 and the European Conference on Computer Vision in 2012. He is an IAPR Fellow, and an Associate Editor of *Multimedia Tools and Applications*, *Pattern Analysis and Applications*, the *Journal of Visual Languages and Computing*, and the *International Journal of Image and Video Processing*, and was an Associate Editor of *Pattern Recognition*, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He serves as the Editor-in-Chief of the *ACM Transactions on Multimedia Computing, Communications, and Applications*.

Joint Registration and Representation Learning for Unconstrained Face Identification

Munawar Hayat*†, Salman H. Khan*‡, Naoufel Werghi††, Roland Goecke†
†University of Canberra, Australia, ‡Data61 - CSIRO and ANU, Australia
††Khalifa University, Abu Dhabi, United Arab Emirates
{munawar.hayat,roland.goecke}@canberra.edu.au, salman.khan@csiro.au, naoufel.werghi@kustar.ac.ae

Abstract

Recent advances in deep learning have resulted in human-level performances on popular unconstrained face datasets including Labeled Faces in the Wild and YouTube Faces. To further advance research, IJB-A benchmark was recently introduced with more challenges especially in the form of extreme head poses. Registration of such faces is quite demanding and often requires laborious procedures like facial landmark localization. In this paper, we propose a Convolutional Neural Networks based data-driven approach which learns to simultaneously register and represent faces. We validate the proposed scheme on template based unconstrained face identification. Here, a template contains multiple media in the form of images and video frames. Unlike existing methods which synthesize all template media information at feature level, we propose to keep the template media intact. Instead, we represent gallery templates by their trained one-vs-rest discriminative models and then employ a Bayesian strategy which optimally fuses decisions of all medias in a query template. We demonstrate the efficacy of the proposed scheme on IJB-A, YouTube Celebrities and COX datasets where our approach achieves significant relative performance boosts of 3.6%, 21.6% and 12.8% respectively.

1. Introduction

Owing to its wide range of potential applications, face recognition has been rigorously researched in computer vision community. Challenges in face recognition are associated with commonly occurring nuisances of facial data which include head pose rotations, illumination variations and expression deformations. In its initial days, facial data was systematically captured in controlled environments and algorithms were developed to individually tackle each of these nuisances [24]. Such algorithms could achieve impressive performance in constrained environments but failed in real-life scenarios. To advance research in unconstrained face recognition, Labelled Faces in the Wild (LFW) [15] and YouTube Faces (YTF) [39] datasets were released in 2007 and 2011 respectively. At the time of their release, the existing methods (developed using constrained data) performed poorly on LFW and YTF. Since then, a large focus of face recognition research has been on the development of algorithms which achieve superior performance on LFW and YTF. With the recent advances in deep learning, the current state of the art algorithms [33, 27] can now achieve human level performance on these datasets. Unconstrained face recognition is however still considered largely unresolved [22]. This is mainly because both LFW and YTF have a well-know frontal selection bias. Specifically, face images in both of these datasets were automatically detected using Viola and Jones [34], which frequently fails for non-frontal faces. The state of the art on YTF and LFW therefore performs poorly in the presence of large head rotations and extreme head poses [22, 6].

In this paper, we aim to address face recognition across extreme head rotations. Registration of such facial images is quite a challenging task and often requires sophisticated pre-processing steps such as landmark localization and frontalization. We propose to automatically learn facial image registration along with feature encoding as part of an end-to-end trainable Convolutional Neural Network. The proposed network (Sec. 3) has two modules: a registration module to learn a set of transformation parameters, and a representation module to learn meaningful feature encoding of input face images. The network is trained on 2.6 million images of 2622 subjects [27]. The proposed scheme is then evaluated on IJB-A [22], YouTube Celebrities [20] and COX [16] datasets for template based face identification. The IJB-A benchmark is specifically quite challenging and contains face images and video frames across extreme head poses and profile views (see Fig. 4). The proposed method achieves a significant performance boost on all of the evaluated datasets (Sec. 5).

*Equal contribution

The problem of face recognition is studied under verification and identification tasks. For verification, we compute a one-one similarity of a given probe face to verify its claimed identity. For identification, one-to-many similarities of the probe are computed in order to find its best match within a gallery of enrolled subjects. Face identification is therefore more challenging compared with face verification. Unconstrained face identification has however been largely neglected over the past few years. This is mainly because most of the research was driven by LFW and YTF datasets and their evaluation protocols are defined for verification only. In this paper, we address template based unconstrained face identification. A template may contain multiple heterogeneous medias in the form of still images or video frames. Face identification from templates is relevant in many commercial systems (*e.g*. FBI's most wanted list) where multiple images of an individual are simultaneously available. Although a template contains more information, it simultaneously poses challenges to effectively utilize this information. Unlike existing methods which merge all template media at feature level, we propose to keep it intact. To leverage from this myriad of information, we train one-vs-rest discriminative models for gallery templates (Sec. 4.3) and employ a Bayesian approach which optimally fuses classification decisions for medias of a given query template (Sec. 4.4).

2. Related Work

A generic face recognition system has three major components: **i)** registration of raw facial images, **ii)** feature encoding of the registered faces, and finally **iii)** classification (verification or identification). In the existing literature, techniques have been developed to individually deal with each of these three components. For **registration**, 2D and 3D face alignment methods have been devised [27, 33, 1]. These methods usually warp automatically detected facial landmarks onto a model face which has a canonical frontal view. For facial feature **representation**, the descriptors can either be manually designed or automatically learnt from large scale facial data. Local Binary Patterns [25], Histogram of Oriented Gradients [7] and Gabor wavelets [42] are some popular examples of the designed features. Most of the recent top performing face recognition methods employ features learnt from a large amount of training data using a Convolutional Neural Network (CNN). Examples include DeepFace [33], VGG-Face [27], FaceNet [30] and DeepID [32]. DeepFace and VGG-Face are based on common CNN architectures whereas FaceNet and DeepID use a specialized inception architecture. As a final step in feature learning, some of these methods employ metric learning (*e.g*. triplet loss embedding [29]) to learn optimal task specific feature embedding (*e.g*. for face verification using LFW and YTF datasets [33, 27]). After registration and feature encoding, the final step is **classification**. Any off-the-shelf classifier can be adapted for verification or identification. Different from previous works, this paper combines registration and representation steps. We propose to learn these as part of a single network. This avoids pre-processing procedures such as landmark localization which are not only computationally expensive but can also introduce many challenges specially in scenarios with extreme head poses (*e.g*. in IJB-A dataset).

With advancements in deep learning for image classification [23, 18, 13], face recognition performances on YTF and LFW datasets have reached human level [33, 30, 32, 27] and began to saturate. To further advance research, IJB-A dataset was introduced recently as a benchmark for unconstrained face recognition. Compared with the existing face datasets, IJB-A is quite challenging since it contains a wide range of appearance variations specially in the form of extreme head poses and variable image quality (see examples in Fig. 4). Since its release, the performances on IJB-A have gradually improved. The top performing methods on IJB-A employ learned feature representations from a large scale external database. For example, CNN features in combination with triplet loss embedding are used in [4, 29]. Chen *et al*. [3] use joint Bayseian metric learning along with CNN features. Five pose-specific CNN models are trained from facial data generated by 3D pose rendering in [1]. Features from a bilinear CNN architecture are used in [4]. The current top performing method [6] on IJB-A dataset uses a template adaptation strategy in combination with learnt features [27]. In order to compute a similarity score between two templates X and Y, it trains two binary classifiers $\mathcal{X}$ and $\mathcal{Y}$. Classifier $\mathcal{X}$ is trained using the media in X as positive class against a large negative media set. Classifier $\mathcal{Y}$ is trained in a similar fashion using the media in Y as the positive class. The similarity score between X and Y is then given by: $\frac{1}{2}\mathcal{X}(y) + \frac{1}{2}\mathcal{Y}(x)$, where $\mathcal{X}(y)$ is the similarity of template Y's media encoding (y) against classifier $\mathcal{X}$.

The IJB-A evaluation protocols are for template based face recognition, where both probe and gallery instances are represented with multiple visual items. Prior to the release of IJB-A dataset, image set classification based face recognition has been actively researched [40, 21, 2, 37, 41, 43, 9, 10, 11, 12]. Similar to a template, an image set is an unordered collection of multiple medias (such as mugshot images or video frames). While template (or image set) based classification provides many promises in the forms of multitude of data being readily available, it simultaneously poses modeling challenges emanating from the heterogeneity of such data in terms of both quality and content. A number of methods have been proposed in the literature to effectively model this information. For example, a template being represented on a non-linear manifold geometry (*e.g*. a point on the Grassmannian manifold [38] or Lie Group of Riemannian manifold [37]) or by media combination (*e.g*. average pooling [8, 26]). In this paper, instead of representing all template medias by a single entity, we propose to keep it intact. The proposed scheme proves to be quite effective (evidenced by its superior performance in Sec. 5) since it avoids loss of any potential information contained in the template.

3. Joint Registration and Representation

Registration of a face to a canonical frontal view is quite crucial for the subsequent feature representation and classification steps. While the recently proposed data driven methods can automatically learn to represent faces, they resort to specially engineered techniques for registration. For example, DeepFace [33] warps a face to a canonical 3D model with the help of detected facial landmarks. In this paper, we propose to learn face registration jointly with the representation. For this purpose, we train a Convolutional Neural Network (CNN) which consists of two interconnected modules (Fig. 2). First, a registration module which learns a set of transformation parameters to optimally register a facial image. Second, a representation module which learns a distinctive feature encoding of the registered face image. The two modules are connected with the output of the registration module being input to the representation module. These modules are described next.

3.1. Registration Module

Registration of facial images typically involves cropping the most relevant facial region (with minimal background) and applying morphing operations on the cropped region to transform it to a canonical frontal view. This usually requires sophisticated facial pre-processing procedures (such as automatic landmark localization) which can be quite challenging, specially in the presence of extreme head poses. In this paper, we propose to adapt a dynamic learn-able mechanism, which automatically estimates a set of optimal parameters to spatially transform a given input face image. Our approach is CNNs based and deploys a Spatial Transformer Network [17] which has three parts: a *localization network* to regress a set of registration parameters. These parameters are then used by a *grid generator*, which outputs a sampling grid. Finally, a *sampler* which maps the input image onto the generated grid. The architecture of the localization network is shows in Fig. 3. Note that the first pooling layer implements mean pooling while the rest perform max operation. A pooling filter of 2×2 pixels is used in all layers. Each parameter layer is followed by a rectifier linear unit (ReLU) layer, except the final fully connected (FC) layer which regresses the transformation parameters.

For a given input image, the localization network outputs a set of six parameters of affine transformation, which are used to generate the sampling grid. The pixel values of the input image are then sampled onto the grid. This results in affine transformations (cropping, translation, rotation, scaling and skewing) of the input image. The registered face image then becomes an input to the subsequent representation module (described next).

3.2. Representation Module

In order to learn facial feature encoding, we employ VGG-16 [27]. It comprises of 8 convolutional and three fully connected layers, each of which is followed by one or more non-linearities (ReLU, pooling). With a relatively simple architecture, VGG-16 has shown superior performance on YTF and LFW benchmarks [27]. The complete network (with both the modules) is then trained using the publicly available face dataset by Parkhi *et al*. [27]. The dataset has 2.6 million face images of 2622 subjects. For training, the detected face regions (provided with the dataset) are loosely cropped. A cropped image contains full face region and may also have some background. The amount of background region is more in case of non-frontal and profile views. The registration module of the network is therefore deployed to only focus on the relevant facial region of interest and ignore any background. The subsequent representation module then learns a discriminative and distinctive feature encoding of the input face image. For an efficient training, we initialize the parameters of the representation module by VGG-Face model [27]. Parameters of the registration modules are initialized by seperately training it to output identity transformation parameters. After learning the parameters of the network, we consider the output of first fully connected layer of the representation module as feature encoding for an input image.

4. Template based Face Identification

A template is a set of images or video frames of the same subject. Face recognition from templates is relevant in scenarios where historical records of observations is readily available and should be leveraged to enhance systems performance. It becomes directly applicable in many real world commercial systems where multiple enrollments of a subject are simultaneously available. Examples include mugshots history of a criminal on the run in forensic search scenarios, lifetime enrollment images in national databases (passports, national identity cards and driver licenses) for access control systems, and multiple images of a person of interest in watch list scenarios (such as FBI's most wanted list). While multitude of heterogeneous data in a templates can be used to enhance face recognition performance, it simultaneously introduces many modeling challenges to make an effective use of this information. To leverage from this information, we propose to learn a discriminative model for each enrolled subject in the gallery and then deploy a

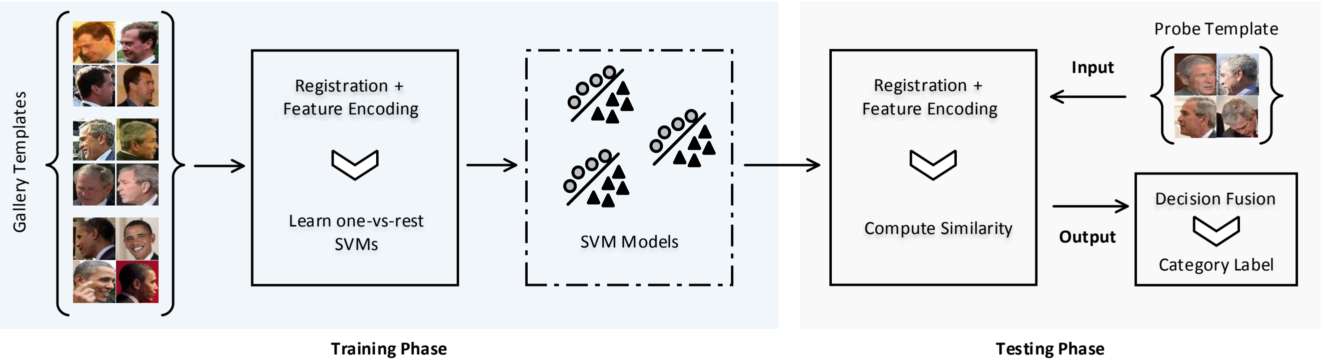


Figure 1: Block diagram of the proposed method. Class-specific discriminative models are learned after joint registration and feature encoding from a deep model during the training process. At test time, these models are used to compute similarity with the enrolled subjects and the individual decisions are combined to obtain a category label.

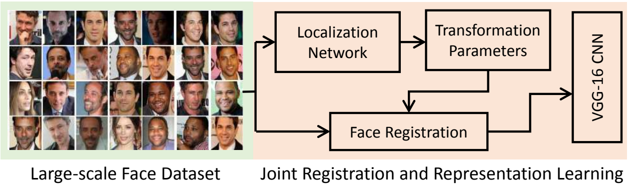

Localization Network
Transformation Parameters
Face Registration
VGG-16 CNN
Large-scale Face Dataset
Joint Registration and Representation Learning

Figure 2: Joint face registration and representation.

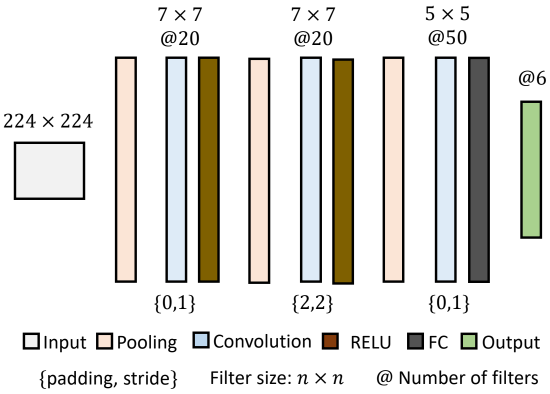


Figure 3: Localization network in the registration module.

score level fusion strategy for the probe templates. The details are given next.

4.1. Problem Description

For template based face identification, the gallery contains N templates $\{\mathcal{T}_1, \mathcal{T}_1, \mathcal{T}_1, \ldots \mathcal{T}_N\}$ corresponding to N enrolled subjects. Each template $\mathcal{T}_i = \{x_1, x_2, \ldots x_M\}$ has M medias (a media is an image or a video frame). Note that M is variable for each enrolled subject. At test time, we are given a query template $\mathcal{T}_q$, and the task is to find its best match with one of the enrolled gallery templates, or determine if $\mathcal{T}_q$ is not enrolled in the gallery.

4.2. Template Media Representation

Given a template $\mathcal{T}_i = \{x_m\} : m = 1 \cdots M$, we encode each media x_m by feed forwarding it through our trained Convolutional Neural Network model (as described in Sec. 3). The output of the first fully connected layer of the representation module is considered as the feature encoding for the template media. Given multiple template media encodings, there are different strategies proposed in the literature to effectively model them. Most of them find a suitable single entity representation for all template media. For example, all images and video frames in the template can be represented by a point on a geometric surface such as Grassmannian manifold [36], or Lie Group of Riemannian manifold [37]. The template media can also be represented by simply taking the mean of all media encodings [26, 8].

In this paper, instead of finding a single entity representation for heterogeneous template data, we propose to keep the media encodings intact. This helps avoid any loss of potential information contained in the template. In order to optimally use the multitude of data contained in the gallery templates, we propose to learn person specific models for each of the enrolled subjects in the gallery (details in Sec. 4.3). To optimally use the probe template data at classification, we employ a fusion strategy (details in Sec. 4.4). In our experimental evaluations (Sec. 5.2), we show that keeping the template media encodings intact is quite effective and results in significant performance boost.

4.3. Person-Specific Discriminative Models

For each of the enrolled subjects in the gallery, we learn a discriminative model. For this purpose, we train a simple one-vs-rest binary SVM classifier. Specifically, to learn the model parameters for a person, we consider feature encodings of all template medias for that person as the positive class, whereas the encodings of the remaining subjects are considered as the negative class. A binary SVM is then trained to learn a hyper-plane which optimally discriminates the two classes.

$$\min_{\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_t \left(\max\left(0, 1 - \ell_t\mathbf{w}^T\mathbf{x}_t\right)\right)^2, \qquad (1)$$

where $\ell_t = \{1, -1\}$. Following this procedure, we learn a set of model parameters $\{\mathbf{w}_i\} : i = 1 \dots N$ for N enrolled subjects in the gallery.

4.4. Query Template Classification

At classification, we are given a query template $\mathcal{T}_q = \{\mathbf{x}_m\} : m = 1 \cdots M$, where $\mathbf{x}_m$ is the encoding for mth media in the template. The task is to find $\mathcal{T}_q$'s best match with the enrolled gallery templates. Using our learnt person-specific models $\{\mathbf{w}_i\} : i = 1 \cdots N$, we can compute a decision value d_i^m for the mth template media to belong to ith enrolled subject. This is given by

$$d_i^m = \frac{1/\left(1 + \exp^{-\mathbf{w}_i^T\mathbf{x}_m}\right)}{\sum_{i=1}^N 1/\left(1 + \exp^{-\mathbf{w}_i^T\mathbf{x}_m}\right)} \qquad (2)$$

The above procedure gives us a set of decision values $\{d_i^m\} : m \in [1, M], i = [1, N]$. In order to combine these multiple decisions for all media in the query template, we explore two schemes. *First*, a simple mean of decision values approach, where given $\{d_i^m\}$, the predicted class label y_q of the query template $\mathcal{T}_q$ is determined by,

$$y_q = \arg\max_i \sum_m d_i^m. \qquad (3)$$

Second, we employ a Bayesian approach inspired by the Bayesian Classifier Combination (BCC) model proposed in [19]. For each of the template media $\mathbf{x}_m$, we have a hidden true label $y_i \in [1, N]$ which matches it with an enrolled subject. We assume conditional independence between decisions d_i^m given the actual label y_i. Let us assume that y_i is generated by a multinomial distribution whose parameters are denoted by $\mathbf{p} : p(y_i = j|\mathbf{p}) = p_j$, where p_j represents the class probabilities (or proportions). Similarly, it can be assumed that decisions d_i^m for each media are generated by a multinomial distribution whose parameters are denoted by $\boldsymbol{\pi}_j^m : p(d_i^m = k|y_i = j) = \pi_{j,k}^m$. Note that $\boldsymbol{\pi}_j^m$ represents the rows of the confusion matrix $\boldsymbol{\pi}^m$ corresponding to each media representation. Therefore, the discriminative ability of each media representation is encoded in the Bayesian model.

The prior distributions for the parameters $\boldsymbol{\pi}_j^m$ and $\mathbf{p}$ are modeled by Dirichlet distributions with hyper-parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$p(\boldsymbol{\pi}_j^m|\boldsymbol{\alpha}_j^m) = \text{Dir}(\boldsymbol{\pi}_j^m; \boldsymbol{\alpha}_j^m) \qquad (4)$$

$$p(\mathbf{p}|\boldsymbol{\beta}) = \text{Dir}(\boldsymbol{p}; \boldsymbol{\beta}) \qquad (5)$$

Here, $\boldsymbol{\alpha}_j^m = [\alpha_{j,1}^m \dots \alpha_{j,N}^m]$ and $\boldsymbol{\beta} = [\beta_1 \dots \beta_N]$. Further, we also define $\boldsymbol{\pi} = \{\boldsymbol{\pi}_j^m : j \in [1, N], m \in [1, M]\}$ and $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_j^m : j \in [1, N], m \in [1, M]\}$. Then, we can define the joint posterior probability of the unobserved variables conditioned on the observed class decisions as:

$$p(\boldsymbol{y}, \mathbf{p}, \boldsymbol{\pi}|\boldsymbol{d}) \propto \prod_{i=1}^N \left\{ p_{y_i} \prod_{m=1}^M \pi_{y_i, d_i^m}^m \right\} p(\mathbf{p}|\boldsymbol{\beta})p(\boldsymbol{\pi}|\boldsymbol{\alpha}) \qquad (6)$$

The original BCC model [19] utilizes Gibbs sampling for inference which is computationally expensive and slow in convergence. To achieve an efficient approximate inference, we use the Variational Bayesian (VB) formulation of Simpson *et al.* [31] which works similar to the Expectation Maximization (EM) algorithm. The VB approach analytically approximates posterior distribution $p(\boldsymbol{y}, \mathbf{p}, \boldsymbol{\pi}|\boldsymbol{d})$ (defined in Eq. 6) by a simpler and tractable distribution $q(\boldsymbol{y}, \mathbf{p}, \boldsymbol{\pi})$ which factorizes over its variables as follows:

$$q(\boldsymbol{y}, \mathbf{p}, \boldsymbol{\pi}) = q(\boldsymbol{y})q(\mathbf{p})q(\boldsymbol{\pi}) \qquad (7)$$

where,

$$q(y_i = j) = \mathbb{E}_{\boldsymbol{y}}[y_i = j] = \rho_{i,j} / \sum_{k=1}^N \rho_{i,k} \qquad (8)$$

$$\text{s.t. } \rho_{i,j} = \exp(\mathbb{E}_{\mathbf{p}}[\ln p_j] + \sum_{m=1}^M \mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_{j,d_i^m}^m]) \qquad (9)$$

$$q(\boldsymbol{p}) \propto \text{Dir}(\boldsymbol{p}; \boldsymbol{\beta}) \qquad (10)$$

$$q(\boldsymbol{\pi}_j^m) \propto \text{Dir}(\boldsymbol{\pi}_j^m; \boldsymbol{\alpha}_j^m) \qquad (11)$$

where the hyper-parameters are updated as follows:

$$\boldsymbol{\alpha}_j^m = \hat{\boldsymbol{\alpha}}_j^m + \left[\sum_{i=1}^N \delta_{[\![d_i^m = k]\!]} \mathbb{E}_{\boldsymbol{y}}[y_i = j]\right]_{k=1}^N$$

$$\boldsymbol{\beta} = \hat{\boldsymbol{\beta}} + \left[\sum_{i=1}^N \mathbb{E}_{\boldsymbol{y}}[y_i = k]\right]_{k=1}^N \qquad (12)$$

$\hat{\boldsymbol{\alpha}}_j^m, \hat{\boldsymbol{\beta}}$ denote the previous estimate of hyper-parameters. Using the current estimates of expectations in Eq. 8, we update the variational distribution in Eq. 7 (E-step). We then update the expectations in Eq. 8 as follows (M-step):

$$\mathbb{E}_{\mathbf{p}}[\ln p_j] = \frac{\Gamma'(\beta_j)}{\Gamma(\beta_j)} + \frac{\Gamma'(\sum_{k=1}^N \beta_k)}{\Gamma(\sum_{k=1}^N \beta_k)} \qquad (13)$$

$$\mathbb{E}_{\boldsymbol{\pi}}[\ln \pi_{j,d_i^m}^m]) = \frac{\Gamma'(\alpha_{j,d_i^m}^m)}{\Gamma(\alpha_{j,d_i^m}^m)} + \frac{\Gamma'(\sum_{k=1}^N \alpha_{j,k}^m)}{\Gamma(\sum_{k=1}^N \alpha_{j,k}^m)}, \qquad (14)$$

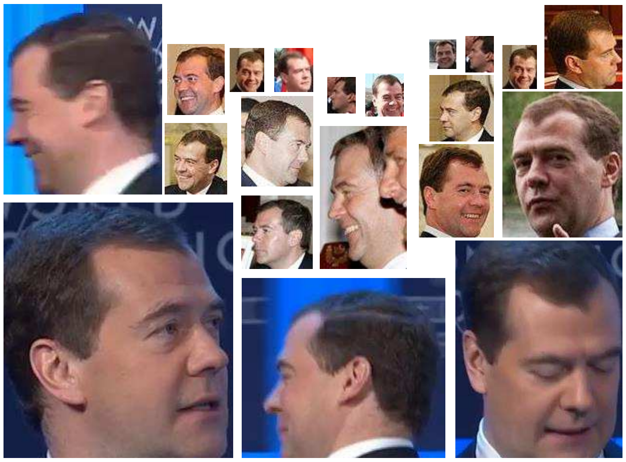


Figure 4: Sample Images of a person from IJB-A dataset. Note the extreme head poses and variations in image resolutions.

where $\Gamma(\cdot)$ is the standard gamma function used in the normalization constant of Dirichlet distributions. The VB algorithm for decision fusion works by iteratively updating the hidden output variables (actual labels y) and the model parameters $(\boldsymbol{\pi}, \mathbf{p})$.

5. Experiments

We extensively evaluate the performance of our proposed method on three datasets: IJB-A [22], YouTube Celebrities (YTC) [20] and COX [16]. For performance evaluation and comparison with existing state of the art, we use Cumulative Match Characteristics (CMC) and Decision Error Trade off (DET) curves. These metrics are defined in Sec. 5.2. Below, we first briefly describe the datasets used in our experiments.

5.1. Datasets

IJB-A dataset: contains 5712 images and 2085 videos of 500 subjects (from diverse geographic locations) captured in real life scenarios. While majority of other face recognition datasets contain either still images or video frames, IJB-A dataset contains both. The images and frames in the dataset exhibit diversity in terms of ethnicity, country of origin and head poses. The most challenging aspects of the dataset are the appearance variations caused by extreme head poses and variable image resolution. A few example images of a subject are shown in Fig. 4. In the presence of such extreme head rotations, automatic face detection fails quite often. The media in the dataset was therefore manually annotated to generate facial bounding boxes [22]. This avoids any frontal selection bias as a result of automated face detection failures in the presence of extreme head poses (e.g., in YTF and LFW datasets).

The IJB-A dataset is released with well-defined evaluation protocols. For template based face identification, 10 random training and testing splits are provided. Each split uses data of all 500 subjects with 333 subjects randomly sampled into the training set and the remaining 167 subjects form the testing set. The testing set contains probe and gallery templates. In order to make evaluation further challenging, 55 (randomly sampled) out of 167 subjects are removed from the gallery in the testing set. This corresponds to scenarios where probe subjects are not enrolled in the gallery. The probe templates of all 167 subjects are to be searched against the gallery templates of only 112 subjects.

YouTube celebrities [20] dataset contains 1910 videos of 47 celebrities downloaded from YouTube. Since the videos are acquired in real life situations, the resolution of the face images is very low and automatic face detection [34] fails for many videos. We therefore use tracking [28] to extract face regions from video frames. The extracted face region is then re-sized to 30 × 30 pixels. For template based face identification, we use five fold cross validation experimental protocol as in [14, 37]. Specifically, the complete dataset is divided into five equal splits with minimal overlap. Each split has nine templates (termed as image sets in [37, 14, 2]) per subject, three of which are used to form the gallery whereas the remaining six are the probe templates.

COX [16] dataset contains 4000 uncontrolled low resolution video sequences of 1000 subjects. In order to capture the videos, the subjects are asked to walk naturally inside a gymnasium without enforcing any constraints on their facial expression, lighting conditions and head poses. For our template based face identification experiments, we consider the frames of each video as a template and follow a leave-one-out strategy. Specifically, one template per subject is held-out as probe whereas the remaining form the gallery. For consistency, four runs of experiments are performed by swapping the probe and gallery templates.

5.2. Results

Evaluation Metrics: Face identification performance is commonly evaluated in terms of a Cumulative Match Characteristics (CMC) curve. A CMC curve plots identification rates corresponding to different ranks. A rank-k identification rate is defined as the percentage of probe searches whose gallery match is returned within the top-k matches. For scenarios where probes are not necessarily enrolled in the gallery, face identification performance is evaluated in terms of a Decision Error Trade-off (DET) curve, which plots False Negative Identification Rate (FNIR) vs False Positive Identification (FPIR) rate as a function of a similarity threshold for the top 20 candidates in the gallery. FPIR is the proportion of non-mate (not enrolled) probe searches returned above a similarity threshold. FNIR is the proportion

Table 1: Performance Evaluation on IJB-A dataset.

Methods	TPIR@FPIR=0.01	TPIR@FPIR=0.1	TPIR@Rank=1	TPIR@Rank=10
Bilinear-CNN [5]	14.2 ± 2.7	34.1±3.2	58.8 ± 2.2	–
Face Search [35]	38.3 ± 6.3	61.3 ± 3.2	82.0 ± 2.4	–
Deep Multipose [1]	52.0	75.0	86.4	94.7
Triplet Similarity [3]	55.6 ± 6.5	75.4 ± 1.4	88.0 ± 1.5	97.4 ± 0.6
Joint Bayesian [29]	57.7 ± 9.4	79.0 ± 3.3	90.3 ± 1.2	97.7 ± 0.7
VGG-Face [6, 27]	46.1 ± 7.7	67.0 ± 3.1	91.3 ± 1.1	98.1 ± 0.5
Template Adaptation [6]	77.4 ± 4.9	88.2 ± 1.6	92.8 ± 1.0	98.6 ± 0.3
This Paper	88.6 ± 4.1	96.0 ± 1.0	96.4 ± 0.8	100.0 ± 0.0

.

of mate (enrolled) probe searches which are returned either below a similarity threshold or outside the top 20 ranks. For DET, we report True Positive Identification Rates (TPIR) at FPIR of 0.1 and 0.01, where TPIR= 1−FNIR.

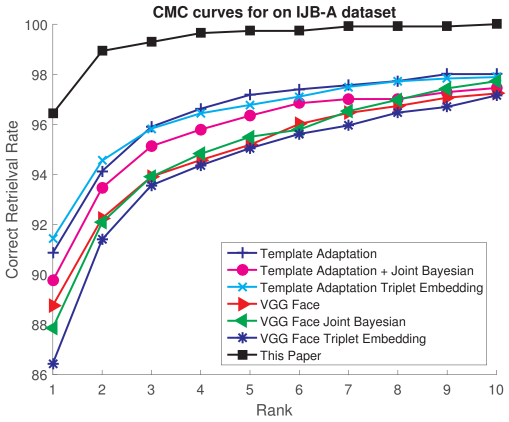


Figure 5: CMC curves on IJB-A dataset (best in colors).

Results on IJB-A Dataset: We compare the face identification performances on IJB-A benchmark in Table. 1. The results for the existing methods are reported from [6]. Due to a standard evaluation protocol on IJB-A dataset, the reported results are directly comparable. Our proposed method achieves average rank-1 and rank-10 identification rates of 96.4% and 100.0% respectively. For evaluations in the presence of non-mate probe searches, our method achieves average TPIR of 88.6% and 96.0% corresponding to FPIR of 0.01% and 0.1% respectively. Compared with the existing state of the art, the proposed method gains a relative performance boost of 3.9% (rank-1), 1.4% (rank-10), 8.8% (@FPIR=0.1) and 14.5% (@FPIR=0.01).

Results on YTC and COX Datasets: We further validate the efficacy of our proposed method on YTC and COX datasets. These datasets have been used in the literature for performance evaluation of image set classification methods. For the purpose of this paper, an image set can be considered as a template, as it contains multiple images or video frames. In Figure 6, we compare the performance of our method with a number of recently introduced image set classification methods. These include Mutual Subspace Method (MSM) [40], Discriminant Canonical Correlation Analysis (DCC) [21], the linear version of the Affine Hull-based Image Set Distance (AHISD) [2], Sparse Approximated Nearest Points (SANP) [14], Co-variance Discriminative Learning (CDL) [37], Regularized Nearest Points (RNP) [41], Set to Set Distance Metric Learning (SSDML) [43], Non-Linear Reconstruction Models (NLRM) [9] and Reverse Training (RT) [10]. For the compared methods, we use standard implementations provided by the respective authors. In order to encode facial images, we first use the original features proposed in the respective papers. We also evaluate the existing methods with our proposed features. The experimental results summarized in Figure. 6 show that our proposed method significantly outperforms the current state of the art by achieving average rank-1 identification rates of 90.1% and 83.6% on YTC and COX datasets respectively.

5.3. Discussion

We believe two major aspects of the proposed method contribute to its achieved superior performance. *First*, its strong feature representation capability. The proposed method learns to automatically register raw facial images while simultaneously finding a distinctive feature representation. Below, we show the effectiveness of the proposed features by evaluating them with existing methods. *Second*, its capability to synthesize multitude of information in the template media with proposed decision level fusion scheme. We further elaborate these aspects next.

Facial Feature Encoding: In order to demonstrate the effectiveness of our proposed learnt features, we evaluate them in conjunction with the existing image set classification methods in the literature. Specifically, instead of using the original features proposed in their respective papers, we use the facial features extracted by our method. By keeping the rest of the pipeline for the compared image set classi-

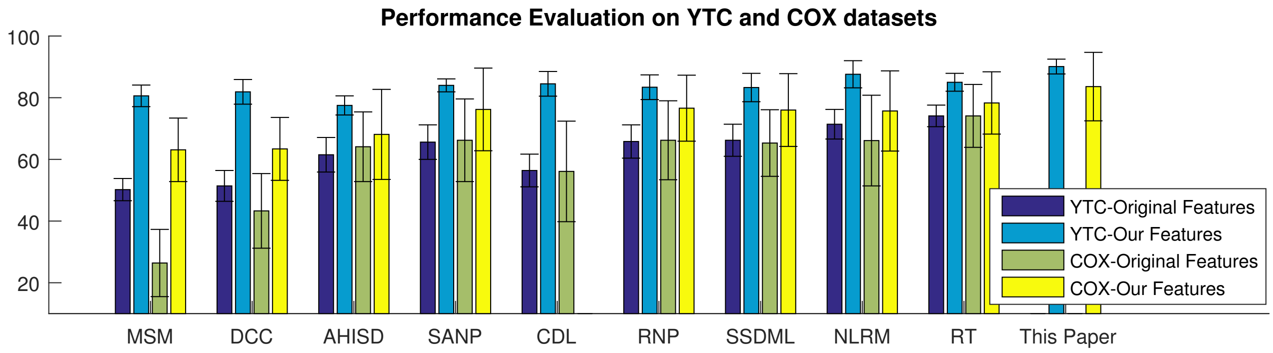
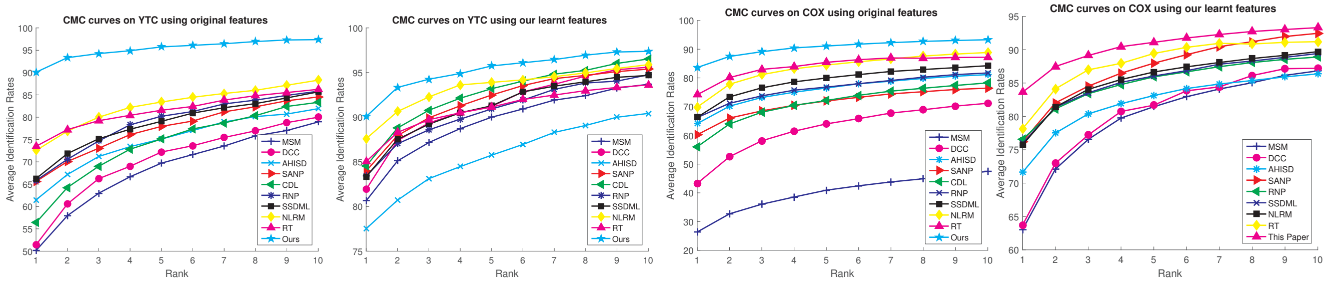


Figure 6: Rank-1 identification rates of different image set classification methods on YTC and COX datasets. Due to high memory requirements, CDL could not be evaluated on COX dataset with learnt features. Figure best seen in colors.



(a) YTC: Original Features (b) YTC: Learnt Features (c) COX: Original Features (d) COX: Learnt Features

Figure 7: CMC curves for different methods on YTC and COX datasets using their original features (a) & (c), and our learnt features (b) & (d). Figure best seen in colors.

fication methods fixed, our experimental results in Fig. 6 suggest that the performance of all methods significantly improves in combination with our proposed features. Note that due to large memory requirements, we were unable to evaluate CDL using learnt features on the COX dataset with available computational resources. CMC curves on the YTC and COX datasets in Figure 7 demonstrate that a consistent performance boost is achieved across all ranks.

Fusion - Feature vs Decision Level: For template (or image set) based face identification, multitude of information is present in the form of heterogeneous template media. Effectively utilizing this information is quite crucial to the overall face identification performance. In the existing literature, different strategies have been devised to find a suitable representation for the template media. These include a template represented by a point on a manifold geometry [38, 37], representative exemplars (*e.g.* derived from affine or convex hull models [2]) or by simply pooled media encodings [26, 8]. The existing methods therefore combine the information from multiple template medias at feature (media) level. In this paper, we keep the template media intact and do not find any single entity representation. Instead, we propose to synthesize information from all template medias at decision level. Even with the simple mean of decision values approach, we achieve a rank-1 identification rate of 94.2 ± 0.9 on IJB-A dataset. The proposed scheme to fuse information at decision level, instead of feature level, therefore avoids any potential loss of information and yields superior performance.

6. Conclusion

Template based face identification is pertinent in many real-world applications where multiple images of a persons' face are concurrently available, such as security and surveillance systems, watch list scenarios and access control systems. We presented a simple yet effective strategy to handle multitude of template media information. Unlike existing methods, which combine this information at initial feature level, we employed a Bayesian approach to fuse it later at decision level. For registration of unconstrained face data with extreme head poses, we presented a data driven approach to jointly learn registration with representation in a single Convolution Neural Network. Effectiveness of the proposed schemes is demonstrated by their significantly superior performance on challenging unconstrained face identification benchmarks.

References

[1] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, et al. Face recognition using deep multi-pose representations. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[2] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *Computer Vision and Pattern Recognition, 2010. CVPR 2010. IEEE Conference on*, pages 2567–2573. IEEE, 2010.

[3] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[4] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 118–126, 2015.

[5] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[6] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv preprint arXiv:1603.03958*, 2016.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[8] T. Hassner, I. Masi, J. Kim, J. Choi, and S. Harel. Pooling faces: Template based face recognition with pooled face images. In *CVPR workshop*, pages 59–67. IEEE, 2016.

[9] M. Hayat, M. Bennamoun, and S. An. Learning non-linear reconstruction models for image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014.

[10] M. Hayat, M. Bennamoun, and S. An. Reverse training: An efficient approach for image set classification. In *European Conference on Computer Vision*, pages 784–799. Springer, 2014.

[11] M. Hayat, M. Bennamoun, and S. An. Deep reconstruction models for image set classification. *IEEE transactions on pattern analysis and machine intelligence*, 37(4):713–727, 2015.

[12] M. Hayat, S. H. Khan, and M. Bennamoun. Empowering simple binary classifiers for image set based face recognition. *International Journal of Computer Vision*, 2017.

[13] M. Hayat, S. H. Khan, M. Bennamoun, and S. An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016.

[14] Y. Hu, A. S. Mian, and R. Owens. Face recognition using sparse approximated nearest points between image sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1992–2004, 2012.

[15] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report.

[16] Z. Huang, S. Shan, H. Zhang, S. Lao, A. Kuerban, and X. Chen. Benchmarking still-to-video face recognition via partial and local linear discriminant analysis on COX-S2V dataset. In *Computer Vision–ACCV 2012*, pages 589–600. Springer, 2013.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[18] S. H. Khan, M. Hayat, M. Bennamoun, R. Togneri, and F. A. Sohel. A discriminative representation of convolutional features for indoor scene recognition. *IEEE Transactions on Image Processing*, 25(7):3372–3383, 2016.

[19] H.-c. Kim and Z. Ghahramani. Bayesian classifier combination. In *International Conference on Artificial Intelligence and Statistics*, pages 619–627, 2012.

[20] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Computer Vision and Pattern Recognition (CVPR), 2008 IEEE Conference on*, pages 1–8. IEEE, 2008.

[21] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1005–1018, 2007.

[22] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939. IEEE, 2015.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[24] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*, pages 189–248. Springer, 2016.

[25] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[26] E. Ortiz, A. Wright, and M. Shah. Face recognition in movie trailers via mean sequence sparse representation-based classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3531–3538, 2013.

[27] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015.

[28] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[29] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *arXiv preprint arXiv:1602.03418*, 2016.

[30] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[31] E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, pages 1–35. Springer, 2013.

[32] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[33] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[34] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[35] D. Wang, C. Otto, and A. K. Jain. Face search at scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[36] R. Wang and X. Chen. Manifold discriminant analysis. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 429–436. IEEE, 2009.

[37] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[38] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[39] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.

[40] O. Yamaguchi, K. Fukui, and K.-i. Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition (FG), 1998 IEEE International Conference on*, pages 318–323. IEEE, 1998.

[41] M. Yang, P. Zhu, L. V. Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. pages 1–7, 2013.

[42] P. Yang, S. Shan, W. Gao, S. Z. Li, and D. Zhang. Face recognition using ada-boosted gabor features. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 356–361. IEEE, 2004.

[43] P. Zhu, L. Zhang, W. Zuo, and D. Zhang. From point to set: Extend the learning of distance metrics. In *International Conference on Computer Vision (ICCV), 2013 IEEE Conference on*. IEEE, 2013.

Multiscale Roughness Approach for Assessing Posterior Capsule Opacification

Aruna Vivekanand, Naoufel Werghi, *Member, IEEE*, and Hussain Al-Ahmad, *Senior Member, IEEE*

***Abstract*—Posterior capsule opacification (PCO) is a common complication in patients who have undergone cataract surgery, occurring in up to 50% of patients by two to three years after the operation. Assessment of PCO has been mainly subjective, making it difficult to understand its progression over time or assess the effectiveness of strategies used for the prevention of PCO. Fully automated PCO assessment systems developed so far offer objective grades. However, they do not provide morphological PCO data useful for an effective analysis of scores. This paper proposes a novel method based on multiscale roughness estimation to detect and quantify the PCO areas. This method is also characterized by its robustness against monotonic illumination variations. Extensive experimentation showcases a distinctive analysis and assessment power of our method compared to other competitive methods. The results show a high correlation of 84.6% with respect to clinical scores**

***Index Terms*—Computer-aided detection, entropy, illumination variation, multiscale roughness, posterior capsule opacification (PCO), segmentation.**

I. Introduction

POSTERIOR capsule opacification (PCO) is a common complication of cataract surgery in patients who have undergone the extra capsular cataract extraction surgery. PCO is caused by the growth of the lens epithelium cells (LECs) remaining in the posterior capsular area of the eye after the cataract surgery. These cells develop as different types of PCO, namely, pearls, fibrosis, and wrinkles as shown in Fig. 1. Severe PCO causes blurry vision, which may be worse than it was before cataract surgery. Though PCO can be corrected by Nd:YAG laser capsulotomy, the treatment apart from being expensive is associated with increased risk of retinal detachment [1]. Considerable research has been done to study the influence of surgical techniques, lens material, and design on the growth of PCO. However, the highly subjective clinical assessment of PCO makes it difficult to assess the effectiveness of strategies suggested for the prevention of PCO. Hence, there is a need for development of a standard PCO quantification system, which ensures an objective and reliable assessment of PCO with minimal human intervention.

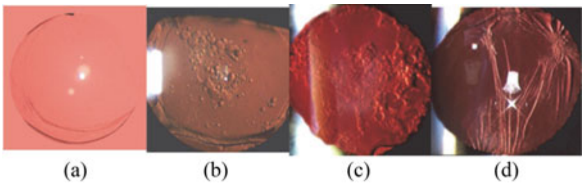


Fig. 1. PCO image samples: (a) clear eye capsule, (b) pearls PCO, (c) fibrosis PCO, and (d) wrinkles PCO.

Digital images captured using retroillumination and Scheimpflug photography helped computerize the image analysis. The basic idea is to detect and classify the PCO areas in the digital image and then quantify PCO from the number of classified pixels. A number of digital image analysis techniques [2]–[4] have been proposed for the quantification of PCO. However, these systems provide subjective scores due to human involvement. PCO images demonstrate complex nondeterministic texture, and in many cases, PCO areas can have similar intensity and texture compared to the clear areas, depending on the severity of PCO and inherent uneven illumination conditions of retroillumination photography [5]. These factors further increase the difficulty of segmentation of PCO images.

An ideal PCO quantification system should be capable of handling a variety of PCO textures and provide morphological details along with the quantification of PCO without human intervention. The system should also be robust to illumination variations as the captured PCO images vary widely with respect to background illumination. In order to address these issues, this paper proposes a method that computes the multiscale roughness values for every pixel in the image. The roughness image obtained is then clustered using histogram-based thresholding technique. Finally, PCO percentage is computed from the number of pixels present in each cluster. The results obtained using this method showed high correlation with respect to clinical grading and other existing methods apart from being robust to monotonic illumination variations.

The rest of the paper is organized as follows: Section II provides the background to the existing methods of PCO assessment. Section III provides the details of the proposed method of PCO assessment. Section IV comprises the experimental results and comparison with the previous methods of PCO assessment. The conclusions of our research work are presented in Section V.

II. Background

Quantitative assessment of PCO has received the attention of many researchers from both the medical and computer vision community. Tetz *et al.* [2] proposed an interactive system

Manuscript received August 12, 2013; revised November 23, 2013; accepted January 29, 2014. Date of publication February 6, 2014; date of current version November 3, 2014. This work was supported by the National Research Foundation under Grant UIRCA 2012-21803.

The authors are with the Department of Electrical and Computer Engineering, Khalifa University of Science and Technology, Sharjah 573, UAE (e-mail: aruna.vivekanand@gmail.com; naoufel.werghi@kustar.ac.ae; alahmad@kustar.ac.ae).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JBHI.2014.2304965

called evaluation of PCO (EPCO) whereby the operator manually traces the boundaries of the textured areas in the image. Then, the operator assigns a color code with a score between 0 and 4 to each textured area in the image based on the severity of PCO. A PCO grade is then calculated by summing the scores weighted by the fractional area of each zone. Though the scores are highly influenced by operator bias, this system has been widely used as a standard in clinical comparison studies. The intensity-based thresholding technique developed by Wang and Woung [6] is very sensitive to illumination variations. The intensity-based segmentation techniques like k-means, fuzzy C-means, and mean shift do not work for the segmentation of PCO images as the PCO areas tend to exhibit intensity levels similar to the clear areas. Hence, it becomes necessary to use texture analysis for the segmentation of PCO images.

Searching for a particular pattern is not possible with the PCO images, because they possess quite an un-deterministic nature. Paplinski and Boyce [4] proposed that PCO areas in an image are rich in texture. They calculated a set of conjugate images from the original image using directional variance edge filters. Gray-level co-occurrence matrices (GLCMs) are obtained from the conjugate images. The thresholds obtained from the main diagonal of GLCM are used for the segmentation of PCO images. However, the distribution of the gray levels in the image data is very often not truly Gaussian; hence, it is not possible to find an optimum threshold value automatically and best segmentation results are observed when the thresholds are set manually [7]. Posterior Capsule Opacity (POCO) software [5] is based on the same principle with a technique devised to find thresholds automatically. However, this software does not provide a measure of severity of PCO; hence, it might result in high scores even in some mild cases of PCO when a thin LEC membrane covers most of the capsular area. Moreover the texture-based segmentation methods were only able to detect the borders of pearls and incapable of detecting the interiors of pearls as severe PCO, because the interior areas exhibit smooth texture, which is similar to the neighboring clear areas.

Based on the fact that PCO areas are characterized by randomness, statistical measures were used for the quantification of PCO. Automated quantification of after cataract (AQUA) system [8] and open-access systematic capsule assessment (OSCA) system [9] have used entropy for texture analysis as this feature measures the randomness of gray-level distribution. AQUA system used entropy calculated from GLCM for texture analysis, whereas OSCA system used local entropy calculated from the intensity histogram of the region as a measure of randomness of gray-level distribution. Additionally, OSCA system takes the location of PCO areas into consideration when calculating PCO scores, as the areas of PCO distant from the center of the visual axis have a reduced effect on vision compared to PCO areas at the center [10]. However, PCO scores based on entropy calculation are highly sensitive to illumination variations across the image. It was proposed by Werghi *et al.* [11] that roughness can be used as a texture measure to quantify PCO. They evaluated the roughness using the concept of regions. Pixels are classified into clusters based on their chromatic values and each class of pixels is decomposed into regions. PCO score is computed from the total number of regions. A limitation with this approach is that the system might result in low PCO scores in some cases of severe PCO when the PCO image exhibits a few number of homogenous PCO regions but with large areas. Also, none of these methods provide morphological details of PCO. The morphological details of PCO are important for validating PCO scores and identification of patients for an effective treatment of PCO. Grewal *et al.* [12] introduced a new method for PCO quantification using Pentacam, a rotating Scheimpflug imaging system. However, unlike the retroillumination imaging system, Scheimpflug system is not available at most ophthalmological departments. Further the scatter light intensity measurements of Scheimpflug system are influenced by the intraocular lens (IOL) material as shown by Tanaka *et al.* [13], which would lead to inaccuracies in PCO quantification results. Hence, this leaves scope for further investigation of a method that can model the complex texture of PCO.

III. Proposed Method

Our proposed method of PCO assessment consists of three stages: preprocessing, estimation of multiscale roughness, and classification.

A. *Preprocessing*

The images are obtained using the retroillumination photography. Since the central area of the capsule is considered to be visually significant [14], the central 60% of the area is treated as the region of interest and the remaining area is set with the average intensity level, after which the images are converted to gray scale. The images captured using retroillumination photography have characteristically uneven background illumination. Hence, the background illumination is estimated by performing morphological opening using a disk-shaped structuring element. The size of the structuring element is chosen such that it does not contain the details of the image. Finally, the preprocessed image is obtained by subtracting the illumination estimate from the original image.

B. *Estimation of Point-Wise Multiscale Roughness*

The proposed method uses roughness as a texture measure to quantify PCO. It is based on the idea that a pixel in the interiors of pearls may demonstrate smoothness in the immediate vicinity but exhibit roughness when estimated over a larger area. Hence, the multiscale roughness value for each pixel (x, y) in the image is derived from the roughness values estimated over different concentric rings surrounding the pixel as shown in Fig. 2, in an attempt to simulate the human ability to visualize the uniformity of a region at different scales. The bilinearly interpolated neighboring pixel values are considered in the computation of roughness values. The rings with larger radius contain larger number of neighbors to capture a larger area.

This idea of multiscale descriptor has been inspired from the Holder exponent representation, which is used as a regularity descriptor to describe the salient features of an image. Holder exponent is described by Liu and Li [15] using (1). $\alpha(x)$ is called

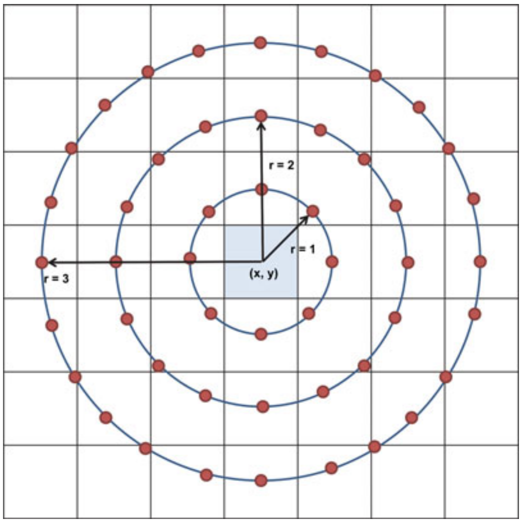


Fig. 2. Circular neighborhoods of pixel (*x, y*) at radius $r = 1$, 2, and 3. The neighboring pixel values are bilinearly interpolated.

the Holder exponent on x for a predefined measure μ defined on a compact set Ω, if

$$\forall x \in \Omega, \exists \alpha(x), \text{ such that } \mu\left(B_r(x)\right) \sim r^{\alpha}, \text{ for small } r \quad (1)$$

where $B_r(x)$ is the ball of radius r centered on x.

They proposed different predefined measures for the computation of Holder exponents through which different purposes can be accomplished. Chakraborty *et al.* [16] have proposed a new predefined measure for the segmentation of high-resolution satellite images. They have estimated the intensity value for each radius using linear regression analysis from the logarithmic plot of neighboring pixel values against radius. The difference between the estimated intensity value and the actual neighboring pixel intensity values for each radius is used as the predefined measure. However, the same method cannot be applied to PCO images, as linear regression analysis requires a data model that is linear in the model coefficients that does not hold for PCO images. Hence, we used the measure of dispersion of neighboring pixel intensity values on a particular ring from the mean value as the predefined measure.

For a two-dimensional (2-D) image, let I(*x, y*) represent the intensity value of the pixel (*x, y*) in the image. For each pixel (*x, y*), *R* number of concentric rings of radius r, where r varies from $r_{\min}$ to $r_{\max}$, are considered. The predefined measure μ_{rk} is the amount of dispersion of kth interpolated neighboring pixel value from the mean intensity value on a ring of radius r and is given by

$$\mu_{rk} = |P_{rk} - M_r| \quad (2)$$

where

P_{rk} is the kth interpolated neighboring pixel intensity value on a ring of radius r,

M_r is the mean intensity value of the interpolated neighboring pixels on a ring of radius r, and

μ_{rk} is the predefined measure.

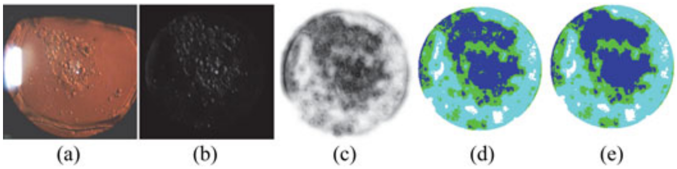


Fig. 3. Steps of proposed method: (a) original PCO image, (b) preprocessed image, (c) complement of roughness image, (d) segmented image using Otsu method, and (e) final segmented image after filling holes. (d) and (e) Blue, green, cyan, and white colors, respectively, indicate severe, moderate, mild, and clear areas of PCO. Colors are best seen in the softcopy version.

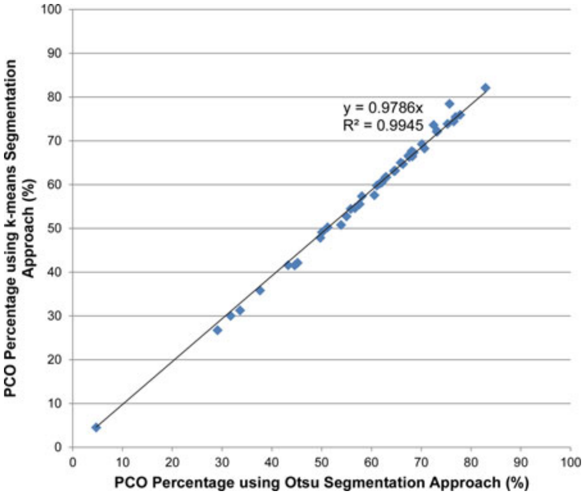


Fig. 4. PCO percentage values obtained using Otsu and k-means segmentation methods shown with line of equality.

The multiscale roughness value α of a pixel (*x, y*) in the image is defined as

$$\alpha = \frac{1}{N} \sum_{r=r_{\min}}^{r_{\max}} \sum_{k=1}^{n_r} \frac{\log \mu_{rk}}{\log r} \quad (3)$$

where

n_r is the number of interpolated neighboring pixels on a ring of radius r and

N is the total number of interpolated neighboring pixels.

The choice of radius r used in the calculation of α is very important. α calculated over large r captures the roughness of large areas, but the details are neglected. When roughness values are evaluated using small r, the interiors of the huge pearls could not be detected as severe PCO.

The roughness image S is obtained from the point-wise roughness values by normalizing the values to the range 0–255 using (4)

$$S(x, y) = \frac{\alpha(x, y) - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} \times 255 \quad (4)$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum and maximum multiscale roughness values, respectively.

C. Classification

Once the original image has been transformed into roughness image, the histogram-based thresholding method proposed by

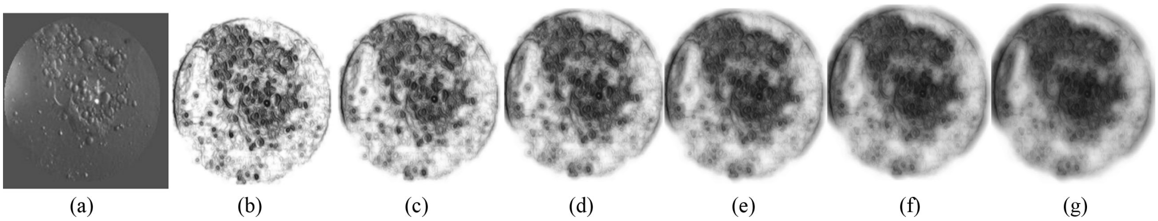


Fig. 5. Roughness images computed using proposed method with different multiscale radius values. (a) PCO gray-scale image with region of interest selected (b) $r = 1$–3, (c) $r = 1$–5, (d) $r = 1$–7, (e) $r = 1$–9, (f) $r = 1$–11, and (g) $r = 1$–13. For better visualization, the complement of roughness image is shown.

TABLE I
ASSESSING AGREEMENT BETWEEN EPCO METHOD AND PROPOSED METHOD FOR DIFFERENT MULTISCALE RADIUS VALUES USING BLAND–ALTMAN ANALYSIS

(%)	r = 1 to 3	r = 1 to 5	r = 1 to 7	r = 1 to 9	r = 1 to 11	r = 1 to 13
Bias (Mean)	10.67	4.583	2.8912	-2.2684	-4.8553	-7.0384
Upper Limit of Agreement (Mean + 2*Standard Deviation)	27.4203	20.1871	18.9912	13.5313	10.4388	8.6649
Lower Limit of Agreement (Mean – 2*Standard Deviation)	-6.0803	-11.0211	-13.2088	-18.068	-20.1495	-22.7417

Otsu [17] is used to segment the roughness image into four clusters of different PCO severity as clear (grade 0), mild (grade 1), moderate (grade 2), and severe (grade 3). The advantage of this method is that optimal thresholds are found automatically from the histogram by minimizing the intraclass variance of the different classes. However, it is possible that the interiors of some pearls may still not be classified as severe PCO because of smooth texture, which is quite similar to clear areas. But since the borders of the pearls tend to be rough, it is highly probable that the pearl border pixels are classified as severe PCO. The binary image corresponding to severe PCO pixels is obtained and morphological opening is applied on this binary image along with a set of logical operations to fill the small holes in the image that correspond to the pearl interior areas. The clusters are then updated accordingly.

PCO score is calculated from the number of pixels falling in each cluster of the region of interest, multiplied by the severity grade of that cluster

$$\text{PCO Score} = \frac{\sum_{i=0}^{3} p_i * i}{\sum_{i=0}^{3} p_i} \tag{5}$$

where p_i is the pixel count with a grade i.

Since grade 3 is the highest grade, the PCO percentage is obtained by multiplying the PCO score obtained in (5) with 100/3.

The intermediate results obtained at different steps of the proposed method are shown in Fig. 3.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A series of experiments were conducted using 43 real PCO images. The images are obtained using digital retroillumination imaging system. The patient is asked to place chin on a chin rest and look at the illumination light. The technician centers the recording optics on the posterior capsule under direct vision and acquires the image. The image is inspected and stored in jpeg format. The images we used in our experiments are the raw images initially saved by the technician without undergoing any quality enhancement processing.

To evaluate the results from the proposed method, they are compared with the following methods.

1) Clinical grading—This is a subjective and discrete grading performed by clinicians with grades given as 1, 2, 3, or 4, depending on the severity of PCO. The clinician staff involved in the evaluation comprises of three practicing consultants having more than 8 years of postqualification experience and two senior consultants having more than 25 years of experience. The scores related to the 43 PCO images were obtained as follows. The grading was performed first by the three consultants in a double-blind manner, meaning the grader neither knows the grades given by other colleagues, nor the grades obtained in our approach. The grading was performed in identical time and space circumstances. Images that received grades with disparities greater than 1 are examined by the two senior consultants.
2) EPCO Software—This is a free software that is available for download at [18] and has been used as a standard in many PCO clinical comparison studies. This software allows classification of PCO image into five areas of different PCO severity and the evaluated PCO scores are in the range 0–4.
3) Global entropy method—AQUA software [8], which is based on this method is also used in many clinical PCO studies. The global entropy value of each preprocessed image is computed from the GLCM of the preprocessed image using (6) as defined in [19]

$$H_g = -\sum_{i}\sum_{j} p\left(i,j\right) \log\left(p\left(i,j\right)\right) \tag{6}$$

where $p\left(i,j\right)$ is the *(i, j)*th entry in a normalized GLCM.
4) Local entropy filtering method—The local entropy filter as defined in [20] is applied on the preprocessed image to evaluate irregularity around each pixel. The local entropy of the 2 m+1 by $2n$+1 neighborhood is computed

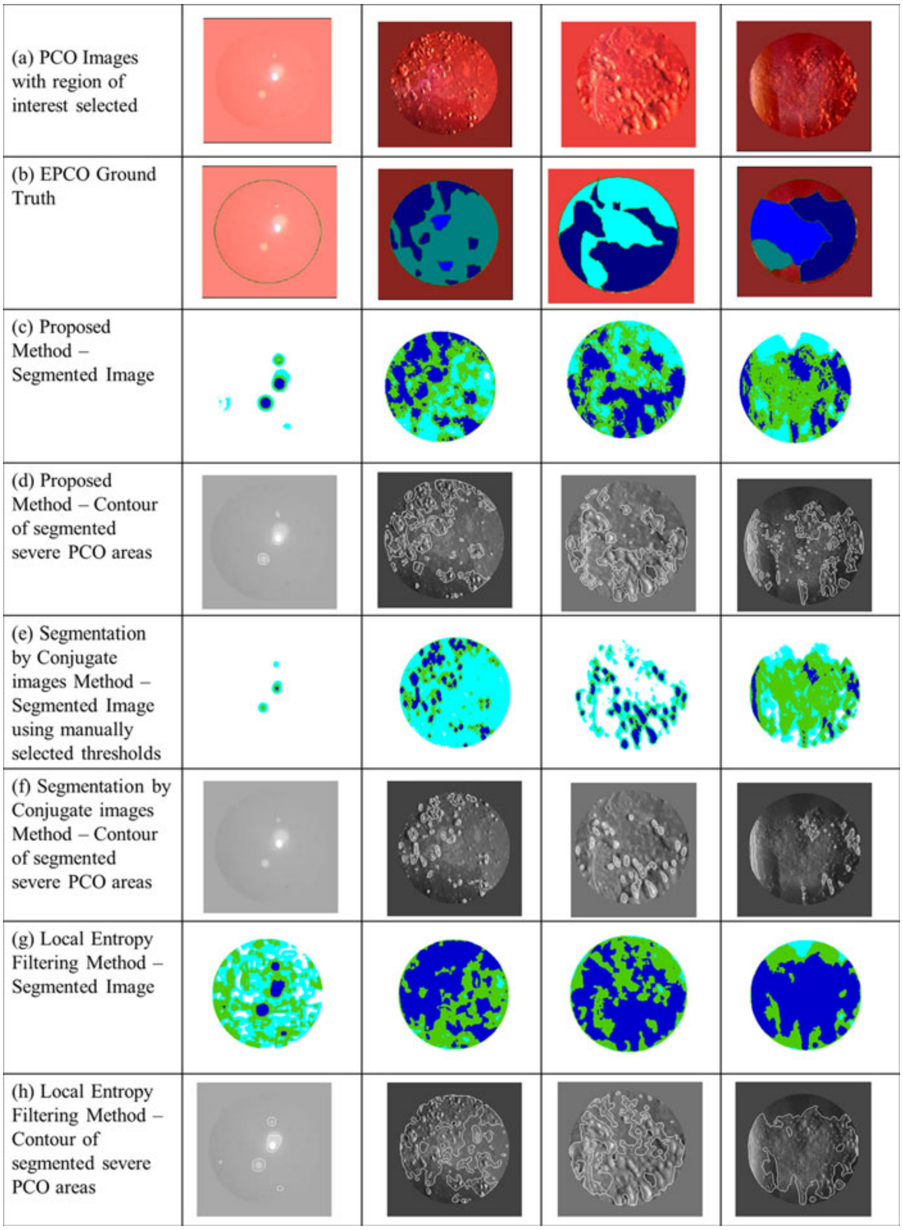

(a) PCO Images with region of interest selected
(b) EPCO Ground Truth
(c) Proposed Method – Segmented Image
(d) Proposed Method – Contour of segmented severe PCO areas
(e) Segmentation by Conjugate images Method – Segmented Image using manually selected thresholds
(f) Segmentation by Conjugate images Method – Contour of segmented severe PCO areas
(g) Local Entropy Filtering Method – Segmented Image
(h) Local Entropy Filtering Method – Contour of segmented severe PCO areas

Fig. 6. Comparison of segmented results of proposed method, segmentation by conjugate images method and local entropy filtering method with EPCO ground truth. (b) Transparent, cyan, turquoise, light blue, and dark blue, respectively, indicate [0–4] regions of PCO severity. (c), (e), and (g) White, cyan, green, and blue, respectively, indicate clear, mild, moderate, and severe PCO areas. Colors are best seen in the softcopy version.

TABLE II
PEARSON CORRELATION COEFFICIENTS TO MEASURE THE CORRELATION BETWEEN THE DIFFERENT PCO ASSESSMENT METHODS

vs.	EPCO Method	Global Entropy Method	Number of Regions Method	Local Entropy Filtering Method	Proposed Method
Clinical Grading	0.8585 $p=1.8412*10^{-13}$	0.7887 $p=3.3575*10^{-10}$	0.6176 $p=1.0229*10^{-5}$	0.8109 $p=4.3449*10^{-11}$	0.8460 $p=9.2104*10^{-13}$
EPCO Method		0.8603 $p=1.4370*10^{-13}$	0.6771 $p=6.18*10^{-7}$	0.9107 $p=2.4392*10^{-17}$	0.9393 $p=1.1913*10^{-20}$
Global Entropy Method			0.6226 $p=8.2627*10^{-6}$	0.9171 $p=5.6947*10^{-18}$	0.9271 $p=4.4962*10^{-19}$
Number of Regions Method				0.6574 $p=1.6786*10^{-6}$	0.6931 $p=2.583*10^{-7}$
Local Entropy Filtering Method					0.9499 $p=2.586*10^{-22}$

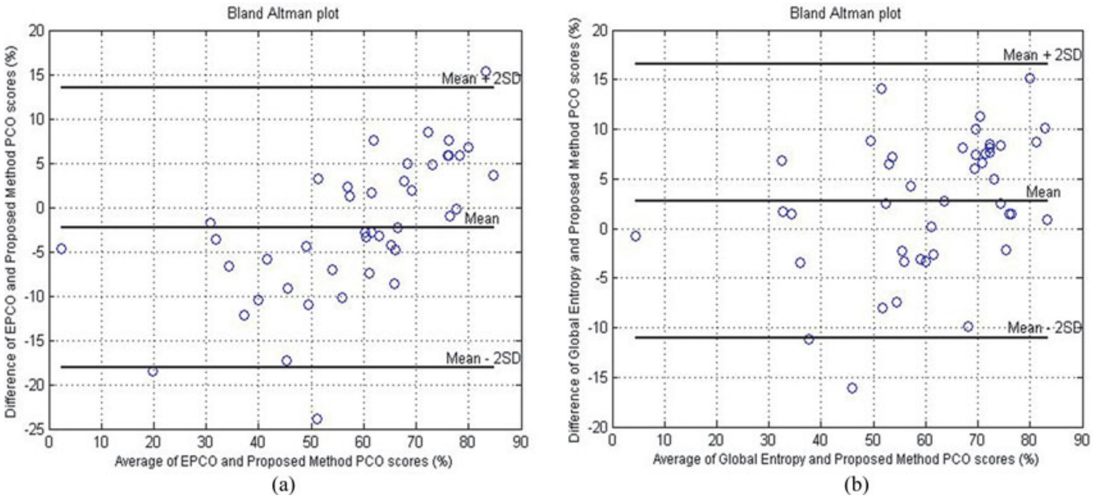


Fig. 7. Bland Altman plots to assess the agreement between (a) EPCO and proposed method and (b) global entropy and proposed method.

using (7)

$$H(i,j) = -\sum_{u=i-m}^{i+m} \sum_{v=j-n}^{j+n} p(I(u,v)) \log(p(I(u,v)) \quad (7)$$

where $I(u,v)$ is the intensity of the pixel (u,v) in the image. $p(I)$ is the probability mass function of the image intensity within the local window. The chosen window size is 9 × 9.

5) Number of regions method—PCO score is derived from the number of regions computed in the image as proposed in [11].

6) Segmentation by conjugate images—This texture-based segmentation method, which is proposed by Paplinski and Boyce [4], uses the GLCM derived from conjugate images. However, this method does not yield automatic thresholds and requires manual setting of thresholds. Hence, it is only used for the sake of visual comparison of segmented results where optimum thresholds are chosen manually. Moreover, it is observed that the best results are obtained when the images are subjected to the preprocessing stage in Section III-A where background illumination is eliminated.

The following aspects have been checked using the experiments:

A. *Choice of Segmentation Method*

The roughness image obtained in Section III-B is segmented into four clusters using both k-means and Otsu methods and PCO percentage values of the 43 images obtained are compared. The results showed a high correlation of 97.86% as shown in Fig. 4; hence, any of these methods can be used for the sake of segmentation. In the following sections, we have used Otsu method for the segmentation purpose. The mean shift method is not tested because of the complexity of calculation and also it offers surplus regions. Also the results of mean shift segmentation depend on the size of the window.

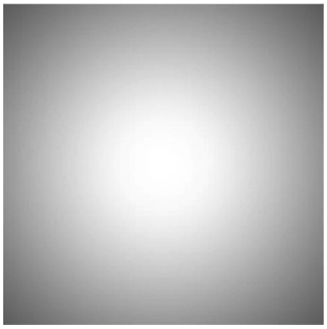


Fig. 8. Artificial illumination pattern.

B. *Optimum Value of R (Number of Concentric Rings) to Compute Roughness Image*

We have tested the proposed method for different R with $r_{\min} = 1$ and $r_{\max} = 3, 5, 7, 9, 11,$ and 13. The roughness images for different values of R are demonstrated in Fig. 5. It can be observed that as the number of scales increases, the vicinity of irregularity becomes wider; however, less scales result in a noisy image.

The measure of agreement between the computed PCO percentage values for different R and EPCO method is studied using Bland–Altman analysis by plotting the difference between the scores from the two methods against the mean value of the PCO scores. The biases, upper and lower limits of agreement are shown in Table I. Based upon these results, $r_{\max} = 9$ *is* chosen as the optimum maximum radius to compute the roughness image.

C. *Visual Correlation of Segmented Results With Respect to Original Image and Other Existing Methods*

Segmented results obtained using EPCO, proposed method, Segmentation by conjugate images method and Local entropy filtering method are shown in Fig. 6. Since severe PCO areas have got significant weightage in the PCO percentage, the contour of the detected severe PCO areas using the different

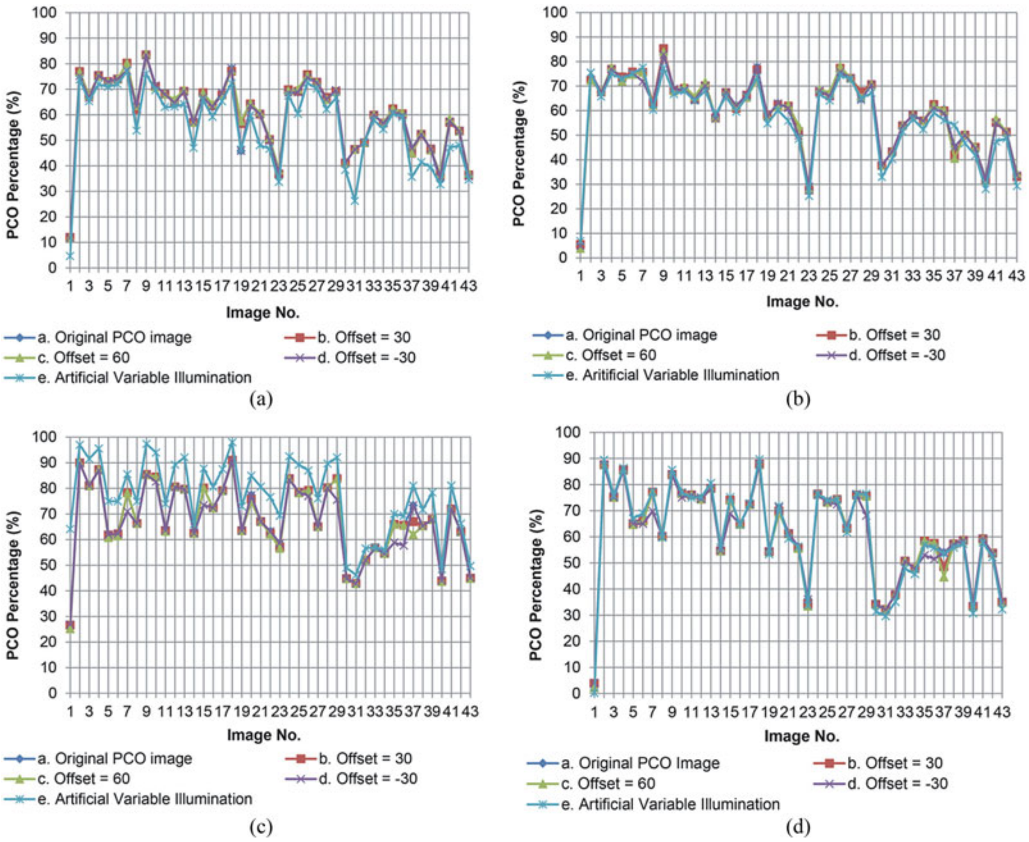


Fig. 9. Robustness of the proposed method and global entropy method against illumination changes, (a) proposed method without the elimination of background illumination step, (b) proposed method with the elimination of background illumination step, (c) global entropy method without the elimination of background illumination step, and (d) global entropy method with the elimination of background illumination step.

methods is also shown in Fig. 6. The first image in Fig. 6(a) shows a clear eye without PCO, and thereby the segmented image using the proposed method shows the maximum portion of the capsule in white indicating clear eye except for reflection artifacts, which gives a PCO percentage of about 4.7%. It should be noted here that the main aim of this research work is to find a method, which could effectively detect the PCO and clear areas. The problem of reflection artifacts is not dealt here; though, it could be addressed by capturing multiple images of posterior capsule in different directions of gaze as discussed by Findl *et al*. in [21]. From the results, it is clear that the proposed method not only identifies the large pearls as severe PCO but effectively classifies the PCO areas into different severity based upon the roughness. Local entropy filtering method results in oversegmentation and hence provides high PCO scores even for mild cases of PCO as shown in Fig. 6(g). Segmentation by conjugate images method requires the optimum thresholds to be selected manually.

D. Agreement of Proposed Method With the Clinician's Assessment and Other Methods

In order to assess the validity of PCO percentage values estimated by the proposed method, the results are compared with clinicians' grading and other existing methods. PCO scores from all the methods except Segmentation by Conjugate images method are compared using Pearson correlation analysis. This is the standard tool used for the evaluation of correlation between different assessment methods. The Pearson coefficient ranges from -1.0 to $+1.0$, where $+1.0$ indicates a strong positive relation, -1.0 indicates a strong negative relation, and 0 indicates no relation. To assess the statistical significance of the correlation, the p-values are also computed along with Pearson coefficient. If the p-value is low (<0.05), then the correlation is statistically significant and the calculated Pearson coefficient can be used for analysis. The Pearson correlation coefficients of all the methods and their corresponding p values are presented in Table II. EPCO and Global entropy methods are widely used in PCO assessment and prevention research studies [22]–[24]. In order to assess the agreement between the proposed method and these methods, we have used Bland–Altman analysis, where the differences between the PCO scores from two methods are plotted against the averages of the PCO scores. From the biases, upper, and lower limits of agreement indicated in Fig. 7, it can be seen that the proposed method shows good agreement with these methods. Since clinical grading range is crisp and discrete, Bland–Altman analysis between clinical grading and proposed method is not shown.

The results presented in Table II demonstrate that the proposed method shows better correlation with clinical grading compared to global entropy, number of regions, and local entropy filtering methods. Though EPCO method exhibited a little higher correlation with clinical grading compared to the proposed method, it is to be noted that the proposed method is a fully automated technique. EPCO system is manually operated, which is time-consuming and provides biased results depending on the operator experience and illumination conditions under which the images are acquired. Though global entropy method provides an objective and quantitative score, it does not provide the morphological details. Hence, the proposed method is better than the existing methods of PCO quantification and could become a standardized grading system for communication between clinicians.

E. Robustness of Proposed Method Against Changes in Illumination

When estimating the roughness around a pixel the measure of dispersion at a particular ring is calculated with respect to the mean value of the neighboring pixels. So for any monotonous changes in illumination of the image, the value remains stable. To understand the robustness of the method to illumination, the images are brightened by adding an offset to pixel intensity values and the resultant sum is fixed to be within the range 0–255. The offsets that are tested are 30, 60, and -30. These result in application of monotonic illumination across the image. In order to check the robustness of the proposed method with respect to variable illumination, we have created an artificial illumination pattern as shown in Fig. 8. The original images are multiplied with this artificial illumination pattern to result in variable illumination across the image, apart from the uneven background illumination inherent of the image.

A comparison of PCO percentages of the original images and the illuminated images is shown in Fig. 9(a) and (b). The results are demonstrated with and without the elimination of background illumination step mentioned in Section III-A. For comparison, the illumination invariance of the global entropy method is also tested against different illumination changes and presented in Fig. 9(c) and (d). From Fig. 9(a), it can be observed that the proposed method is stable against monotonic illumination variations but less stable in case of uneven illumination changes across the image. Fig. 9(c) demonstrates the stability of global entropy method with monotonic illumination variations but highly sensitive nature of this method with variable illumination across the image. Fig. 9(b) and (d) demonstrates that including the elimination of background illumination step in the preprocessing stage results in more stable PCO percentage values.

V. Conclusion

We proposed a novel method for PCO assessment introducing the multiscale roughness concept. The PCO scores from the proposed method compared well with that of clinicians' grading and other widely used methods like EPCO and global entropy. The correlation between the proposed method and clinicians' grading is 84.6%, which is high, considering the fact that the clinicians employ a highly subjective grading range. In addition, this method has addressed some key issues of PCO assessment such as complete automatic detection and quantification of PCO areas, ability to identify mild cases of PCO, and robustness against illumination variations across the image.

The proposed software can be easily installed on any personal computer running MATLAB. The system just requires digital photographs obtained from the retroillumination imaging system, which is widely available in the majority of ophthalmological departments. As the grading process is completely automated, this method could be adopted as a standardized grading system for communication between clinicians. The objectivity, reproducibility and reliability of the grading of PCO by this method will help in effective referral of patients from peripheral and remote centers to PCO expertise centers for treatment. Availability of morphological data along with the PCO percentage will ensure proper identification of patients who need an effective treatment for PCO. The proposed method is implemented in MATLAB on Windows 7 machine with Intel Xeon E5-1607 processor. The average time required for the computation of PCO score of an image is 2.9 s, which permits the clinician to conduct the PCO assessment in an acceptable interactive time. Efficiency can be further improved with a full C++ language implementation and can eventually be significantly enhanced with a hardware implementation.

Since the proposed method is roughness based, it could not effectively detect some regions in extreme PCO pearl cases exhibiting very large pearl uniform areas. Another limitation with this approach is that this method cannot handle images that are severely corrupted by light artifacts. However, such images can be rejected during the image acquisition. We plan to extend the research to address these limitations as well as the detection of type of PCO and computation of severity of PCO based on its presence from the visual axis.

Acknowledgment

The authors would like to thank Dr. T. Aslam from the Manchester Eye Hospital, Dr. P. Bhatia, Dr. A. Rao, Dr. P. Warhekar, Dr. K.H. Sathish, and Dr. B.S. Chidamber from the Ophthalmology Department at Welcare Hospital, for their kind collaboration.

References

[1] S. K. Powell and R. J. Olson, "Incidence of retinal detachment after cataract surgery and neodymium: YAG laser capsulotomy," *J. Cataract Refractive Surg.*, vol. 21, no. 2, pp. 132–135, Mar. 1995.

[2] M. R. Tetz, G. U. Auffrath, M. Speaker, M. Blum, and H. E. Volcker, "Photographic image analysis system of posterior capsule opacification," *J. Cataract Refractive Surg.*, vol. 23, no. 10, pp. 1515–1520, Dec. 1997.

[3] L. Bender, D. J. Spalton, B. Uyanonvara, J. Boyce, C. Heatley, R. Jose, and J. Khan, "POCOman: New system for quantifying posterior capsule opacification," *J. Cataract Refractive Surg.*, vol. 30, no. 10, pp. 2058–2063, Oct. 2004.

[4] A. P. Paplinski and J. F. Boyce, "Computational aspects of segmentation of a class of medical images using the concept of conjugate images," Monash Univ., Clayton, Vic., Australia, Tech. Rep., 1995.

[5] S. A. Barman, E. J. Hollick, J. F. Boyce, D.J. Spalton, B. Uyyanonvara, G. Sanguinetti, and W. Meacock, "Quantification of posterior capsular opacification in digital images after cataract surgery," *Investig. Ophthalmol. Visual Sci.*, vol. 41, no. 12, pp. 3882–3892, Nov. 2000.

[6] M. C. Wang and L. C. Woung, "Digital retroilluminated photography to analyze posterior capsule opacification in eyes with intraocular lenses," *J. Cataract Refract. Surg.*, vol. 26, no. 1, pp. 56–61, Jan. 2000.

[7] H. Siegl, "Quantification of posterior capsule opacification after cataract surgery," M.S. thesis, Graz Univ. Technol., Austria, 2000.

[8] H. Siegl, A. Pinz, W. Buhl, M. Georgopoulos, O. Findl, and R. Menapace, "Assessment of posterior capsule opacification after cataract surgery," in *Proc. 12th Scandinavian Conf. Image Anal.*, Bergen, 2001, pp. 54–61.

[9] T. M. Aslam, N. Patton, and C. J. Rose, "OSCA: A comprehensive open-access system of analysis of posterior capsular opacification," *BMC Ophthalmol.*, vol. 6, pp. 1–6, Aug. 2006.

[10] T. M. Aslam, N. Patton, and J. Graham, "A freely accessible, evidence based, objective system of analysis of posterior capsular opacification: Evidence for its validity and reliability," *BMC Ophthalmol.*, vol. 5, pp. 1–10, Apr. 2005.

[11] N. Werghi, R. Sammouda, and F. AlKirbi, "An unsupervised learning approach based on Hopfield-like network for assessing posterior capsule opacification," *Pattern Anal. Appl.*, vol. 13, no. 4, pp. 383–396, Nov. 2007.

[12] D. Grewal, R. Jain, G. S. Brar, and S. P. S. Grewal, "Pentacam tomograms: A novel method for quantification of posterior capsule opacification," *Investig. Ophthalmol. Visual Sci.*, vol. 49, no. 5, pp. 2004–2008, May 2008.

[13] Y. Tanaka, S. Kato, K. Miyata, M. Honbo, R. Nejima, S. Kitano, S. Amano, and T. Oshika, "Limitation of Scheimpflug video photography system in quantifying posterior capsule opacification after intraocular lens implantation," *Amer. J. Ophthalmol.*, vol. 137, no. 4, pp. 732–735, Apr. 2004.

[14] T. M. Aslam, B. Dhillon, N. Werghi, A. Taguri, and A. Wadood, "Systems of analysis of posterior capsule opacification," *Brit. J. Ophthalmol.*, vol. 86, no. 10, pp. 1181–1186, 2002.

[15] Y. Liu and Y. Li, "New approaches of multifractal image analysis," in *Proc. Int. Conf. Inform., Commun. Signal Process.*, 1997, Singapore, pp. 970–974.

[16] D. Chakraborty, G. K. Sen, and S. Hazra, "High-resolution satellite image segmentation using Hölder exponents," *J. Earth Syst. Sci.*, vol. 118, no. 5, pp. 609–617, Oct. 2009.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[18] M. R. Tetz. (2013, Jun.). *EPCO2000: Software for the evaluation of posterior capsule opacification* [Online]. Available: www.epco2000.de

[19] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[20] X. Gao, H. Li, J. H. Lim, and T. Y. Wong, "Computer-aided cataract detection using enhanced texture features on retro-illumination lens images," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 1565–1568.

[21] O. Findl, W. Buehl, H. Siegl, and A. Pinz, "Removal of reflections in the photographic assessment of PCO by fusion of digital retroillumination images," *Investig. Ophthalmol. Visual Sci.*, vol. 44, no. 1, pp. 275–280, Jan. 2003.

[22] M. Vetrugno, F. Masselli, G. Greco, D. Sisto, A. Maino, S. Ficarelli, and G. Sborgia, "The influence of posterior capsule opacification on scanning laser polarimetry," *Eye*, vol. 21, no. 6, pp. 760–763, Jun. 2007.

[23] R. Zemaitiene, V. Jasinskas, V. Barzdziukas, and G. U. Auffarth, "Prevention of posterior capsule opacification using different intraocular lenses (results of one-year clinical study)," *Medicina (Kaunas)*, vol. 40, no. 8, pp. 721–730, 2004.

[24] L. Vock, R. Menapace, E. Stifter, M. Georgopoulos, S. Sacu, and W. Buhl, "Posterior capsule opacification and neodymium: YAG laser capsulotomy rates with a round-edged silicone and a sharp-edged hydrophobic acrylic intraocular lens 10 years after surgery," *J. Cataract Refractive Surg.*, vol. 35, no. 3, pp. 459–465, Mar. 2009.

Aruna Vivekanand received the B.Tech. degree in electronics and communications from Jawaharlal Nehru Technological University, Hyderabad, India, in 2003, the M.Tech. degree in communication systems from Indian Institute of Technology Madras, Chennai, India, in 2006, and the M.Sc. degree in information technology-mobile communications from Heriot Watt University, Edinburgh, UK, in 2011.

From 2006 to 2008, she worked as Assistant Systems Engineer at Tata Consultancy Services Ltd., Mumbai, India. She was a Lecturer in the Department of Information Technology, Thakur College of Engineering and Technology, Mumbai, India, from 2009 to 2010, and Emirates College of Management and Information Technology, Dubai, in 2012. Since 2013, she has been a Research Assistant at Khalifa University of Science, Technology and Research, Sharjah, UAE. Her current research interests include medical image analysis and development of computer aided diagnostic procedures.

Naoufel Werghi (M'00) received the Ph.D. degree in computer vision from the University of Strasbourg, Strasbourg, France, and the M.Sc. degree in instrumentation and control from the University of Rouen, Mont-Saint-Aignan, France.

He was a Research Fellow at the Division of Informatics, University of Edinburgh; a Lecturer in the Department of Computer Sciences, University of Glasgow; an Assistant Professor in the College of Information technology, University of Dubai, where he founded and headed the Intelligent Information System Research Group. He was a Visiting Professor in the Department of Electrical and Computer Engineering, University of Louisville. He is currently an Associate Professor in the Electrical and Computer Engineering Department, Khalifa University of Science, Technology and Research, Sharjah, UAE. His current research interests include image analysis and interpretation. He has been leading several funded projects in the areas of biometrics, medical imaging geometrical reverse engineering, and intelligent systems. He has authored or coauthored more than 80 journal and conference papers.

Dr. Naoufel received the Best Paper Award in the International Conference of Computer Vision, Theory and Applications 2011, and the University-Industry Research Award from the National Research Foundation, UAE in 2011 and 2013. In 2012, he cofounded the Emirates Computational Intelligence and Vision Research Group. He is currently the Secretary Treasurer of the IEEE computer chapter in UAE.

Hussain Al-Ahmad (S'78–M'83–SM'90) was born in Iraq. He received the B.Sc. degree in electrical engineering from the University of Basra, Basra, Iraq, in 1976, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Leeds, Leeds, UK, in 1979 and 1984, respectively.

He was with Portsmouth University, Leeds Metropolitan University, Kuwaiti Faculty of Technological Studies, University of Bradford, and Etisalat University College. He is currently a Full Professor of electronic engineering at Khalifa University of Science and Technology, Sharjah, UAE. He is a Member of the VSAP Research Lab. He is the author or coauthor of more than 90 journal articles and referred conference papers. His current research interests include signal and image processing.

Prof. Al-Ahmad is a Fellow of the IET, a Chartered Engineer, a Member of BCS, a Chartered IT Professional, and a Fellow of the Royal Photographic Society. He is the Vice Chair of the IEEE UAE section. He was the Secretary of the IEEE Kuwaiti section, the Chair of the IEEE UAE computer chapter, and the Vice Chair of the IEEE UAE signal processing and communication chapter. He was a Member of the Technical Program Committees of many IEEE conferences such as ICECS, ICSPC, ISSPIT, and ICIP.

CONVOLUTIONAL NEURAL NETWORK AS A FEATURE EXTRACTOR FOR AUTOMATIC POLYP DETECTION

Bilal Taha, Jorge Dias, Naoufel Werghi

Department of Electrical and Computer Engineering
Khalifa University, Abu Dhabi, UAE

ABSTRACT

Colorectal cancer is one of the major causes of cancer deaths worldwide. To achieve early cancer screening, detecting the presence of polyps in the colon tract is the preferred technique. In this paper, a deep learning approach for identifying polyps in colonoscopy images is proposed. The novelty of our technique stems from the fact that it fully employs a pre-trained Convolutional Neural Network (CNN) architecture as a feature extractor. Contrary to the conventional methods which either perform fine-tuning or train the CNN from scratch, we utilize the CNN output features as an input to train the Support Vector Machine (SVM) Classifier. The efficiency of the presented framework is demonstrated on the public CVC ColonDB, in which the experimental results indicate that our methodology significantly outperforms other competitive paradigms.

***Index Terms*—** Automatic polyp detection, Deep learning, CNN, feature extractor.

1. INTRODUCTION

According to the Centers for Disease Control and Prevention (CDC) in the United States, colorectal cancer (CRC) is the third most common cancer in the world [1]. Colorectal cancer starts with small protrusions growing inside the colorectal which could eventually lead to CRC [1]. These protrusions are known as polyps. Fig. 1 shows examples of polyps with different shapes and appearances. It is the ability to detect polyps and remove them in early stages that saves more lives and results in better prevention of CRC [2]. The most common method for this process is by visual inspection using endoscopic videos. However, clinical examination is not sufficient enough as a final judgment since there are many sources of error and false diagnosis. These sources of errors could be correlated with the medical level of expertise and the nature and appearance of the polyp itself. Indeed, the variety of shapes and sizes in which the polyps appear, and the limited field of view inside the colon, makes it difficult to the clinical examiner to keep continuous and consistent evaluations on detecting the polyps and support diagnosis. Thus, its turned out essential to develop an automated system to support the physician to detect and classify polyps. Several computer aided diagnosis (CAD) systems have developed to that end. Here we report the main works and we refer the reader to [3] for a more exhaustive survey. Some authors proposed to use texture features [4, 5], shape features [6, 7] or feature fusion [8, 9] coupled with standard classifiers. However, these methods still suffer from a high false positive rate. In addition, defining optimal descriptors proper for polyp presentation seemed to be quite complex and dependent on the correct tuning of multiple parameters. To bypass this problem, there has been a recent trend to use deep learning approaches to benefit from its powerful feature learning capacity [10]. The authors of [11] have used and trained the Convolutional Neural Network (CNN) from scratch to outperform the existing methods. However, this approach requires a huge database for the CNN to learn features from the images. In a subsequent work the same authors in [12] implemented a fine tune CNN approach demonstrating a competitive performance when compared to training a CNN from scratch. The latter work represents a neat advance in terms of efficient CNN usage. Indeed, training the CNN from scratch while it creates abstract features more related to the database, is time and computation demanding, and requires a huge labeled database which might be impractical in real applications and very costly. Fine tuning, while reducing the size of the data needed, still requires substantial data training to get the parameters accurately tuned to a specific database.

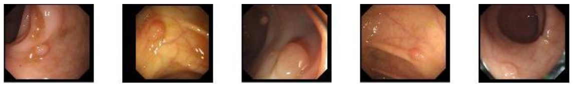


Fig. 1: Polyp samples from CVC-ColonDB database

In this work, we focus more on the CNN deployment efficiency aspect, and propose a method that capitalizes on the capacity of well-established convolution neural network architecture for producing generic features, which can be tailored for a specific classification application. The key contributions in this work are (1) Efficient employment of CNN architecture without the need for training from scratch; (2) Accommodating partial polyp appearances in the colonoscopy images; (3) Finding middle-layer features of the computational architecture that can be more effective than the end-layer features, as would be expected; (4) Evaluating and showing the superiority of our approach when compared with competitive methods.

The remainder of the paper will be organized as follows: Section 2 introduces the proposed approach and its rational. Section 3 describes the different experiments and the related results. Section 4 concludes the paper.

2. THE PROPOSED METHOD

In image analysis, designing the appropriate features for a given interpretation task has been a central problem in computer vision and medical image understanding. Explicit feature design extraction for medical images require subject-matter expertise. In this process, the visual information on which the physician relies in his assessment is not necessarily reflected into a suitable computational representation. Moreover, the practical considerations in the extraction and the usage of these features make the reproducibility of such related methods often problematic [13]. To overcome these challenges, we propose a deep learning approach, whereby Convolution Neural Network (CNN) is employed to replace hand-crafted and customized features that are often strongly sensitive to multiple parameters.

It is known that through a hierarchical unsupervised or semi-supervised feature design, CNN's can produce effective representation of the visual data [10]. Basically, a CNN architecture is composed of a sequence of cascading layers performing basic operations such as convolution, subsampling, followed by another sequence of fully connected layers, which act similarly as a classic artificial neural network.

In another hand, training a CNN network from scratch requires a large dataset. Such a process is quite tedious, in addition large datasets cannot be afforded easily in medical applications, including dataset of polyp detection. Apart of the computational resources requirements, there is no systematic guidelines as for the optimal choice of the architecture in terms of depth (number of layers) and structure.

An economic alternative is to use pre-trained CNN architectures, that are proven to have good performance through training and validation over a huge database, and then tune, via training conducted on a specific application dataset, the pre-trained weights of the architecture. This procedure, known as fine-tuning, which can be performed either across the whole CNN or at specific layers.

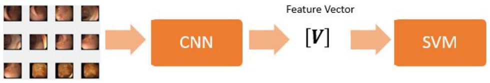


Fig. 2: Block diagram for the CNN as a feature extractor

Table 1: Summary of AlexNet architecture

Layer	Type	Input	Kernel	Stride	Pad	Output
Data	Input image	227x227x3	N/A	N/A	N/A	227x227x3
conv1	Conv	227x227x3	11x11	4	0	96x55x55
pool1	Max pooling	55x55x96	3x3	2	0	96x27x27
conv2	Conv	27x27x96	5x5	1	2	256x27x27
pool2	Max pooling	27x27x256	3x3	2	0	256x13x13
conv3	Conv	13x13x256	3x3	1	1	384x13x13
conv4	Conv	13x13x384	3x3	1	1	384x13x13
conv5	Conv	13x13x384	3x3	1	1	256x13x13
pool5	Max pooling	13x13x256	3x3	2	0	256x6x6
FC6	fully connected	6x6x256	6x6	1	0	4096x1
FC7	fully connected	1x4096	1x1	1	0	4096x1
FC8	fully connected	1x4096	1x1	1	0	1000x1

IIn our approach, we advocate the hypothesis that a trained CNN architecture embeds sufficiently rich feature representations that can be utilized as an input to train a standard classifier, such as the Support Vector Machine(SVM), relieving thus the system from laborious training from scratch or fine tuning. Therefore a pre-trained CNN is then deployed as a feature extractor for our specific image interpretation task of polyp detection, as depicted in the block diagram in Fig.2. There are several pre-trained CNN architectures that can be investigated, such as GoogleNet [14] and VGGNet [15]. In our method, we explored AlexNet [16]. This CNN architecture was trained with 1.2 million images for 1000 different classes, thus the learned features are expected to span a large spectrum of visual information. The main layers of the AlexNet architecture is briefly described in Table 1. As aforementioned, features from first layers are too generic to be employed as discriminative descriptors, so we investigated features from the middle layers and onward, namely, Conv4 till FC8. The output from one of these layers will be a sort of feature encoding for a full (or partial) colonoscopy image. These features will be then fed into the subsequent classifier block, SVM, as depicted in Fig. 2. An SVM converges to a global and unique solution, and has the capacity to deal with a high-dimension input without compromising the computational complexity, and thus can map the huge number of feature vectors x_i, $i = 1...N$, generated by the CNN. When training an SVM, each feature vector is given a label either polyp or non-polyp (abnormal, normal) to create the feature-class pair $\{x, y\}$. Therefore, given L features $\{x_i, y_i\}$ such that $i = 1...L$, and $y_i \in \{1, -1\}$, $x \in \Re^D$, where D is the vector size. A hyper-plane separating the two classes could be written as

$$w^T x + b = 0 \tag{1}$$

the w is known as a weight vector which is normal to the separation hyper-plane, and b is a bias. In order to separate the two classes with a hyper-plane, equation (2) should be optimized

$$min(\frac{1}{2} w^T w + C \sum_{i=1}^{L} \xi_i) \tag{2}$$

subject to the constrain $y_i((w^T x_i) + b) >= 1 - \xi_i$, where $\xi_i >= 0$ for $i = 1, ..., L$, and C is the penalty parameter. This will lead to the optimal hyper-plane that minimizes the distance between itself and all the training examples. The optimal hyper-plane, allows a classification to be done according to a decision function such as:

$$f(x) = \text{sgn}(w^T x + b) \tag{3}$$

3. EXPERIMENTATION

To evaluate the performance of our method, CVC-ColonDB [17] was used for training and testing. This database consists of 15 short colonoscopy videos for different 15 cases. It includes different polyp sizes, appearances and colors. We conducted a series of extensive experiments on the specified database that aimed to assess the performance of the CNN as a feature extractor and its effectiveness in the detection scenarios. In these experiments we studied the effects of 1) Selecting features from different layers of the CNN, 2) The image patch size, and 3) The polyp appearance in each patch, that is the minimum portion of polyp area visible in a patch to be considered as genuine case (true positive).

In the first experiment, the main focus is on the quality of extracted features from the CNN. The features were employed from different layers from the pre-trained CNN. AlexNet was trained using the ImageNet database which consists of non-medical images, therefore there is a need to know the best layer that will provide the best features discriminating polyp from non-polyp cases. As we mentioned earlier, we considered only deep layers, starting from Conv4. In this transfer learning scheme, the layers up to the output features layers are frozen and the output features are used to train the SVM classifier. For example, considering Conv5, as the feature output layer, we keep the weights across the layers conv1 to conv5 at their pre-trained values, while training the SVM classifier. While this scheme reduces the number of trained entities, the number of features remain large, as an example, the dimension of the obtained feature vector from the fourth convolution layer (C4) is 13x13x384 which is equal to 64896 features. In this experiment, the patch size was fixed to 16 patches/image, each patch is 100x100, making a total of 4800 patches, and we considered any polyp appearance to be a positive case meaning no threshold as for the minimum size of its partial appearance. For training protocol, we adopted the 70%, 30% for training and testing, respectively. Fig.4.a depicts the obtained ROC curves related to the different output layers. For instance, C5 refers to the features coming out from the layer Conv5. Table 2 reports the best recall and precision performance obtained for each layer. It is interesting to notice that the top performance is obtained with features coming out from a middle layer (conv5), which are less descriptive than their deeper layered counterparts (e.g. FC8 - see Table 1).

In the second experiment, the effect of patch size on the performance of the CNN as a feature extractor is investigated. The patches are constructed by utilizing a sliding window vertically and horizontally without any overlapping, dividing thus the image into patches. The choice of optimal patch size is a bit problematic. While reducing the patch size increases the volume of samples used for training and testing, which is good for the over-fitting problem, it increases also the number of small partial polyps, and thus jeopardizing the ability to extract good features. On the other hand, the big patch size reduces the probability of partial polyps in each patch but, at the same time it reduces the size of the training samples. To address this issue, we investigated the optimal size empirically by experimenting three patch sizes 200x200, 100x100, and 50x50. Fig. 3. (a) depicts the three patch sizes on the polyp images, respectively. Fig. 4.b shows the ROC curves related to each patch size whereas the best recall and precision values for experiment 2 are reported in Table 2.

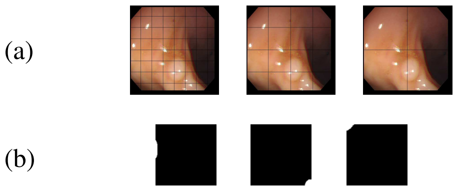


Fig. 3: (a) samples of different patch size, (b) masks corresponding to patch samples with small polyp portion.

In the third experiment, we studied the effect of the threshold on the polyp area portion for considering a patch as containing a polyp (positive sample) or not (negative sample). The motivation behind this experiment is that in practical situations, these small polyp parts are not noticeable by the physicians and thus should be considered rather than as a negative sample. The mask samples already reported in Fig. 3. (b) illustrate examples of odd small polyp portion in a patch. In this experiment, the patch size was fixed to 100x100 and the features were taken after the fifth convolution layer (C5). ROC curves obtained with different thresholds are depicted in Fig 4.c. We notice that the best ROC curve corresponds to a 7% threshold. This is also reflected in the recall and precision scores in Table 2.

Normally, one should test all the combinations of the three parameters (CNN layer, patch size and polyp portion threshold) to come out with a combination that achieves the best performance. However, performing such exhaustive procedure needs to conduct $5 \times 3 \times 4 = 60$ different training. As a less demanding alternative, though sub-optimal, we considered the best parameter in each of the previous three experiments (Conv5 layer, 50×50, 7%), then re-evaluated the performance of the system. We compared our method with six state of the art methods that used the same database [17, 18, 19, 20, 21, 22]. In [17] the authors proposed an algorithm based on the polyp distinct shape and used a segmentation algorithm, to minimize the number of most likely polyp. Then, they utilized Sector Accumulation-Depth of Valleys Accumulation (SA-DOVA) as a descriptor for the detection process. In their subsequent work [20], they have improved their methodolgy by focusing more on the preprocessing stage where they tackled the effect of specular lights and blood vessels. Furthermore, authors of [21] proposed a system to handle the imbalance size between the polyp and non-polyp samples. They have employed least-squares analysis to learn different types of features. In [18], the authors used Haar features, one layer of classification, and a voting method to detect polyps. Then, in [19] they implemented a two stage edge classification scheme to obtain a refined edge map and the direction of the normal for the polyp-like edges. Afterwards, a new voting scheme is applied to the refined edge map to localize polyps by detecting curvy boundaries. In a recent work, the authors in [22] augmented their previous shape-based approach with context-clues information derived around the polyp boundaries. Table 3 reports the performances of the seven methods. We found that our approach outperforms all the existing paradigms in terms of recall with a score of 96%, concurrently, illustrating a very close score in term of precision compared with the best value, where the difference is only 0.3%.

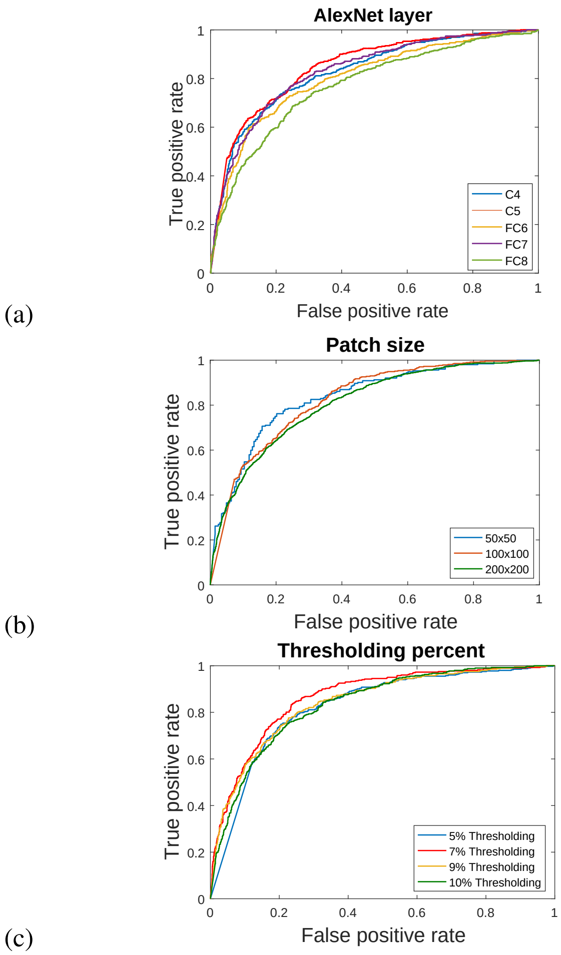


Fig. 4: ROC curves related to experiments 1(a), 2(b) and 3(c).

Table 2: The recall and precision values for each experiment

	Experiment 1		
		Recall	Precision
CNN feature	Conv4	74%	90.8%
	Conv5	**89.6%**	**86.3%**
	FC6	98.9%	76.5%
	FC7	99.1%	76.1%
	FC8	97.2%	77.9%
	Experiment 2		
patch size	200×200	79.8%	72.3%
	100 × 100	65.3%	91.4%
	50 × 50	**99.4%**	**85.2%**
	Experiment 3		
Threshold percent	5%	56.2%	95.4%
	7%	**96.3%**	**84.3%**
	9%	91.1%	87.4%
	10%	90.4%	86.8%

Table 3: Recall and precision scores in percent by setting the parameters according to the best results in each experiment compared to other paradigms.

Method	[17]	[18]	[21]	[20]	[19]	[22]	Ours
Recall	47.15	60	70.6	67.6	80	88	**96**
Precision	71.6	88	70.7	-	**93**	-	92.7

4. CONCLUSION

In this work, we have introduced a deep learning solution for detecting polyps from colonoscopy. The novel deployment of the AlexNet, a pre-trained architecture used as a feature extractor, along with a classical SVM classifier was proposed. By adopting this approach, the system circumvents the high computational complexity and high resource demand of CNN required in training from scratch and fine-tuning. The series of experiments conducted with the CVC colonDB database, confirmed the rationale behind our hypothesis, which implies that the features derived from a CNN architecture (pre-trained by means of colossal datasets), embed sufficient discriminatory information that could be tailored to our specific CVC-ColonDB dataset. The comparison with state of the art methods clearly confirmed the boost of performance brought by our method. For future work, we plan to deploy our method on other polyp datasets including ASU-Mayo clinic database, as well as other standard trained CNN architectures such as VGGNet.

5. REFERENCES

[1] F. Haggar and R. Boushey, “Colorectal cancer epidemiology: Incidence, mortality, survival, and risk factors,” *Clinics in Colon and Rectal Surgery*, vol. 22, no. 04, pp. 191–197, nov 2009.

[2] A. Castells, “Choosing the optimal method in programmatic colorectal cancer screening: current evidence and controversies,” *Therapeutic Advances in Gastroenterology*, vol. 8, no. 4, pp. 221–233, mar 2015.

[3] B. Taha, N. Werghi, and J. Dias, “Automatic polyp detection in endoscopy videos: A survey,” in *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, Feb 2017, pp. 233–240.

[4] G.D. Magoulas, V.P. Plagianakos, and M.N. Vrahatis, “Neural network-based colonoscopic diagnosis using on-line learning and differential evolution,” *Applied Soft Computing*, vol. 4, no. 4, pp. 369–379, sep 2004.

[5] D.E. Maroulis et al. “CoLD: a versatile detection system for colorectal lesions in endoscopy video-frames,” *Computer Methods and Programs in Biomedicine*, vol. 70, no. 2, pp. 151–166, feb 2003.

[6] Y. Wang et al. “Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1379–1389, July 2014.

[7] C. van Wijk et al. “Detection and segmentation of colonic polyps on implicit isosurfaces by second principal curvature flow,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 688–698, March 2010.

[8] A. El Khatib, N. Werghi, and H. Al-Ahmad, “Automatic polyp detection: A comparative study,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 2669–2672.

[9] A. E. Khatib, N. Werghi, and H. Al-Ahmad, “Enhancing automatic polyp detection accuracy using fusion techniques,” in *Int. Midwest Symposium on Circuits and Systems (MWSCAS)*, Oct 2016, pp. 1–4.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, may 2015.

[11] N. Tajbakhsh, S. R. Gurudu, and J. Liang, “Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks,” in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, April 2015, pp. 79–83.

[12] N. Tajbakhsh, J.Y. Shin, S.R. Gurudu, R.T. Hurst, C.B. Kendall, M.B. Gotway, and J. Liang, “Convolutional neural networks for medical image analysis: Full training or fine tuning?,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[13] J. Kovacevic P. Vandewalle and M. Vetterli, “Reproducible research in signal processing,” *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 37–47, v 2009.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1–9.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[16] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F.c.n. Pereira, C.j.c. Burges, L. Bottou, and K.q. Weinberger, Eds., pp. 1106–1114. 2012.

[17] J. Bernal, J. Snchez, and F. Vilario, “Towards automatic polyp detection with a polyp appearance model,” *Pattern Recognition*, vol. 45, no. 9, pp. 3166 – 3182, 2012.

[18] N. Tajbakhsh et al. *A Classification-Enhanced Vote Accumulation Scheme for Detecting Colonic Polyps*, pp. 53–62, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[19] N. Tajbakhsh, C. Chi, S. R. Gurudu, and J. Liang, “Automatic polyp detection from learned boundaries,” in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, April 2014, pp. 97–100.

[20] J. Bernal, J. Snchez, and F. Vilario, “Impact of image preprocessing methods on polyp localization in colonoscopy frames,” in *Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2013, pp. 7350–7354.

[21] S. H. Bae and K. J. Yoon, “Polyp detection via imbalanced learning and discriminative feature learning,” *IEEE Transactions on Medical Imaging*, vol. 34, no. 11, pp. 2379–2393, Nov 2015.

[22] N. Tajbakhsh and S. R. Gurudu and J. Liang, “Automated polyp detection in colonoscopy videos using shape and context information,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 2, pp. 630–644, Feb 2016.

Fast polyps tracking with color image regions alignment

Bilal TAHA [‡,⋆]***, Christophe DOIGNON*** [‡]***, Naoufel WERGHI*** [⋆] ***and Jorge DIAS*** [⋆]
‡ *ICube Laboratory - UMR 7357, University of Strasbourg, France.*
Contact: c.doignon@unistra.fr
⋆ *Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, UAE*

This paper addresses the visual tracking of polyps to finely capture the texture and shape characteristics along a video endoscopy. Manual inspection of long videos endoscopy suffers from an estimated 9-28% miss rate, hence the introduction of automatic polyp detection and visual tracking of such abnormal protrusions would open large prospects. The aim of this work is to develop an appropriate tool based on recent advances in deterministic visual tracking techniques, so as to assist the diagnostic of colorectal cancer. To this end, our method combines intensity and chromatic signals in the same framework - a novel similarity function which embeds multiple signals - so as to handle both the size, shape, color and illumination variabilities. Furthermore, special attention have been dedicated to two aspects in this work, 1) the delimitation of the region-of-interest when one has to deal with missing or irrelevant image data, and 2) the real-time issue for practical achievement.

1 Introduction

According to the World Health Organization, colorectal cancer (CRC) is the second most common cancer both for men and women in France, causing an estimated 11.4% of all cancer-related deaths every year for men, and 13,7% for women [2]. A crucial element in the prevention of CRC is the early detection and removal of abnormal protrusions, called polyps (see Fig. 1), in the colorectal tracts by means of regular endoscopy procedures. Endoscopies, however, involve manual inspection of long videos by a specialist, a tedious process that suffers from an estimated 9-28% miss rate [3].

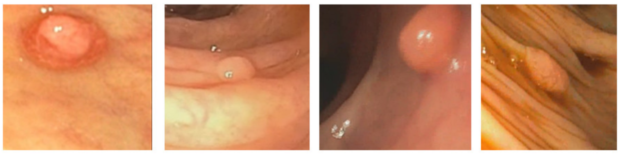


Figure 1: *Examples of polyps with different shapes, sizes and colors (ASU-Mayo Clinic database [1]).*

The introduction of automatic polyp detection algorithms, based on image analysis and machine learning techniques, can assist specialists in inspecting endoscopy videos, and thus can reduce the probability of missed polyps. Looking at the polyp samples in Fig. 1, we can see that polyps may appear with different shapes, sizes, and colors. The variability may be due to both intrinsic properties as well as external factors, such as lighting, viewpoint, etc. This variability makes necessary to target methods with a large scale-invariant degree, robust to changes in illumination, as well as general enough to capture global discriminating features common to diverse sets of polyps [1].

2 Contribution

Recently, an automatic polyps detection and classification algorithm has been proposed by the Khalifa University research center at Abu Dhabi, UAE [4]. It is based on deep learning approach (pre-trained convolutional neural network and SVM classifier) and it has been successfully validated with several clinical databases. To go on further, detection and tracking methods should be sufficiently efficient (fast and robust) such that the latter can be performed in real-time or at least within an interactive time frame, a practical trait that has not been given much consideration in the literature. Hence, starting a collaborative research project with the ICube laboratory of Strasbourg University and CNRS, the expected main objective of this work is to finely capture both the texture and shape characteristics of the selected image regions and to track them across the video frames so as to assist the diagnostic of colorectal cancer.
Most of visual tracking techniques involve features-based, simple geometrical-based, or grid-based reference region as input data to characterize the region-of-interest (or ROI for short). Once polyps detection is done, firstly one can look for closed contours to ac-

curately circumscribe the segmented regions, and to equally characterize the geometry of the borders of each ROI, providing a measurement of the evolving shape. Secondly, the color pixel distribution within the ROI (normalized luminance intensity and color differences) is encoded in a two-dimensional signal (per pixel), and it is used herein to proceed the temporal coherency of the tracking between successive frames, by means of an image region alignment, whereas it is usually performed with the luminance signal only. In this paper, we present a visual tracking method involving the minimization of a similarity function based on both the intensity and the chromatic signals inside the ROI, in order to perform the expected alignment.

3 Methodology

3.1 Background and related work

An efficient tracking should take care of temporal continuities by relying on inter-frame dependencies [5] and must allow as well to focus the analysis at regions which are more likely to contain a close appearance, the so-called region-of-interest (ROI) $\mathcal{R}$, subjected to local deformations and displacements, all gathered within an appearance motion model and a vector p of parameters. One of the most well-known region-based tracking methods is the Lucas-Kanade approach (LK) [6] which is an image alignment carried out with the minimization of errors at pixel level $\mathbf{x}$ between the current image I warped back, $I(W(\mathbf{x}; p + \Delta p))$ and the reference ROI (or template) $T(\mathbf{x})$. The warp $W(\mathbf{x}; p)$ takes the pixel $\mathbf{x}$ in the coordinate frame of the template T and maps it to the sub-pixel location $W(\mathbf{x}; p)$ in the coordinate frame of the image I. Nowadays many variants of the LK algorithm are available and provide significant improvements wrt to the original method, reducing the computational cost with some pre-computations steps [7], increasing the robustness wrt to illumination changes [8], extending the inter-frames linear motion model with parameter set p to non-linear models [9], or considering other similarity functions, other than the SSD (Sum of Squared Differences) like the Mutual Information (MI) [10] or the Sum of Conditional Variance (SCV) [8]. As this dense data-based technique leads to deterministic computer algorithm (as opposed to feature-based technique with outliers rejection, predictive filtering and matching process between subsets of features in adjacent frames [11]), it could be unpractical indeed for real-time issue regarding the high resolution of most visual sensors (Full HD and UHD) used by endoscopic color cameras in the operating room. It is therefore of prime importance to accurately delimitate the ROI with the detected polyp image, to estimate the incremental apparent motion (including deformations) Δp whatever are the illumination changes, sizes and shape variations.

3.2 Details of the method

To tackle above problems, we have combined 1) the inverse compositional alignment technique with $I(W(\mathbf{x}; p))$ and $T(W(\mathbf{x}; \Delta p))$ in a 2) SSD criterion with preconditioning of 3) normalized color images using 4) the Ohta representation of pixel colors inside the region-of-interest $\mathcal{R}$. The first three ingredients have been already applied to tracking purposes; The inverse compositional alignment is one of the least computationally demanding iterative alignment algorithm, and it has been recently be reformulated by Lui *et al.* [12] to support a preconditioning step for dealing with missing data in the ROI (specular effects, shadows, self-occlusions,...).
According to Ohta et al. [13], the three color features $I_1 = (R + G + B)/3$ (intensity), and $I_2 = R - B$ and $I_3 = (2G - R - B)/2$ (the two chromatic signals) are simple linear combinations of the (R, G, B) and are orthogonal each other. These two properties are fully exploited in our method to parallelize the warping process (with the same motion model p for computing the warping transformation W, the same bilinear interpolation function, but not the same data) with the following similarity function of that Inverse Color Compositional (ICC) alignment:

$$\mathcal{C}_{ICC} = \sum_{\mathbf{x}\in\mathcal{R}} \sum_{i=1}^{3} \{T_i(W(\mathbf{x}; \Delta p)) - I_i(W(\mathbf{x}; p))\}^2 \qquad (1)$$

and to easily compute the pixel derivatives in the minimization process.
To achieve fast computations of the successive alignments, one has to bring several efforts for the implementation issue; To that purpose, we have chosen to fully parallelize the program execution by means of the SIMD (Single Instruction Multiple Data) method for multithreading programming with OpenMP [14].

4 Perspectives

We are being developping the polyps tracking with Matlab software and video databases. This software platform is currently used to analyze the bottleneck instructions inside the set of online steps, especially in the iterative optimization of the alignment process, and it is worth noting that the preconditoning stage proposed by [12] is helpful to speed-up the convergence of the algorithm. Moreover, the Ohta's color representation proves to be a good choice when one has to compute the color errors and derivatives in a linear way. In the near future, we plan to efficiently implement the algorithm onto a dedicated multicore workstation by means of SIMD programming.

References

[1] N. Tajbakhsh, S. Gurudu, and J. Liang, “Automatic polyp detection using global geometric con-

straints and local intensity variation patterns," in *Medical Image Computing and Computer-Assisted Intervention MICCAI*, vol. 8674, Lecture Notes in Computer Science, Springer, 2014, pp. 179–187.

[2] "Cancer country profiles," *World Health Organization*, 2014. [Online]. Available: http://www.who.int/cancer/country-profiles/en/.

[3] D. Heresbach *et al.*, "Miss rate for colorectal neoplastic polyps: A prospective multicenter study of back-to-back video colonoscopies," *Endoscopy*, vol. 40, no. 4, pp. 284–290, April 2008.

[4] B. Taha, J. Dias, and N. Werghi, "convolutional neural network as a feature extractor for automatic polyp detection," in *IEEE Int'l Conf. on Image Processing*, (accepted), 2017.

[5] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, 2006.

[6] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the Int'l Joint Conference on Artificial Intelligence*, 1981.

[7] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int'l Journal of Computer Vision*, vol. 53, no. 3, pp. 221–255, 2004.

[8] B. Delabarre and E. Marchand, "Dense non-rigid visual tracking with a robust similarity function," in *IEEE Int.'l Conf. on Image Processing*, Paris, France, 2014.

[9] V. Gay-Bellile, A. Bartoli, and P. Sayd, "Direct estimation of nonrigid registrations with image-based self-occlusion reasoning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, p. 87104, 2010.

[10] N. Dowson and R. Bowden, "Mutual information for lucas-Kanade tracking (milk): An inverse compositional formulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, January 2008.

[11] R. Szeliski, "Image alignement and stitching: a tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[12] V. Lui, D. Gamage, and T. Drummond, "Fast inverse compositional image alignment with missing data and re-weighting," in *British Machine Vision Conference*, 2015, pp. 54.1–54.12.

[13] Y. Ohta, T. Kanade, and T. Sokai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222–241, 1980.

[14] C. Doignon, "Multiprogrammation et programmation multitâches," in *Computer Engineering Lecture at Télécom Physique Strasbourg*, 2017.

Classification of Cervical-Cancer Using Pap-Smear Images: A Convolutional Neural Network Approach

Bilal Taha(✉), Jorge Dias, and Naoufel Werghi

Department of Electrical and Computer Engineering, Khalifa University,
Abu Dhabi, UAE
bilal.taha@kustar.ac.ae

Abstract. Cervical cancer is the second most common and the fifth deadliest cancer in women. In this paper, we propose a deep learning approach for detecting cervix cancer from pap-smear images. Rather than designing and training a convolutional neural network (CNN) from the scratch, we show that we can employ a pre-trained CNN architecture as a feature extractor and use the output features as input to train a Support Vector Machine Classifier. We demonstrate the efficacy of such a new employment on the Herlev public database for single cell pap-smear, whereby the experimental results show that our proposed system neatly outperforms other state of the art methods.

Keywords: Pap-smear classification · Deep learning · Convolutional neural network

1 Introduction

For many years, cancer has been one of the biggest threats to human life, and the number of new cases is expected to rise by about 70% over the next 2 decades [2]. Cervical cancer, in particular, is the second most common and the fifth deadliest cancer in women [24]. The low rate of cancer survival is due to the fact that the majority of cancer cases are detected at advanced stages. There is a consensus in the medical community on the vital need of early detection of cancer for effective treatment. Indeed, some studies reported that cervical cancer is the most preventable disease with the incidence rates getting reduced by 80% [12] through early detection, although this sharp increase in figures might have been influenced by lead time bias and over-diagnosis [11]. Accurate and early cancer detection is very important for timely diagnosis and effective treatment. In fact, the inability to detect cancer in its early stages may cause the treatment to be delayed to a more advanced stage with more severe implications for survival rates and resource utilization. On the other hand, false detection of cancer may lead to unnecessary invasive treatments that might be both physically and emotionally traumatic to the patient, in addition to being costly to the health care system

M. Valdés Hernández and V. González-Castro (Eds.): MIUA 2017, CCIS 723, pp. 261–272, 2017.
DOI: 10.1007/978-3-319-60964-5_23

in terms of human and logistic resources. A variety of diagnostic tools are used in screening depending on the type of cancer. These tools include chest x-ray, computerized tomography (CT) scan, bronchoscopy, positron emission tomography (PET) scans and microscopic images. This last modality utilized in the detection of cervical cancer has the advantage of little or no side effects, as the procedure employed for the acquisition of microscopic images is virtually non-invasive. Screening for cervical cancer uses microscopic images of sample of cells collected from the cervix area. The cells undergo Papanicolaou staining method [4] which aims to visualize cells and cell components under the microscope allowing to display the variations of cellular morphology, and to differentiate the main cells from the debris cells.

The Pap smear slides usually contain both of single cells and clusters of cells. Most of cells are found with high degree of overlapping. Similar to other cells in human body, a cervical cell consists of two main components. One is the nucleus located about the center of cell surrounded by the cytoplasm. Normally, nucleus shape is small and almost round. Its intensity is darker than cytoplasm. In dysplastic cells, or abnormal cells, the cell will not grow and divide as it should. This is referred as precancerous cell. A sample of normal and abnormal cells is shown in Fig. 1. The dysplastic cells are categorized into mild, moderate, and severe dysplastic. A high amount of the mild dysplastic cells will disappear without becoming malignant, whereas severe dysplastic cells are likely to turn into malignant cells. The squamous dysplastic cells generally have larger and darker nuclei and tend to cling together in clusters. In severe dysplastic cells, nuclei are large, with dark granules and usually deformed. In Pap Smear image analysis, the cervical cells are divided into 7 classes, categorized by cell appearance, especially related to the nucleus.

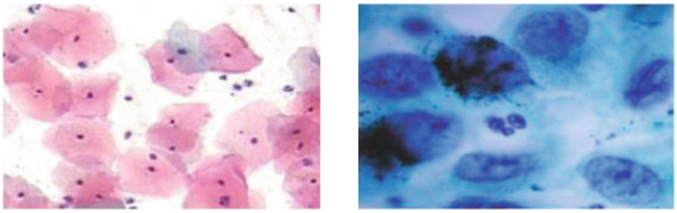


Fig. 1. Sample of cervical cell microscopic images. (a) Normal cells (b) Abnormal cells

Cervix cancer classification using pap-smear images adopts, basically, a three-stage paradigm, namely cell detection, cell segmentation, and cell classification. For cell detection, some methods, as explained in [28] employed contrast limited adaptive histogram equalization and global thresholding applied to the red, green and blue channels of the image. Automatic dell detection reached mature stage, and does not present any particular challenge nowadays.

For the cell segmentation a variety of methods have been proposed. They can be categorized into region-based methods and contour-based methods. In the former, cell pixels are separated into nucleus and cytoplasm based on regional image pixel similarity (or dissimilarity). Methods in this category fall into grey level histogram methods [20] and clustering methods using watersheds [13,19]. Threshold selection is a major issue with the gray-level histogram methods. Clustering methods, though they are threshold-free, result in an over-segmentation scenario as they fix the number of clusters to two (two clusters corresponding to the nucleus and the cytoplasm), whereas the actual number might be larger because of the color variability within the cell region. The watershed methods, though more robust, present an intrinsic limitation arising from the fact that it relies on the principle of touching regions exhibiting a narrow "neck" on the area of contract. Consequently, nucleus exhibiting thick or blurred boundaries with the cytoplasm cannot be detected reliably [26]. Contour-based methods detect the contour in the image by marking the boundary of the nuclei with respect to the cytoplasm. [27] used the concept of active contour, which is a kind of parameterized closed curve that iteratively deform until it fits the boundary edges. However this method requires manual initialization. Other resent work employed edge detector techniques [22] which work fine for clear single cell images, but their performance degrades considerably for cases of non-homogeneous cell regions and overlapping cells. And above all, these methods inherit the sensitivity to image noise and artifacts that characterize edge detection operators.

Once the cell is segmented, features are extracted to be used as input for cell classification. Jantzen et al. [14] proposed several important cell features that are used for Pap smear image analysis, derived from the nucleus and cytoplasm areas, and which include brightness, shorter diameter, longest diameter, elongation, roundness, perimeter, maxima, minima (the number of pixels with the maximum/minimum intensity value in a 3×3 neighborhood of the specific area). In terms of the features that are extracted from both the cytoplasm and the nucleus area, the nucleus position and the ratio $nucleus\ size)/cytoplasm\ size$ are calculated. Thus, a total of 20 features are considered important for the analysis of Pap smear images. These feature has been used later in [22]. Recently Boral et al. [5] Consolidated the set of features by novel color and texture features extracted using Ripplet Type I transform, Histogram first order statistics and Gray Level Co-occurrence Matrix Ripplet Type I transform, Histogram first order statistics and Gray Level Co-occurrence Matrix.

In the classification stage, most approaches proposed machine learning methods [5,9,21,22,28]. Marinakis and Dounias [21] employed two classifiers in their method namely 1-Nearest Neighbor and the Weighted (w) k-Nearest Neighbor classifier. The wk-Nearest-Neighbor is used to give different weighting for the features according to the distance to the test samples. To accommodate for the huge number of features extracted from the pap-smear images, Ant Colony Optimization method is utilized as a feature reduction mechanism, whereby Plissiti et al. [28] proposed to use an unsupervised learning technique for the classification of pap-smear images. They have focused on the nucleus features only and applied different feature reduction methods to select a subset from the features. The low dimensional features used in Spectral Clustering and fuzzy C-means classifiers for the decision making process. Chen in [9] proposed and integrated system providing tolls for: selecting the cell, automatically detect the nucleus and the cytoplasm regions, extracting 13 morphological and texture descriptors, and using a Support Vector Machine (SVM) for classification. The same classifier has been used in [22]. In the recent work [5], Bora et al. employed a majority-voting fusion method including a Multilayer Perceptron (MLP), Random Forest (RF) and Least Square Support Vector Machine (LSSVM).

In this work, we focus on the classification step where we propose a novel approach for cervix cancer detection. Rather than designing a convolutional neural network from the scratch, we employ a pre-trained CNN architecture coupled with a support vector machine at the back-end, saving thus time and resources. This employment of the pre-trained CNN architecture as a features extractor is deeply instigated with different experiments, and thus comparing the performance of the features across the different architecture layers outputs. The closest work to our approach is [6] whom used a CNN as a feature extractor. However, our method is distinguished by the following aspects:

- CNN was used to transfer different level of features from different layers contradicting to the work in [6] where they assume the best features are obtained from FC7 without any experimentation. Since there is a very few work implemented on the classification of cervix cancer from pap-smear images using deep learning, it turns out essential to perform more experimentation to obtain a deeper understanding for this method.
- Two testing sets was performed in this work by employing the Herlev public database where the other method emphasized more on their own generated database.
- SVM classifier was utilized for the decision making process because of its capability to handle high dimensional features where the other work used least-squares SVM (LSSVM) and softmax regression classifiers.
- Since we are using SVM which is capable to manage huge number of features, feature selection techniques was not empowered because the focus is on the usage of the CNN as a feature extractor and the transferred features from its different layers. However, the work in [6] have implemented Maximal Information Compression Index as a feature selection to reduce the number of features obtained from the CNN.

The evaluation of our paradigm illustrates the superiority of our method when compared with state of the art methods. The remainder of the paper will be organized as follows: Sect. 2 introduces the proposed approach and elaborates on its rational. Section 3 describes the different experiments and the related results. Section 4 concludes the paper.

2 The Proposed Method

In image analysis, designing the appropriate features for a given interpretation task has been a central problem in computer vision and in medical image analysis. Explicit feature design extraction in medical image analysis requires subject-matter expertise. In this process, the visual information on which the physician relies in his assessment is not necessarily reflected into a suitable computational representation. Moreover, the practical considerations in the extraction and the usages of these features make the reproducibility of these related methods often problematic [25]. To overcome these challenges we propose a deep learning approach, whereby Convolution Neural Network (CNN) is employed to replace a handcrafted and customized features, which would be strongly sensitive to multiple parameters.

It is known that through a hierarchical unsupervised or semi-supervised feature design, CNN's can produce effective representation of the visual data [18]. Basically, a CNN deep learning architecture is composed of a sequence of cascading layers performing basic operations such as convolution, subsampling, followed by another sequence of fully connected layers, which act similarly as a classic artificial neural network. These fully connected layers can be replaced by a support vector machine classifier. In another hand, training a CNN network from scratch requires a large data-set, which is a tedious process and often cannot be afforded in medical applications, including database of pap-smear classification. Also, in addition to requiring considerable computational resources, there is no systematic guidelines as for the optimal choice of the architecture in terms of depth (number of layers) and structure.

An economic alternative is to use pre-trained CNN architectures, that are proven to have good performance through training and validation over a huge database, and then tune, via training conducted on a specific application dataset, the pre-trained weights of the architecture. This procedure, known as fine-tuning, can be performed either across the whole CNN or at specific layers. In this paradigm, the lower layers are kept the same since they have learned generic features (e.g. edge, region) that are less dependent on the final application [18], whereas the top layers are removed and the linear classifier is trained to accommodate the new application-specific database.

In our approach, we advocate the hypothesis that a trained CNN architecture embeds sufficiently rich feature representations that can be utilized as input to train a standard classifier, such as the Support Vector Machine, relieving thus the system from laborious training from scratch or fine tuning. Therefore a pre-trained CNN is then deployed as a feature extractor for our specific image interpretation task of pap-smear detection, as depicted in the block diagram in Fig. 2. The database employed for training and testing include different patch sizees of the pap-smear. Therefore, the resolution for the patches less than the standard size 227 $times$227 adopted in AlexNet, was completed by empty regions with a white background.

There are several pretrained CNN architectures that can be investigated, such as GoogleNet [1] and VGGNet [29]. In our method, we explored AlexNet [16]. This CNN architecture was trained with 1.2 million images for 1000 different classes, thus the learned features are expected to span a large spectrum of visual information. The main layers of the AlexNet architecture is briefly described in Table 1.

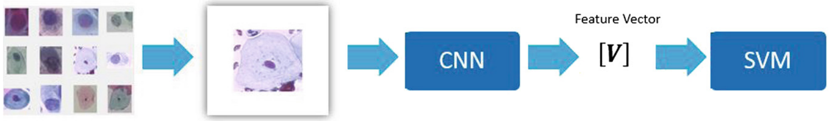


Fig. 2. Block diagram for the CNN as a feature extractor

Table 1. Summary of AlexNet architecture

Layer	Type	Input	Kernel	Stride	Pad	Output
Data	Input image	$227 \times 227 \times 3$	N/A	N/A	N/A	$227 \times 227 \times 3$
conv1	Conv	$227 \times 227 \times 3$	11×11	4	0	$96 \times 55 \times 55$
pool1	Max pooling	$55 \times 55 \times 96$	3×3	2	0	$96 \times 27 \times 27$
conv2	Conv	$27 \times 27 \times 96$	5×5	1	2	$256 \times 27 \times 27$
pool2	Max pooling	$27 \times 27 \times 256$	3×3	2	0	$256 \times 13 \times 13$
conv3	Conv	$13 \times 13 \times 256$	3×3	1	1	$384 \times 13 \times 13$
conv4	Conv	$13 \times 13 \times 384$	3×3	1	1	$384 \times 13 \times 13$
conv5	Conv	$13 \times 13 \times 384$	3×3	1	1	$256 \times 13 \times 13$
pool5	Max pooling	$13 \times 13 \times 256$	3×3	2	0	$256 \times 6 \times 6$
FC6	Fully connected	$6 \times 6 \times 256$	6×6	1	0	4096×1
FC7	Fully connected	1×4096	1×1	1	0	4096×1
FC8	Fully connected	1×4096	1×1	1	0	1000×1

Usually features from first layers are too generic to be employed as discriminative descriptors (see Fig. 3). As a result, we investigated features from the middle layers and onward, namely, Con4 till FC8. The output from one of these layers will be a sort of feature encoding of the pap smear images. These features will be then fed into the subsequent classifier block.

Here a fully connected neural network (ANN), a SoftMax classifier, or a SVM can be used. We choose the SVM for the following reasons: SVM and has the capacity to deal with a high-dimension input without compromising the computational complexity, and thus, contrary to ANN or SoftMax, can map the huge number of feature vectors across the different layers of the CNN. For instance, in [23], the SVM has been deployed across all six layers from layer 1 to layer 7 except 6 of the ConvNet model, whereas softmax has been used only at layer 5 and 7. Moreover, SVM classifier showed better overall discriminating power on that model where recently it has been observed that coupling SVM as a final layer improves the learning rate [17]. With regard to overfitting, the SVM is assumed to be less sensitive to overfitting, at least in principle, because of aforementioned feature dimensionality, in practice it provides mechanisms to control overfitting through the C parameter. Having said that. Generally, overfitting remains a general problem that practically has to be dealt with for any

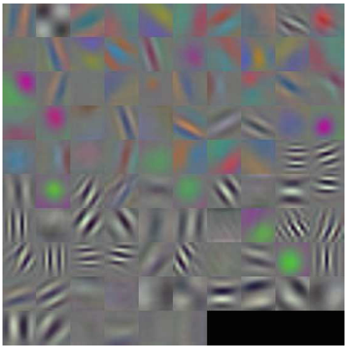


Fig. 3. Visualization for the first convolutional layer weights. Since we have 96 weights in the first layer of the AlexNet, the last 4 weights appeared black.

classifier [8]. In CNN specifically, the problem is more crucial because of the high dimensionality that characterize their architecture. Several mechanism have been proposed to address this issue, which include, dropout [15], data augmentation [15], regularization [3] and stochastic pooling [30].

When training the SVM, each feature vector is given a label either -1 or 1 (normal, abnormal) to create the feature-class pair $\{x, y\}$. Therefore, given L features $\{x_i, y_i\}$ such that $i = 1...L$, and $y_i \in \{1, -1\}$, $x \in \Re^D$, where D is the vector size. A hyper-plane separating the two classes could be written as

$$w^T x + b = 0 \tag{1}$$

the w is known as the weight vector which is normal to the separation hyper-plane, and b is known as the bias. In order to separate the two classes with the hyper-plane the following equation should be optimized

$$min(\frac{1}{2} w^T w + C \sum_{i=1}^{L} \xi_i) \tag{2}$$

subject to the constrain $y_i(w^T x_i + b) >= 1 - \xi_i$, where $\xi_i >= 0$ for $i = 1, ..., L$, and C is known as the penalty parameter. This will lead to the optimal hyper-plane that minimizes the distance between itself and all the training examples. The optimal hyper-plane, allows a classification to be done according to a decision function such as:

$$f(x) = \text{sgn}(w^T x + b) \tag{3}$$

3 Experimentation

To evaluate the performance of our method, Herlev pap smear database was used for training and testing. This database is publicly available and consists of 917 single cell images divided into 7 classes. Four categories are considered as abnormal images with different severity namely light dysplastic, moderate dysplastic,

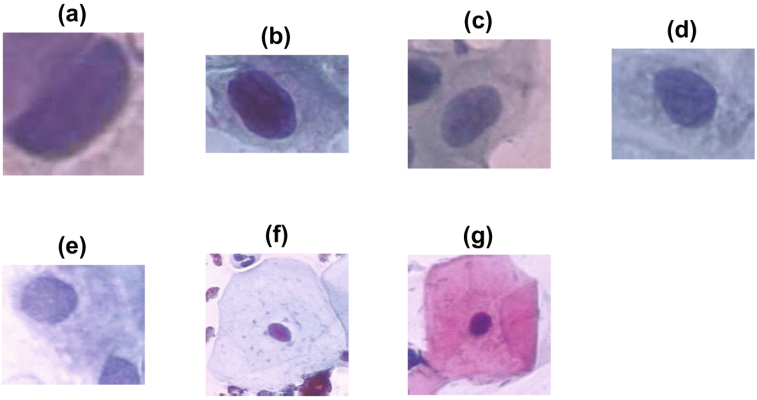


Fig. 4. Sample images from the 7 classes of the Herlev pap smear database (a) carcinoma in situ, (b) severe dysplastic, (c) moderate dysplastic, (d) light dysplastic, (e) normal columnar, (f) normal intermediate, (g) normal superficial.

severe dysplastic, and carcinoma in situ. The other three categories considered as normal are normal columnar, normal intermediate, normal superficial (Fig. 4). However, according to [10] when considering a two class classification problem, the columnar class is not considered as a normal nor an abnormal. As a result, the total number of normal class images is 144 while the total number of abnormal cell images is 675.

We selected and transferred features from different layers of the CNN, from shallow to deep layers, where the aim was to assess the performance of the CNN as a feature extractor and its effectiveness in the detection scenarios on the specified database. The main focus in the experiment is the quality, in terms of classification power, of the extracted features from the CNN. The features were deployed using different layers from the pre-trained CNN. AlexNet was trained using the ImageNet database which consists of non-medical images, therefore there is a need to know the best layer that will provide the best features discriminating normal pap-smear cell images from the abnormal pap-smear ones. As we mentioned earlier, we considered only deep layers, starting from Conv4. In this transfer learning scheme, the layers up to the output features layers are frozen and the output features are used to train the SVM classifier. For example, considering conv5, as the feature output layer, we keep the weights across the layers conv1 to conv5 at their pre-trained values, while training the SVM classifier. While this scheme reduces the number trained entities, the number of features remain large, as an example, the dimension of the obtained feature vector from the fourth convolution layer (C4) is $13 \times 13 \times 384$ which is equal to 64896 features. The two-class classification problem was implemented where the number of normal pap-smear cell images is 144, while the number of the abnormal images is 675. For training protocol we adopted the 70%, 30% for training and testing, respectively. Table 2 reports the best recall and precision performance from each layer. For instance, C5 in Table 2 refers to the features coming out from the layer Conv5. It is interesting to notice that the top performance is obtained with features coming out from a deep layer (fc7), which are more descriptive than their shallow layers counterparts (e.g. Conv4 - see Table 1).

Table 2. The recall and precision values for 2-class classification of pap-smear without the columnar class.

Experiment 1				
		Recall	Precision	Accuracy
CNN feature	Conv4	99.01%	99.02%	98.37%
	Conv5	100%	98.54%	98.78%
	FC6	99.5%	99.01%	98.6%
	FC7	**99.51%**	**99.5%**	**99.19%**
	FC8	98.6%	97.8%	97.9%

In another experiment, we tested our method on the data-set without removing the columnar class which results in having a number of 242 normal cell images and 675 abnormal cell images. The recall, precision, and accuracy are reported in Table 3.

Table 3. The recall and precision values for 2-class classification of pap-smear including the columnar class.

Experiment 2			
		Recall	Precision
CNN feature	Conv4	94.1%	87.6%
	Conv5	97.04%	89.14%
	FC6	99.01%	85.2%
	FC7	64.04%	96.3%
	FC8	59.11%	95.24%

We compared our method with two state of the art methods that used the same database and ignore the columnar class [7,14]. Table 4 reports the performances of the two methods together with the results achieved by our method. We found that our approach outperforms all the existing paradigms in terms of recall, precision, specificity and accuracy with scores of 99.51%, 99.5%, 97.67% 99.19% respectively.

Table 4. Recall, precision, specificity, and accuracy scores in percent by setting the parameters according to the best results in each experiment.

Method	[7]	[14]	Our method
Recall	95.11	-	99.51
Precision	-	-	99.5
Specificity	96.53	-	97.67
Accuracy	95.36	93.75	99.19

4 Conclusion

In this work we presented a deep learning solution for cervix cancer screening using pap-smear images. In this application, we proposed a novel employment whereby a pre-trained architecture, the AlexNet, is used as feature extractor, then coupled with a classic SVM classifier. This approach relives the system form the high computational and resource demanding training from the scratch or even fine-tuning. The evaluation and testing conducted with the Herlev database, confirmed the rational of our hypothesis that the features derived from a CNN architecture (pre-trained by means of colossal datasets), embeds sufficient discriminatory information to be tailored to our specific Herlev pap-smear dataset. In the future work we plan to investigate further our approach on other pap-smear datasets and other standard trained CNN architecture such as VGGNet.

References

1. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
2. Ferlay, J., et al.: Cancer incidence and mortality worldwide. In: GLOBOCAN 2012, vol. v1.0 (2010)
3. Srivastava, N., et al.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
4. Arbyn, M., Anttila, A., Jordan, J., Ronco, G., Schenck, U., Segnan, N., Wiener, H., Herbert, A., von Karsa, L.: European guidelines for quality assurance in cervical cancer screening. Ann. Oncol. **21**(3), 448 (2010). Second edition summary document
5. Bora, K., Chowdhury, M., Mahanta, L.B., Kundu, M.K., Das, A.K.: Automated classification of Pap smear images to detect cervical dysplasia. Comput. Methods Programs Biomed. **138**, 31–47 (2017)
6. Bora, K., Chowdhury, M., Mahanta, L.B., Kundu, M.K., Das, A.K.: Pap smear image classification using convolutional neural network. In: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2016, NY, USA, pp. 55:1–55:8. ACM, New York (2016)
7. Cawley, G.C., Talbot, N.L.C.: On over-fitting in model selection and subsequent selection bias in performance evaluation. J. Mach. Learn. Res. **113**(2), 2079–2107 (2014)

8. Chankong, T., Theera-Umpon, N., Auephanwiriyakul, S.: Automatic cervical cell segmentation and classification in Pap smears. Comput. Methods Programs Biomed. **113**(2), 539–556 (2010)
9. Chen, Y.F., Huang, P.C., Lin, K.C., Lin, H.H., Wang, L.E., Cheng, C.C., Chen, T.P., Chan, Y.K., Chiang, J.Y.: Semi-automatic segmentation and classification of Pap smear cells. IEEE J. Biomed. Health Inform. **18**(1), 94–108 (2014)
10. Gençtav, A., Aksoy, S., Önder, S.: Unsupervised segmentation and classification of cervical cell images. Pattern Recogn. **45**(12), 4151–4168 (2012)
11. Gigerenzer, G., Wegwarth, O.: Five year survival rates can mislead. BMJ **346**, f548 (2013)
12. Henschke, C.L., et al.: International early lung cancer action program investigators: survival of patients with stage 1 lung cancer detected on CT screening. N. Engl. J. Med. **335**, 1763–1771 (2006)
13. Costa, J.A.F., Mascarenhas, N.D., de Andrade Netto, M.L.: Cell nuclei segmentation in noisy images using morphological watersheds. In: International Society for Optical Engineering, vol. 3164, pp. 314–324 (1997)
14. Jantzen, J., Norup, J., Dounias, G., Bjerregaard, B.: Pap-smear benchmark data for pattern classification. In: Proceedings of NiSIS 2005: Nature Inspired Smart Information Systems, EU Co-ordination, pp. 1–9 (2005)
15. Krizhevsky, A., Sutskever, I., Hinton, E.: Imagenet classification with deep convolutional neural networks (2012)
16. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1106–1114 (2012)
17. Berrada, L., Zisserman, A., Kumar, M.P.: Trusting SVM for piecewise linear CNNs. In: Proceedings of International Conference on Learning Representations (2017, to appear)
18. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
19. Lezoray, O., Cardot, H.: Cooperation of color pixel classification schemes and color watershed: a study for microscopic images. IEEE Trans. Image Process. **11**(7), 783–789 (2002)
20. Mahanta, L.B., Nath, D.C., Nath, C.K.: Cervix cancer diagnosis from Pap smear images using structure based segmentation and shape analysis. J. Emerg. Trends Comput. Inf. Serv. **3**(2), 245–249 (2012)
21. Marinakis, Y., Dounias, G.: Nature-inspired intelligent techniques for Pap smear diagnosis: ant colony optimization for cell classification (2006)
22. Mbaga, A., ZhiJun, P.: Pap smear images classification for early detection of cervicel cancer. Int. J. Comput. Appl. **118**(7), 10–16 (2016)
23. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings of European Computer Vision Conference, pp. 818–833 (2014)
24. World Health Organization: Fact Sheet No. 297: Cancer, February 2006
25. Vandewalle, P., Kovacevic, J., Vetterli, M.: Reproducible research in signal processing. IEEE Sig. Process. Mag. **26**(3), 37–47 (2009)
26. Pawley, J.B.: Handbook of Biological Confocal Microscopy. Springer, Heidelberg (2006)
27. Plissiti, M.E., Charchanti, A., Krikoni, O., Fotiadis, D.I.: Automated segmentation of cell nuclei in PAP smear images, October 2006
28. Plissiti, M.E., Nikou, C., Charchanti, A.: Automated detection of cell nuclei in Pap smear images using morphological reconstruction and clustering. IEEE Trans. Inf. Technol. Biomed. **15**(2), 233–241 (2011)

29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
30. Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. CoRR abs/1301.3557 (2013)