

THÈSE

présentée à

L'Université Louis Pasteur, Strasbourg I — École doctorale MSII
Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection
UMR 7005 CNRS/ULP

par

M. Frédéric LARUE

pour l'obtention du grade de

Docteur de l'Université Louis Pasteur
Discipline: Informatique

NUMÉRISATION DE PIÈCES D'ART
EN TERMES DE FORME ET D'APPARENCE
POUR LA VISUALISATION RÉALISTE
EN SYNTHÈSE D'IMAGES

Soutenue publiquement le 26 novembre 2008 devant le jury constitué de:

M. Jean-Michel DISCHLER, *Directeur*,
Professeur à l'Université Louis Pasteur de Strasbourg.

M. Christophe DOIGNON, *Rapporteur interne*,
Maître de Conférences à l'Université Louis Pasteur de Strasbourg.

M. Bernard PÉROCHE, *Rapporteur*,
Professeur à l'Université Claude Bernard de Lyon.

M. Daniel MENEVEAUX, *Rapporteur*,
Maître de Conférences à l'Université de Poitiers.

M. Roberto SCOPIGNO, *Examineur*,
Professeur à l'Université de Pise, Italie.

Remerciements

Je tiens, avant toute autre chose, à adresser mes remerciements à un certain nombre de personnes qui, de part leurs soutiens ou leurs discussions, ont contribué à l'aboutissement de ces travaux.

Je remercie d'abord M. Jean-Michel DISCHLER, mon directeur, pour avoir fait des pieds et des mains il y a quatre ans pour m'obtenir cette thèse.

Merci à M. Christophe DOIGNON, qui m'a été d'un grand secours en début de thèse concernant les aspects touchant à la vision.

Merci à M. Jean-Pierre CHAMBARD, directeur de la société HOLO3, pour avoir mis à notre disposition une partie du matériel nécessaire à la réalisation de ces travaux, ainsi qu'à MM. Vincent CHALVIDAN et Mohammed TAZEROUALTI pour leurs collaborations.

Merci aux membres de mon jury, et notamment aux rapporteurs, dont les remarques judicieuses m'ont permis d'améliorer la qualité de ce mémoire.

Merci à tous mes collègues de l'équipe IGG pour leur convivialité qui a aidé à animer un peu ces tristes locaux du LSIIT, mais également pour les discussions prolifiques, bien qu'informelles, que leur compagnie a souvent suscitées. Un merci tout particulier à MM. Lucas AMMANN et Olivier GÉNÉVAUX, avec qui j'ai eu grand plaisir à collaborer.

Merci à mes amis et à ma famille qui ont subi cette thèse bien malgré eux, mais toujours avec le sourire et sans jamais broncher. Un remerciement très spécial et très ému à mon amie Émilie pour m'avoir soutenu du début à la fin, dans les meilleurs comme dans les pires moments.

Enfin, merci à tous ceux qui, autres que ma mère et mon jury, liront ce manuscrit.

- Table des matières -

1	Introduction	11
1	Généralités	11
2	Le projet AMI-3D	13
2.1	Présentation	14
2.2	Objectifs	14
3	Problématiques	15
3.1	Les contraintes de la mesure de formes	15
3.2	Les difficultés liées à la mesure de l'apparence	18
4	Contributions	20
I	État de l'art	23
2	Mesure de formes	25
1	Processus de formation d'images	26
1.1	Le modèle sténopé	26
1.2	Les distorsions optiques	29
1.3	Calibrage	31
2	Lumière structurée par décalage de phase	31
2.1	Triangulation optique et stéréovision active	31
2.2	Acquisition de franges et paramétrisation par phase	33
2.3	Calcul de profondeur par décalage de phase	33
2.4	Démodulation	34
3	Représentation des données acquises	35
3	Recalage géométrique	37
1	Calcul de transformations rigides entre deux surfaces	38
1.1	Énoncé du problème	38
1.2	Solution optimale au sens des moindres carrés	40
2	Recherche automatique de correspondances	41
2.1	Extraction de caractéristiques géométriques	41
2.2	Définition d'invariants par transformations rigides	43
2.3	Caractéristiques et invariants multi-échelle	45

2.4	Détermination de la transformation de recalage	45
2.5	Caractérisation globale	46
3	Résolution par minimisation itérative	47
3.1	Description de l'algorithme	47
3.2	Amélioration de la convergence et de la robustesse	49
3.3	Problème du plus proche voisin	50
3.4	Utilisation d'attributs additionnels	52
4	Recalage global	54
4.1	Arbre de recouvrement minimal	55
4.2	ICP global	55
5	Algorithmes génétiques de recalage	56
6	Bilan	57
4	Intégration et filtrage	59
1	Intégration	60
1.1	Intégration basée maillages	61
1.2	Intégration volumique	64
2	Filtrage	66
2.1	Premier filtre simple : le filtre Laplacien	68
2.2	Autres filtres simples	69
2.3	Filtres adaptatifs à puissance variable	70
2.4	Filtres adaptatifs à masque variable	72
2.5	Travaux connexes	73
3	Bilan	73
5	Capture et restitution de l'apparence	75
1	Définition de l'apparence	76
1.1	La fonction de réflectance	76
1.2	Mesure de réflectances	78
1.3	Modèles de réflectances en synthèse d'images	78
2	Acquisition de l'apparence à partir d'objets réels	81
2.1	Recalage de points de vue	82
2.2	Localisation de sources lumineuses	83
2.3	Ajustement de textures	84
3	Reconstruction de textures bidirectionnelles	85
4	Reconstruction de textures monodirectionnelles	87
4.1	Point de vue fixe	87
4.2	Éclairage fixe — approches purement photométriques	88
4.3	Éclairage fixe — approches surfaciques	90
5	Reconstruction de textures diffuses	92
5.1	Combinaison d'images par moyennage	92
5.2	Extraction de la composante diffuse par illumination inverse	94
6	Bilan	95

II	Travaux réalisés	97
6	Capture de champs lumineux	99
1	Principe	101
2	Détermination des correspondances	102
2.1	Paramétrisation de surfaces réelles par la phase	102
2.2	Sélection des correspondances	104
2.3	Classification des correspondances	106
3	Description du protocole d'acquisition	106
3.1	Échantillonnage local de l'apparence	107
3.2	Recalage géométrique chaîné par bloc	109
3.3	Amélioration de la précision	110
4	Post-traitements	113
4.1	Raffinement à la volée du recalage chaîné	114
4.2	Recalage global	116
5	Résultats expérimentaux	117
5.1	Étude du recalage géométrique par phase	118
5.2	Comparaison des différentes configurations matérielles	121
5.3	Étude du raffinement à la volée	122
5.4	Étude de l'accumulation de l'erreur et du recalage global	124
5.5	Étude du recalage local de points de vue	126
6	Bilan	130
7	Traitements géométriques	133
1	Correction non rigide des acquisitions géométriques	135
1.1	Principe de l'algorithme	135
1.2	Détails d'implantation	136
2	Automatisation du processus de filtrage	137
2.1	Rappel sur le filtre bilatéral	138
2.2	Filtre bilatéral non itératif à un seul paramètre	139
2.3	Estimation automatique de l'unique paramètre	142
3	Intégration	144
4	Résultats préliminaires	144
4.1	Correction non rigide	144
4.2	Étude du filtre proposé et de son estimation automatique	146
4.3	Résultats obtenus après intégration	149
5	Bilan	150
8	Visualisation de champs surfaciques	153
1	Application 1 : visualisation des œuvres numérisées	154
1.1	Modèle de champs lumineux surfaciques	155
1.2	Reconstruction à partir de données réelles	155
1.3	Visualisation temps réel	158
1.4	Résultats expérimentaux	160
2	Application 2 : simulation de la réfringence	164
2.1	Rappels d'optique et travaux antérieurs	165

2.2	Principe de la méthode	167
2.3	Prétraitements	168
2.4	Algorithme de base pour le rendu temps réel	172
2.5	Améliorations	175
2.6	Gestion de l'opacité	177
2.7	Résultats	178
3	Bilan	184
9	Cas d'étude — Numérisation de tableaux	189
1	Travaux antérieurs	191
2	Principe	193
3	Acquisition et prétraitements	194
3.1	Configuration matérielle pour l'acquisition	194
3.2	Reconstruction de la texture bidirectionnelle	196
3.3	Extraction automatique de la toile	197
3.4	Classification du relief	199
4	Rendu	201
4.1	Rendu du relief négligeable	201
4.2	Rendu du relief significatif	202
4.3	Calcul d'illumination	203
4.4	Mécanisme adaptatif dépendant du point de vue	204
5	Résultats expérimentaux	206
5.1	Étude des performances de l'algorithme de rendu	206
5.2	Étude de l'influence de la taille des boîtes	209
5.3	Discussion sur l'utilisation de champs de hauteurs	210
5.4	Discussion sur l'acquisition de la texture bidirectionnelle	211
5.5	Discussion sur la compacité des données	212
6	Bilan	212
III	Conclusion	215
10	Conclusion générale	217
1	Rappel des contributions	217
2	Limitations et perspectives à courts termes	219
2.1	Acquisition et visualisation de champs lumineux surfaciques	220
2.2	Visualisation temps réel d'objets réfringents	221
2.3	Numérisation et visualisation de tableaux	222
3	Perspectives à longs termes	223
IV	Annexes	225
A	Glossaire photométrique	227
1	Définitions radiométriques	227
2	Correspondance radiométrie / photométrie	229
3	Acquisition de quantités photométriques	229

B Réflexion et réfraction	231
C Compression en harmoniques sphériques accélérée	233
D Liste de publications	237
Bibliographie	239

- Introduction -

De la numérisation et de son contexte

1 Généralités

«**numériser** : convertir une information (texte, son, image) sous une forme chiffrée binaire directement utilisable par un système informatique»

Sous cette définition généraliste et quelque peu sibylline se cache une grande diversité de technologies, de méthodologies et de domaines d'application. Aujourd'hui, certaines de ces technologies se sont même popularisées au point de s'intégrer de plain-pied dans notre quotidien. Citons, à titre d'exemple, les appareils photo numériques, proposés par la plupart des marchés de grande distribution, qui se répandent chez le commun des mortels à une vitesse vertigineuse. Et si cet outil est aujourd'hui le plus connu du grand public, il ne s'agit pourtant là que d'une seule application parmi un large éventail de possibilités.

Avec l'évolution des besoins et des technologies, la numérisation *stricto sensu* ne se définit d'ailleurs plus par sa seule dénomination. Ce sont les champs d'application eux-mêmes qui apportent aujourd'hui une définition exacte, mais nécessairement contextuelle, à ce que l'on entend par ce terme. À l'ère de l'informatique, l'acte de numérisation peut en effet être effectué de bien des manières selon la nature de l'information à acquérir : enregistrement sonore, photographie, vidéo, écho radar, etc. Les technologies impliquées sont à chaque fois différentes, et les formats de stockage de données spécifiques à chaque type d'information. Dans la pratique, le choix d'un domaine d'application détermine un certain nombre de besoins spécifiques. Ce sont ces besoins qui définissent quelles sont les quantités mesurables utiles à la bonne représentation de l'information à modéliser et, par là même, les choix et les moyens techniques à mettre en œuvre pour l'acte de prise de données, c'est-à-dire pour la numérisation elle-même.

Dans le secteur automobile, par exemple, la numérisation a quelques applications concrètes. Mais selon que l'information à acquérir est destinée à la simulation de crash-tests ou bien

à la représentation réaliste d'intérieurs de voiture, les moyens utilisés pour l'acquisition et le traitement des données peuvent être radicalement différents, sachant que la première application requerra une connaissance physique sur la déformation des matériaux alors que la seconde se préoccupera plus de l'aspect visuel de ces derniers. En résumé, le terme *numérisation* en tant que tel, sans la moindre motivation sous jacente, ne signifie rien.

L'exemple précédent illustre l'intérêt que peut susciter dans l'industrie les possibilités offertes par la numérisation. Cet intérêt va d'ailleurs grandissant, et s'exprime au travers de multiples applications, dans des domaines souvent très variés. Une forte demande émerge notamment des milieux où la visualisation est l'un des objectifs premiers. L'enrichissement de scènes virtuelles à l'aide d'éléments réels numérisés peut en effet permettre de favoriser l'immersion dans des milieux interactifs, comme par exemple les simulateurs de vol en aéronautique, pour lesquels le réalisme est franchement de mise. Plus prosaïquement, les systèmes de détection couramment utilisés par l'armée, tels que les radars ou les sonars, sont assimilables à des numériseurs. Mais ce qui génère encore une forte demande de la part de la clientèle industrielle est incontestablement la possibilité d'automatiser les tâches de contrôle de qualité à l'aide de l'outil informatique. On parle plutôt dans ce cas de métrologie, puisqu'il s'agit alors de capturer une information précise dont on pourra extraire des quantités mesurables, et donc comparables.

Outre ces considérations productivistes, l'autre grand domaine qui s'intéresse de près aux technologies de numérisation, et à juste titre, est celui de l'art et de la culture. Les apports de l'archivage numérique dans le cadre du patrimoine culturel sont en effet nombreux. Au premier plan des priorités, bien sûr, vient la préservation du patrimoine. Actuellement, des collections entières restent dans l'ombre, cachées du grand public, du fait de leur extrême fragilité. Les ouvrages anciens, dont le papier est si mince et les reliures si fragiles, sont un exemple particulièrement flagrant de ce carcan qui entrave la libre diffusion, et donc l'accès à la culture. Et si c'est avant tout dans un souci de conservation, c'est cependant à notre détriment qu'il s'impose. La copie numérique a ici un rôle important à jouer, offrant une alternative à la libre étude de telles pièces, à leur accessibilité au plus grand nombre, sans risque de les voir se détériorer rapidement, en proie à de trop fréquentes manipulations. La grotte de Lascaux, dont une copie, bien que partielle, fut reconstituée à l'identique sur la base de sa complète numérisation, témoigne de l'intérêt de ces nouvelles technologies sans lesquelles les peintures rupestres qui ornent ses murs ne seraient plus, aujourd'hui, offertes à nos yeux ébahis.

S'inscrivant plus ou moins dans la même veine, vient ensuite l'authentification des œuvres. La capacité à distinguer une copie d'un original est incontestablement l'un des enjeux majeurs dans la lutte contre le marché noir de l'art. L'utilisation d'une empreinte numérique fiable, de nature métrologique, peut, une fois de plus, permettre d'automatiser et de rendre plus robuste la détection d'objets contrefaits. Établie au préalable à partir d'une pièce dont la provenance est indiscutable, elle servira ensuite de signature pour l'identification en cas de perte ou de vol. Encore peu explorée jusqu'alors, cette problématique ouvre pourtant la voie vers un marché certain, tant les implications sont importantes pour la protection et la sécurité du patrimoine, et s'imposera certainement dans les prochaines décennies comme l'impératif de premier ordre.

L'apport du numérique dans le domaine culturel ouvre également un troisième grand volet : celui de la restauration. Là où une empreinte peut servir de substitut à l'étude ou de carte d'identité pour une œuvre d'art, elle peut également faire office de référence à un instant donné sur la composition des pigments ou des matériaux à partir desquels une pièce a été façonnée. Cette mesure inaltérable, utilisée par la suite comme étalonnage lors des futurs travaux de nettoyage ou de restauration, pourra garantir, de manière systématique, l'utilisation de pigments ou de matériaux de compositions identiques, ou dont les propriétés recherchées sont tout du moins similaires. Mais nous pouvons regarder encore plus loin : effectuée sur une période de temps donnée et à intervalles réguliers, cette mesure pourrait également permettre d'étudier le processus de vieillissement d'une œuvre dans son environnement d'exposition afin, d'une part, de vérifier que cet environnement soit bien adapté à sa conservation et, d'autre part, de planifier avec plus de rigueur et de régularité son calendrier de restauration.

Pour enfin clore cette discussion, parlons de l'aspect qui nous touche de plus près, du fait qu'il nous concerne directement : celui de la communication, de l'accès à la culture, de la diffusion de la connaissance. Alors que les applications citées précédemment n'en sont encore qu'au stade embryonnaire, ou qu'à l'état de concept pour certaines, les technologies numériques dans le domaine de la communication connaissent, quant à elles, un formidable essor depuis plusieurs années. L'évolution du multimédia est devenu source de nombreuses possibilités, notamment pour ce qui est des applications Web, dont le potentiel, en termes d'échange de l'information, est de plus en plus étendu. Certains sites rendent d'ores et déjà disponibles nombre de leurs œuvres sur la toile Internet, le plus souvent par de simples photographies, mais parfois également à l'aide d'outils plus interactifs, tels que la vidéo, la photographie panoramique ou l'imagerie de synthèse. Mais cette dimension de communication n'a pas pour seul objectif de rendre accessibles les collections aux internautes du monde entier. En effet, de nombreux musées n'ont actuellement pas la possibilité matérielle d'exposer la totalité de leurs collections en même temps. Prenons l'exemple du musée Van Gogh d'Amsterdam, qui n'expose en parallèle que cinquante tableaux sur les deux cents dont il dispose. Sans compter les cinq cents dessins et les sept cents lettres manuscrites, dont la correspondance de l'artiste avec son frère Théo, qui ne sont représentés que par quelques exemplaires seulement. Des bornes multimédia ont donc été mises à la disposition des visiteurs, apportant un indéniable complément à la visite grâce à un catalogue complet d'œuvres, une documentation détaillée, et une navigation interactive qui nous permet de laisser libre cours à notre soif de culture et d'information.

Ces «*technologies du futur*» (car c'est ainsi que nous les appelions hier encore) connaissent aujourd'hui une forte expansion, et cela grâce à des besoins sans cesse grandissants. Nous n'en avons cité ici que quelques exemples et, puisqu'il s'agit de celui qui nous a le plus concerné dans le cadre de ces travaux, nous nous sommes tout particulièrement attachés au domaine de l'art, en espérant que le lecteur aura pris conscience des enjeux particuliers qu'il suscite. Les problématiques et les questions qu'il soulève, parfois anciennes, parfois plus actuelles, font du monde de l'art et de ses acteurs des interlocuteurs privilégiés de ceux qui travaillent, développent et innovent avec l'outil numérique.

2 Le projet AMI-3D

■ 2.1 Présentation

L'essentiel de ces travaux de thèse a été réalisé dans le cadre d'un projet labellisé par le réseau ministériel *pour la Recherche et l'Innovation en Audiovisuel et Multimédia (RIAM)*, dont le but est de promouvoir l'innovation, notamment par le rapprochement des acteurs des secteurs publics et privés. Financé, via ce réseau, par les ministères de la Culture et de la Communication, de l'Industrie et de la Recherche, ce projet, intitulé *Archivage et Micro-Identification 3 Dimensions (AMI3D)*¹, a débuté en novembre 2004 et s'est achevé en février 2007. Il a fait l'objet d'une collaboration entre les trois partenaires suivants : le CRITT (*Centre Régional pour l'Innovation et le Transfert de Technologie*) HOLO3 basé à Saint Louis, fort de sa longue expérience en métrologie optique et de son savoir faire en termes de numérisation 3D, le laboratoire LIGIV de l'université Jean Monnet de Saint Étienne, spécialisé dans le traitement, la perception et la représentation de la couleur, et, pour finir, le laboratoire LSIIT de l'université Louis Pasteur de Strasbourg et son équipe d'Informatique Géométrique et Graphique, dont les principaux axes de recherche sont la modélisation géométrique, l'animation et la visualisation.

■ 2.2 Objectifs

L'objectif initial de ce projet est la conception de nouveaux outils de mesure, de restitution et d'archivage numérique d'œuvres d'art grâce au développement d'une technique capable d'intégrer, au sein d'un même dispositif, la numérisation de la forme, de la couleur et de la texture en termes de données métrologiques très fidèles. Ces outils ne doivent nécessiter aucune connaissance technique particulière de la part de l'opérateur chargé de la mesure. La prise de données doit, en effet, consister en un acte simple à réaliser, ce qui sous-entend une grande part d'automatisation.

La conception d'un tel outil vise trois enjeux spécifiques. En premier lieu, l'authentification des œuvres par une technique sans contact avec l'objet, et donc non destructive, et pour un coût acceptable par le marché. En second lieu, la réalisation de clichés exempts des droits d'auteur généralement liés à la reproduction d'œuvres en considérant que la prise de données, dans ce cas, n'est plus un acte de création artistique mais bel et bien un acte technique. Enfin, l'archivage de copies numériques très fidèles à des fins d'étude, de communication et d'accessibilité.

Notre travail, au sein du LSIIT, a porté sur l'acquisition de la forme et de la texture en vue d'une restitution réaliste des œuvres numérisées. Le distinguo ici fait entre couleur et texture s'explique ainsi : nous appelons *couleur* ce que l'on pourrait qualifier de teinte pure, c'est-

¹Référence du projet : *RIAM 04C292*

à-dire l'aspect que prend un matériau une fois épuré de tous les effets liés à l'illumination (réflexion, spécularité, etc.), alors que nous dénommons *texture* cette même apparence, mais en tenant sciemment compte cette fois des modifications engendrées par les changements d'éclairage ou de conditions d'observation. C'est cette seconde information qui, une fois mesurée, permettra de simuler de manière réaliste l'aspect de la copie numérique.

3 Problématiques

Parmi les contraintes qui nous ont guidées tout au long de ce projet, voici les plus importantes. La première, d'ordre purement pratique, concerne la technologie de mesure de formes employée. Nous avons dû travailler à partir des outils mis à notre disposition par HOLO3. Nous avons donc employé un *scanner à lumière structurée*, ce qui a motivé certains choix, comme nous le verrons plus tard. Une autre contrainte vient, comme nous l'avons déjà dit, du fait que la mesure ne sera pas nécessairement exécutée par un opérateur expert. La rendre accessible à un individu quelconque nécessite alors une grande part d'automatisation, quitte à déporter la plupart du travail en post-traitements, mais également la mise à disposition de moyens de contrôle interactifs permettant de s'assurer du bon déroulement de la mesure. Enfin, la troisième contrainte, et certainement la moins négligeable de toute, concerne le choix des objets d'intérêt. Étant donné que nous allons numériser des œuvres d'art, il est impératif de proposer une solution qui soit la moins destructive possible, c'est-à-dire qu'elle n'engendre pas, ou peu, de déplacements ou de contacts avec ces objets, souvent très fragiles. Il est donc préférable de déplacer le dispositif de numérisation vers le site de mesure plutôt que de déplacer la pièce elle-même, ce qui demande du matériel simple et mobile, ne nécessitant que peu de calibrage entre deux déplacements.

■ 3.1 Les contraintes de la mesure de formes

Il existe, à l'heure actuelle, de nombreuses technologies de mesure de formes. Néanmoins, quelle que soit la technologie considérée, les principes sous-jacents exploitent des procédés optiques et des algorithmes issus du domaine de la vision pour extraire une information de forme à partir d'une ou de plusieurs images. Qui parle d'images parle de capteurs, et l'usage de capteurs implique nécessairement la notion de point de vue, c'est-à-dire de positionnement du dispositif par rapport à la scène qu'il doit mesurer. En effet, les dispositifs de numérisation 3D peuvent, dans une certaine mesure, être assimilés à des appareils photographiques qui capturent non pas une information de couleur mais une profondeur. Une unique acquisition 3D ne constitue donc généralement qu'une mesure partielle de la forme de l'objet d'intérêt, étant donné que seul le côté visible depuis le point de vue courant peut être mesuré à un instant donné. Par conséquent, la numérisation complète d'un objet ne peut être obtenue que par un processus incrémental, durant lequel chaque nouvelle acquisition vient progressivement compléter la mesure globale de la forme.

Ce principe de numérisation pas à pas est en fait à la source des nombreux problèmes qui font de la mesure de formes une tâche longue et fastidieuse et qui empêchent aujourd'hui encore de tels dispositifs de fonctionner de manière autonome. Voyons en détail quels peuvent être ces problèmes.

Le recalage

Comme cela vient d'être évoqué, chaque acquisition permet d'extraire un morceau de surface qui ne correspond qu'à une sous-partie de l'objet complet. Chacun de ces morceaux de surface est généralement défini dans le référentiel du dispositif de numérisation 3D, référentiel qui change à chaque nouvelle prise de vue. Initialement, le positionnement des différents morceaux de surface les uns par rapport aux autres n'est donc absolument pas cohérent. Pour retrouver la forme de l'objet mesuré, il faut alors commencer par aligner ces morceaux de surface les uns par rapport aux autres en calculant les matrices de changement de repères permettant de passer d'une acquisition à l'autre. En somme, il s'agit de résoudre un grand puzzle dont chaque pièce provient d'une unique acquisition et constitue une sous-partie de la surface finale.

Dans la pratique, cette étape, dite de *recalage géométrique*, peut être résolue de deux manières. Dans le premier cas, il s'agit d'utiliser un dispositif de localisation qui permette, à tout moment, de connaître les déplacements du capteur 3D par rapport à sa cible. La détermination des transformations de recalage est alors immédiate. Mais de tels dispositifs apportent évidemment leurs propres contraintes, comme par exemple des besoins lourds en calibrage lors de chaque déplacement. Si l'outil n'est pas mobile, ce sont alors les objets à mesurer qu'il faut déplacer, ce qui constitue un prérequis inacceptable, comme nous l'avons déjà évoqué plus haut.

La seconde solution, quant à elle, consiste à effectuer le recalage de manière purement combinatoire. La problématique est alors de trouver des correspondances entre les différents morceaux de surface, c'est-à-dire à identifier les régions qui sont censées venir se superposer. Le recalage combinatoire a bien entendu l'avantage de relaxer les contraintes imposées par les dispositifs de localisation, et donc d'offrir une plus grande souplesse au niveau des conditions de mesure, mais il nécessite bien souvent de l'information *a priori* et de la redondance dans les données. Nous verrons effectivement que le principal défaut de la plupart de ces méthodes est le manque d'automatisme.

La planification

La numérisation pas à pas implique des temps d'acquisition qui peuvent être très variables d'un objet à l'autre, selon la complexité de la forme à mesurer. En effet, les surfaces présentant de nombreuses concavités et auto-occlusions nécessitent, *de facto*, un plus grand nombre d'acquisitions avant d'aboutir à une mesure complète. Pour l'opérateur chargé de la mesure, le choix de la prochaine prise de vue n'est donc pas sans conséquence : en mimi-

nisant les redondances lors de la capture, la numérisation complète d'un objet requerra un nombre plus restreint d'acquisitions. Un choix judicieux peut donc conduire à une réduction du temps global de la mesure, ce qui n'est pas sans intérêt lorsque l'on sait que certains objets de grandes tailles peuvent nécessiter jusqu'à plusieurs jours de mesure. De même, pour ce qui est des nombreux traitements qui suivront en vue de reconstruire une copie numérique exploitable, les calculs seront plus rapides s'il y a moins de données à traiter.

Il faut alors pouvoir estimer, à partir de l'information partielle disponible à un instant donné, quel est le point de vue qui est susceptible de maximiser la complétion d'information à l'occasion d'une acquisition supplémentaire. Cependant, comme ce problème de placement est fortement lié à la complexité de l'objet d'intérêt, la solution optimale n'est de loin pas triviale. Une solution logiciel pour l'automatisation de cette tâche de planification est donc primordiale pour garantir que le temps gagné par l'opérateur en termes de temps de mesure effectif ne soit pas perdu en intenses réflexions quant au prochain point de vue le plus judicieux.

Le filtrage

La numérisation 3D est avant tout autre chose un procédé de mesure qui est, à ce titre, sujet à l'incertitude. Ainsi, les dispositifs de mesure de formes fournissent, dans le cas général, une information bruitée : la copie numérique d'une surface réelle parfaitement lisse peut donner un maillage présentant une granulosité importante.

Au même titre que pour les images, il existe des algorithmes de débruitage pour les maillages qui consistent bien souvent à appliquer des opérateurs de convolution tendant à lisser localement la surface. Malheureusement, de la même manière que pour les images, l'application de vulgaires filtres passe-bas ne permet pas de faire la distinction entre ce qui est réellement du bruit et ce qui correspond aux fins détails du relief de la surface. Cela entraîne bien souvent une perte des détails géométriques, qui se traduit par un aplatissement des creux et des bosses ou un arrondissement des arêtes saillantes.

L'utilisation de filtres adaptatifs, qui ajustent l'intensité du filtre selon le comportement local de la surface, peut permettre d'éviter ce genre de problèmes. Malheureusement, ces filtres nécessitent bien souvent des paramètres complémentaires, généralement relatifs à l'intensité du bruit, quantité très difficile à obtenir dans la plupart des cas du fait de sa nature aléatoire. Le réglage de ces filtres se fait alors par tâtonnement, les meilleurs résultats n'étant obtenus qu'à l'issue de nombreuses tentatives.

L'intégration

Une fois que les différents morceaux de surface ont été réalignés et débruités, ils n'en demeurent pas moins déconnectés les uns des autres. En effet, à l'issue des traitements précédents, nous ne disposons pas d'une surface unique et fermée mais d'une multitude de surfaces dont la superposition nous donne la forme générale de l'objet réel dont elles sont

issues. Ces morceaux doivent donc maintenant être reliés les uns aux autres, en prenant bien soin de supprimer la redondance présente dans les zones de recouvrement.

Cette fusion des surfaces, appelée *intégration* dans le cas particulier de la numérisation 3D, ne consiste pas uniquement à créer des coutures pour unifier les maillages. Il s'agit également d'appliquer certaines corrections dans la mesure où, même après le recalage géométrique, les morceaux de surface ne sont pas nécessairement parfaitement alignés. En effet, le recalage ne fait que calculer le meilleur alignement possible, mais certaines aberrations optiques, comme la distortion radiale provoquée par les lentilles des dispositifs, font que les surfaces obtenues après numérisation peuvent être légèrement déformées par rapport aux surfaces réelles dont elles sont issues. Ces déformations ne peuvent être corrigées par des processus de filtrage comme ceux énoncés précédemment, puisque l'impact sur la géométrie est global alors que les opérateurs de débruitage travaillent généralement de manière locale sur les maillages.

Les algorithmes d'intégration doivent donc également mettre en œuvre des procédés de correction fonctionnant à un niveau d'échelle global.

■ 3.2 Les difficultés liées à la mesure de l'apparence

Avant d'aller plus loin, nous allons définir ce que nous entendons par *apparence*. Cette définition a déjà été brièvement introduite lorsque nous avons présenté le projet AMI3D. La question à se poser pour bien comprendre ce que nous cherchons à expliquer est la suivante : qu'y a-t-il à voir au-delà de ce que peut capturer une simple photographie ? Pour y répondre, il suffit de regarder les objets qui nous entourent. Leurs couleurs ne sont pas des teintes figées, constantes : selon la nature de l'éclairage environnant, la réponse d'un matériau varie. Pour le cas des surfaces brillantes, la couleur observée en un point dépend à la fois de l'environnement lumineux et de la position de l'observateur. Cette double dépendance, qui explique notamment les effets de reflets que présentent généralement de telles surfaces, traduit le caractère *bidirectionnel* d'un matériau, c'est-à-dire que sa réponse photométrique est fonction de deux directions : celle d'un rayon lumineux incident et celle d'un vecteur d'observation, partant de l'œil jusqu'à atteindre la surface. Sous ces considérations, il est clair qu'un seul cliché ne constitue pas une mesure suffisante.

L'acquisition

Le seul moyen d'obtenir une mesure précise de l'apparence est un échantillonnage exhaustif des points de vue et des éclairages possibles, c'est-à-dire une multitude de photographies prises en faisant varier ces deux conditions. L'environnement lumineux est alors contrôlé grâce à l'utilisation d'une source lumineuse la plus ponctuelle possible, afin d'avoir en chaque point de la surface un éclairage incident assimilable à un unique rayon.

Les données résultantes peuvent alors être exploitées de deux manières : soit par des méthodes *basées images*, qui combinent les photographies pour générer de nouveaux points de

vue ou pour rééclairer les objets directement en espace image, soit, comme dans notre cas, par des méthodes *basées maillages*, qui replaquent l'information d'apparence sous forme de texture sur un modèle 3D préalablement reconstruit. Les méthodes basées maillages ont l'avantage de permettre une liberté totale lors de la manipulation des copies numériques, chose impossible avec les méthodes basées images qui sont, bien évidemment, incapables de générer des vues radicalement différentes de celles fournies en entrée. L'information peut être interpolée, mais pas totalement réinventée. En revanche, avec les méthodes basées maillages, de nouveaux problèmes de mise en correspondance apparaissent : venir plaquer une photographie sur la description géométrique d'un objet n'est pas chose aisée.

Ce problème de recalage entre une image et un objet 3D revient en fait à déterminer les transformations permettant de projeter l'objet dans l'espace image de la photographie. Dans le cas de mesures réelles, ces transformations sont données par les paramètres optiques et les paramètres de pose de la caméra d'acquisition. L'ensemble de ces paramètres n'est généralement pas connu dans sa totalité (sauf si, comme nous l'avons suggéré plus haut, un dispositif de localisation suffisamment précis est disponible), et seule une procédure de *calibrage* peut alors permettre de les estimer. Le paradoxe est alors le suivant : une telle procédure demande à ce qu'on lui fournisse en entrée un certain nombre de correspondances entre des points 3D de l'objet et des pixels 2D de l'image, ce qui est justement l'information que nous cherchons à déterminer *in fine*.

Ce problème de mise en correspondance se pose d'ailleurs également pour la source lumineuse : il faut pouvoir déterminer sa position relative par rapport à l'objet pour estimer la direction d'incidence lumineuse en chaque point. Mais, dans ce cas, le problème est bien plus difficile à résoudre : une caméra fournit des données sur lesquelles il est possible de travailler pour établir des correspondances avec la géométrie de l'objet mesuré, alors qu'une source lumineuse ne fournit strictement rien, puisqu'il ne s'agit en aucun cas d'un capteur.

La restitution réaliste

L'intérêt de capturer de manière précise l'apparence d'un objet est, bien entendu, de pouvoir la restituer le plus fidèlement possible par la suite. Outre le réalisme, nous nous posons une contrainte supplémentaire : celle de l'interactivité. Nous visons, en effet, la diffusion d'œuvres d'art via des média comme Internet ou par l'intermédiaire de bornes multimédia, proposant donc à l'utilisateur la possibilité de manipuler lui-même la copie numérique. Pour cela, il n'est pas admissible de devoir attendre plusieurs minutes entre chaque rafraîchissement de l'image. On peut même dire qu'au-delà d'une image par seconde, il devient quasiment impossible d'interagir avec une application.

Dans le domaine de la synthèse d'images, trouver un équilibre acceptable entre interactivité et réalisme requiert généralement des compromis de part et d'autre. Cela est d'autant plus vrai lorsque, comme dans notre cas, les données proviennent de mesures réelles. Les masses de données générées par l'exhaustivité de la mesure sont en effet difficiles à exploiter lors d'un processus de rendu. De même, pour une application potentiellement destinée à diffuser des œuvres via la toile Internet, la compacité des copies numériques finales est un facteur

qui mérite d'être pris en compte.

Les données relatives à l'apparence doivent donc être compressées à l'aide d'un modèle approprié, qui permettra par la suite de reconstruire, en fonction des conditions d'éclairage et d'observation, tous les effets d'illumination préalablement capturés. Ce modèle doit satisfaire certaines contraintes. Tout d'abord, il ne doit pas être trop destructif vis à vis de l'information, afin que la compression ne se fasse pas au détriment du réalisme. Du point de vue de l'interactivité, le modèle doit également rester suffisamment simple pour ne pas exiger des temps de calcul trop importants, voire pour permettre son évaluation directe par le matériel graphique afin de bénéficier au mieux des dernières technologies en la matière.

4 Contributions

Au cours de nos travaux, nous nous sommes attachés à résoudre certains des problèmes qui peuvent survenir lors des différentes étapes de la chaîne de numérisation, jusqu'à aboutir à une copie numérique exploitable pour la visualisation.

Nous avons tout d'abord développé un protocole permettant d'unifier, au sein d'une même manipulation, la capture de la forme et de l'apparence dans le cas spécifique de pièces d'art capturées en environnements lumineux fixes. Nous nous sommes placés dans ces conditions particulières, où seules les variations imputables au changement de point de vue sont prises en compte, pour attaquer le problème, dans un premier temps, sous une approche plus simple que le cas extrêmement complexe de la capture et de la représentation du caractère bidirectionnel des matériaux. Ce protocole a pour objectif d'automatiser les deux types de recalage impliqués lors de ce genre d'acquisition (recalage géométrique pur et recalage entre images et géométrie), tâches qui, jusque là, nécessitaient bien souvent des interventions manuelles, comme nous le verrons au chapitre suivant. Il présente également l'avantage de mettre en œuvre un matériel très simple et très mobile, permettant ainsi d'effectuer des mesures sur site et non en laboratoire sans pour autant imposer les contraintes de calibrage généralement liées à ce genre de dispositifs. Ce dernier point est particulièrement important et justifie pleinement l'élaboration d'un tel protocole qui, rappelons-le, est destiné, à plus ou moins long terme, à numériser des œuvres d'art potentiellement très fragiles. Le principe de ce protocole sera détaillé au chapitre 6.

Nous proposons également une nouvelle approche utilisant la redondance dans les données d'acquisition pour estimer les paramètres requis par un filtre adaptatif lors de la phase de débruitage géométrique. En effet, le bruit étant de nature aléatoire (ou du moins majoritairement imprévisible), il est possible de calculer une bonne approximation de la surface théorique idéale par moyennage de plusieurs mesures différentes de la même zone de l'objet, et ainsi réduire l'influence du bruit sur les données. Le point fort de notre méthode réside dans le fait que ces mesures redondantes existent d'emblée, du fait que le recalage géométrique requiert impérativement du recouvrement entre les différents morceaux de surface

pour pouvoir les aligner. L'acte de prise de données n'a donc pas à être modifié en conséquence. Les zones de recouvrement peuvent ainsi être débruitées de manière fiable par l'exploitation judicieuse de l'information redondante. De manière à généraliser le débruitage à l'ensemble de la surface (y compris pour les régions ayant été mesurées sans redondance), nous utilisons le résultat de cette première correction partielle pour estimer les paramètres d'un filtre adaptatif que nous avons développé pour l'occasion. Cette première approche pour supprimer les interventions manuelles lors du processus de filtrage de maillages 3D numérisés sera présentée dans le chapitre 7.

Outre les traitements sur la géométrie, nous avons également abordé la problématique de la visualisation des copies numériques obtenues, toujours dans le cadre spécifique de pièces d'art capturées en environnements lumineux fixes, avec comme objectif un rendu réaliste temps réel. Nous utilisons pour cela un modèle très simple permettant d'exploiter le matériel graphique pour de meilleures performances. Le modèle en question, adapté de travaux bien connus dans la littérature, représente l'information chromatique à la surface d'un objet à l'aide d'un polynôme dépendant de la direction locale d'observation. Ainsi, la couleur de l'objet n'est plus une donnée fixe mais est associée à un *champ surfacique* dont la valeur dépend du point de vue sous lequel ce dernier est observé. Ce modèle sera présenté au chapitre 8 et nous discuterons de sa validité, compte tenu du fait qu'une telle représentation reste somme toute très basique par rapport à des modèles beaucoup plus aboutis.

Nous présentons également dans ce chapitre une extension du principe de champs surfaciques qui nous a permis de restituer de manière interactive et plausible le phénomène optique complexe qu'est la réfringence des matériaux translucides. Ce phénomène, dont le principe est régi selon des lois optiques qui seront rappelées ultérieurement, est celui selon lequel certains objets transparents imposent une déviation aux rayons lumineux qui les traversent. Plus prosaïquement, il s'agit de «l'effet loupe» observable lorsque l'on regarde à travers un verre d'eau, par exemple. Nous verrons que le rendu temps réel de ce phénomène n'est pas trivial en synthèse d'images. Nous avons proposé de représenter ce phénomène à l'aide d'un champ de distortion sur la surface qui, pour un rayon d'observation incident à la surface, renvoie non pas une couleur mais la direction du rayon sortant de l'objet après déflexion. Nous montrerons que cette méthode, bien que contraignante par certains côtés, permet une bonne approximation visuelle du phénomène de réfringence, même dans le cas de géométries complexes, ce qui constitue généralement la limitation des méthodes existantes.

Enfin, nous allons étudier, au cours du chapitre 9, l'élaboration d'une méthode dédiée. Si nous avons particulièrement insisté sur les problématiques liées à la numérisation 3D, c'est pour mieux illustrer le fait que cette mesure consiste en une chaîne extrêmement complexe, comportant de nombreux traitements ponctuels. Choisir préalablement la classe d'objets sur laquelle la mesure doit porter peut permettre de simplifier considérablement cette chaîne de traitements. Nous étudions donc le cas concret de la numérisation de tableaux d'art, catégorie d'objets dont les particularités peuvent permettre de faire fi de certains problèmes classiquement rencontrés dans le cas général. Nous proposons, en fait, un protocole complet allant de la numérisation à la visualisation de ces œuvres. Là où de nombreux travaux représentent la toile peinte à l'aide d'un simple plan lors du rendu, nous utilisons un algorithme

permettant de restituer de manière efficace et en tirant parti du matériel graphique le relief de la toile. Ce relief peut, en effet, être très important, comme dans le cas de la peinture au couteau. Les peintres eux-mêmes apposent de tels *empâtements* dans le but de capter la lumière et de jouer ainsi avec l'environnement. Ce relief ne doit donc pas être ignoré.

Ainsi, notre méthode commence par segmenter le relief en deux classes : une première partie considérée comme négligeable, et une seconde qui va comprendre toutes les régions de la toile dont le relief est suffisamment significatif pour être pris en compte lors du rendu. Nous proposons alors une technique hybride en associant à chacune de ces deux classes l'algorithme qui lui convient le mieux : une approximation très grossière pour la première, et un algorithme tenant réellement compte du relief pour la seconde. Ce dernier étant le plus coûteux des deux algorithmes, nous avons mis en place un mécanisme adaptatif permettant de ne l'utiliser que dans le cas où les conditions d'observation le requièrent, c'est-à-dire lorsque les erreurs de parallaxe deviennent trop importantes.

Du point de vue de l'apparence, nous nous sommes, cette fois, penchés sur une représentation bidirectionnelle complète. La simplification de l'acquisition géométrique induite par la restriction à une classe d'objets spécifique a notamment rendu possible la mesure de cette information. Pour la restituer au moment du rendu, nous avons estimé, à partir des mesures réelles, un modèle n'utilisant que peu de paramètres, toujours dans l'optique d'une évaluation temps réel par le matériel graphique.

Pour finir, nous montrons que les différents choix qui ont été fait concernant la représentation des données, autant pour la géométrie que pour l'apparence, apportent des avantages au niveau de la compacité de la copie numérique finale, grâce à une bonne compressibilité.

Première partie

— État de l'art —

De l'apport de nos pairs

- Mesure de formes -

Présentation de l'outil employé

L'outil de numérisation que nous avons été amenés à utiliser constitue en réalité le point de départ du projet AMI3D. Comme nous le verrons plus tard, une partie de nos travaux exploite certaines spécificités de cette technologie, et c'est pourquoi il est important, avant toute autre chose, d'en présenter le fonctionnement.

Il s'agit d'une technologie *sans contact*, c'est-à-dire que l'information géométrique est calculée par *téléométrie*. Un télémètre est un instrument capable de déterminer la distance d'un objet à l'aide de procédés acoustiques (*sonar*), radioélectriques (*radar*) ou encore, comme c'est le cas de ce que nous allons détailler ici, optiques. Les télémètres optiques passent par l'acquisition et l'analyse d'images pour effectuer une mesure. En informatique, ils ont été particulièrement étudiés dans les domaines de la vision et de la robotique, qui les distinguent en deux grandes catégories : la *vision passive* et la *vision active*. Les systèmes de vision passive travaillent directement et exclusivement à partir des images capturées alors que les systèmes de vision active apportent un complément d'information à la scène afin d'améliorer la mesure, généralement sous la forme d'une projection lumineuse contrôlée.

L'outil que nous avons employé entre dans cette dernière catégorie, puisqu'il s'agit d'un système de stéréovision à lumière structurée. Les principes élémentaires de vision nécessaires à la compréhension de son fonctionnement vont maintenant être présentés. Ainsi, nous allons d'abord parler, dans la section 1, des modèles mathématiques utilisés en vision et en synthèse d'images pour représenter le processus de formation d'images à partir d'une scène 3D. Être capable de représenter ce processus est un point essentiel de tout système de vision. Nous donnerons ensuite, en section 2, une brève description du fonctionnement du dispositif de numérisation que nous avons été amené à utiliser. Exposer les principes de ce dispositif nous sera en effet utile par la suite. Nous terminerons ce chapitre en expliquant rapidement quelles sont les formes que peuvent prendre les données fournies par un tel dispositif. Là encore, ces informations seront utiles à la compréhension des chapitres futurs.

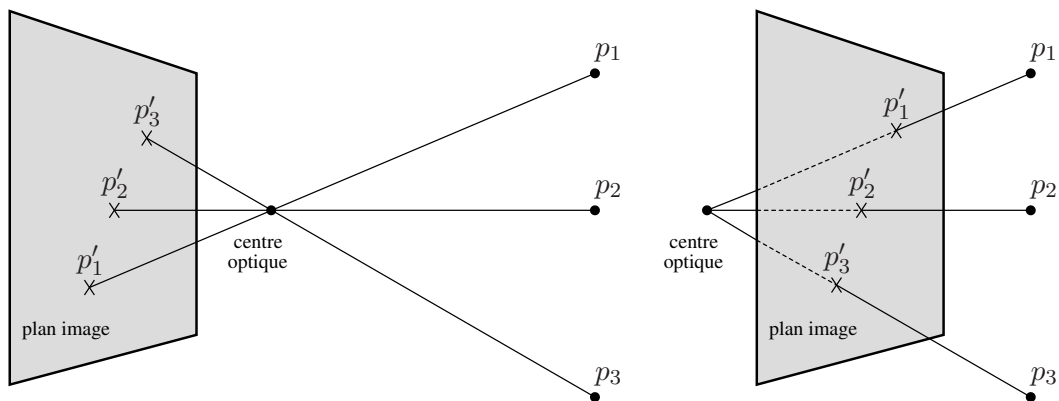


Figure 2.1 – **Gauche** : représentation géométrique d'un système optique à lentilles minces. **Droite** : représentation géométrique du modèle sténopé. Le centre optique est placé à l'arrière du capteur photosensible pour éviter une inversion de l'image lors du passage des rayons à travers la lentille.

1 Processus de formation d'images

Quelle que soit la méthode employée, mesurer une forme à partir d'images demande nécessairement de disposer d'une description géométrique du processus de formation d'images, c'est-à-dire d'un ensemble d'équations décrivant la façon dont un point de l'espace se projette sur le plan image du capteur lors d'une acquisition. Nous verrons effectivement que cette géométrie est importante dans la mesure où elle permet, en inversant le processus, de retrouver l'information géométrique à partir des seules données stockées dans les pixels. Nous allons donc commencer par présenter un modèle mathématique de caméra très largement utilisé dans la communauté.

■ 1.1 Le modèle sténopé

Un modèle de caméra doit proposer un ensemble de paramètres permettant de décrire les caractéristiques du capteur qu'il représente. Ces paramètres se distinguent selon deux catégories :

- les *paramètres intrinsèques*, qui font référence au montage optique du capteur (système de lentilles), c'est-à-dire au mécanisme de génération d'images en lui-même,
- les *paramètres extrinsèques*, qui représentent la pose du capteur, c'est-à-dire sa localisation par rapport à la scène qu'il mesure.

Le *modèle sténopé* est le modèle le plus communément admis en vision par ordinateur et en synthèse d'images pour représenter l'ensemble de ces paramètres. Il décrit un système optique simple à lentilles minces où l'ensemble des rayons lumineux provenant de la scène traversent une lentille, alors approximée par son seul centre optique, et viennent heurter un capteur plan photosensible (film argentique ou grille de cellules CCD). La représentation

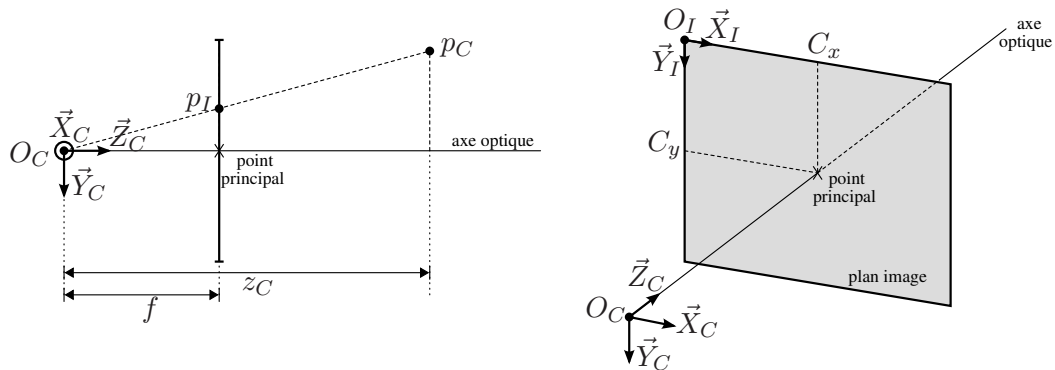


Figure 2.2 – **Gauche** : un repère local placé sur le centre optique et dont les axes sont alignés avec le plan de l'image permet de définir la projection perspective par des relations géométriques simples. **Droite** : le repère image n'est pas forcément de même unité que le repère caméra, de même que son origine n'est généralement pas confondue avec le point principal.

géométrique classique d'un tel système est illustrée sur le schéma de gauche de la figure 2.1.

Cette représentation, même si elle respecte l'agencement des éléments optiques tel qu'il est dans la réalité, présente néanmoins un inconvénient : elle fournit, sur le capteur plan, une image inversée de la scène. Pour pallier ce problème, le modèle sténopé est souvent représenté avec le centre optique à l'arrière du plan image, comme l'illustre le schéma de droite de la figure 2.1. Cette représentation géométrique, qui n'a, bien entendu, aucun lien avec la réalité, n'a été introduite que par commodité.

Considérant alors un repère local $(O_C, \vec{X}_C, \vec{Y}_C, \vec{Z}_C)$ associé à la caméra, dont l'origine O_C coïncide avec le centre optique, les axes \vec{X}_C et \vec{Y}_C sont alignés avec les axes principaux de l'image, et l'axe \vec{Z}_C est aligné avec l'axe *optique* (l'unique droite orthogonale au plan image et incidente au centre optique), ainsi que les coordonnées d'un point $p_C = (x_C, y_C, z_C)^T \in \mathbb{R}^3$ dans ce repère, il est possible de retrouver, grâce à des relations géométriques simples, les équations de projection perspective permettant de calculer les coordonnées $p_I = (x_I, y_I)^T \in \mathbb{R}^2$ du point p_C dans le plan image (voir figure 2.2 gauche) :

$$\begin{cases} x_I = \frac{f}{z_C} x_C \\ y_I = \frac{f}{z_C} y_C \end{cases} \quad (2.1)$$

où f est la distance orthogonale du centre optique au plan image, également appelée *distance focale*.

Ces équations très simplistes reposent sur trois hypothèses relatives au repère $(O_I, \vec{X}_I, \vec{Y}_I)$ de l'image. Tout d'abord, elles admettent que son origine O_I correspond au *point principal* de l'image, c'est-à-dire à l'intersection de l'axe optique avec le plan image. Ensuite, elles considèrent que les repères des espaces caméra et image sont de même unité, ce qui n'est pas forcément le cas dans la mesure où les unités de l'espace image correspondent aux

dimensions d'un pixel. Enfin, ces équations considèrent uniquement des pixels carrés, ce qui, une fois encore, n'est pas systématique.

Les coordonnées ainsi obtenues doivent donc subir un redimensionnement (qui peut être différent pour chaque dimension) ainsi qu'une translation en espace image afin de replacer chaque coordonnée par rapport à la véritable origine O_I (voir figure 2.2 droite) :

$$\begin{cases} x_I = g_x \frac{f}{z_C} x_C + C_x \\ y_I = g_y \frac{f}{z_C} y_C + C_y \end{cases} \quad (2.2)$$

où g_x et g_y sont les *facteurs de grossissement* selon chacun des axes \vec{X}_I et \vec{Y}_I de l'image, et (C_x, C_y) sont les coordonnées du point principal dans le repère image.

En exprimant le point p_I à l'aide des coordonnées projectives $(tx_I, ty_I, t)^T \in \mathbb{P}^2(\mathbb{R})$, il est possible de poser les équations précédentes sous la forme d'un système d'équations linéaires :

$$\begin{bmatrix} tx_I \\ ty_I \\ t \end{bmatrix} = C_I \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \quad (2.3)$$

où C_I est la *matrice des paramètres intrinsèques* définie ainsi :

$$C_I = \begin{bmatrix} G_x & 0 & C_x \\ 0 & G_y & C_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} g_x & 0 & C_x \\ 0 & g_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Tous ces calculs de projection ne sont valables que s'il est possible d'exprimer la scène dans le repère local de la caméra. Pour cela, il est nécessaire de connaître la transformation permettant de passer du repère global au repère $(O_C, \vec{X}_C, \vec{Y}_C, \vec{Z}_C)$. Cette transformation, qui représente la pose de la caméra et qui constitue l'ensemble des *paramètres extrinsèques*, peut être représentée sous la forme matricielle suivante :

$$C_E = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_x \\ r_{2,1} & r_{2,2} & r_{2,3} & t_y \\ r_{3,1} & r_{3,2} & r_{3,3} & t_z \end{bmatrix} = [R_{3 \times 3} \quad T_{3 \times 1}] \quad (2.5)$$

où R est une matrice de rotation et T un vecteur de translation. Le point p_C exprimé dans le repère caméra est alors obtenu en appliquant cette matrice au point $p_S \in \mathbb{R}^3$ correspondant dans la scène, exprimé sous la forme de ses coordonnées homogènes $(x_S, y_S, z_S, 1)^T \in \mathbb{P}^3(\mathbb{R})$:

$$\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = C_E \begin{bmatrix} x_S \\ y_S \\ z_S \\ 1 \end{bmatrix} \quad (2.6)$$

Lorsque le point de vue change, seuls les paramètres extrinsèques sont affectés. En effet, les paramètres intrinsèques ne dépendent que du montage optique de la caméra. Le découplage

des paramètres intrinsèques et extrinsèques en deux transformations distinctes peut donc s'avérer très intéressant. L'application successive de ces deux transformations permet de retrouver directement la projection image p_I d'un point p_S de la scène, et définit ce que l'on appelle la *transformation caméra* :

$$\begin{bmatrix} t x_I \\ t y_I \\ t \end{bmatrix} = C_I C_E \begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} \quad (2.7)$$

Ce modèle ne considère, pour le moment, que le cas d'un capteur idéal, ignorant certaines imprécisions imputables aux défauts de montage que l'on peut rencontrer dans le cas concret. Dans le cas de caméras numériques, par exemple, il se peut que la grille de cellules CCD ne soit pas parfaitement alignée. Cela produit une image dont les deux axes du repère semblent ne pas être orthogonaux (déformation parallélogrammique de l'image). Cette aberration peut être facilement prise en compte par le modèle sténopé en ajoutant un facteur de non-orthogonalité à la matrice des paramètres intrinsèques :

$$C_I = \begin{bmatrix} G_x & \mathbf{S} & C_x \\ 0 & G_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Il existe, par ailleurs, d'autres conséquences à ces défauts de montage. Notamment, l'usage de lentilles dans les systèmes optiques induit certaines distorsions qui, contrairement à l'exemple précédent, ne sont pas toujours représentables par un simple système linéaire. Le modèle sténopé doit donc être complété afin de permettre la correction de ces aberrations.

■ 1.2 Les distorsions optiques

Les défauts de lentilles introduisent dans les images des aberrations pouvant affecter la netteté, la chromaticité, mais également la géométrie de l'image. Il existe deux types d'aberrations géométriques : les *distorsions radiales* et les *distorsions tangentielles*.

Les *distorsions radiales* sont dues à l'épaisseur des lentilles. En effet, dans la réalité, l'approximation par lentilles minces est rarement valable. Ces distorsions engendrent une déviation des rayons lumineux qui tend à déplacer chaque point image le long de la droite qui le relie au point principal. Cela a pour effet de bomber l'image, soit vers l'intérieur, soit vers l'extérieur, selon le système de lentilles utilisé, comme l'illustre la figure 2.3-gauche. On observe souvent ce type de déformations sur des clichés pris à l'aide de systèmes à courtes focales.

Si l'on considère le point $p_D = (x_D, y_D)^T \in \mathbb{R}^2$ d'une image sujette aux distorsions, il est possible de retrouver le point idéal p_I par une simple translation en espace image :

$$p_I = p_D + \Delta_r \quad (2.9)$$

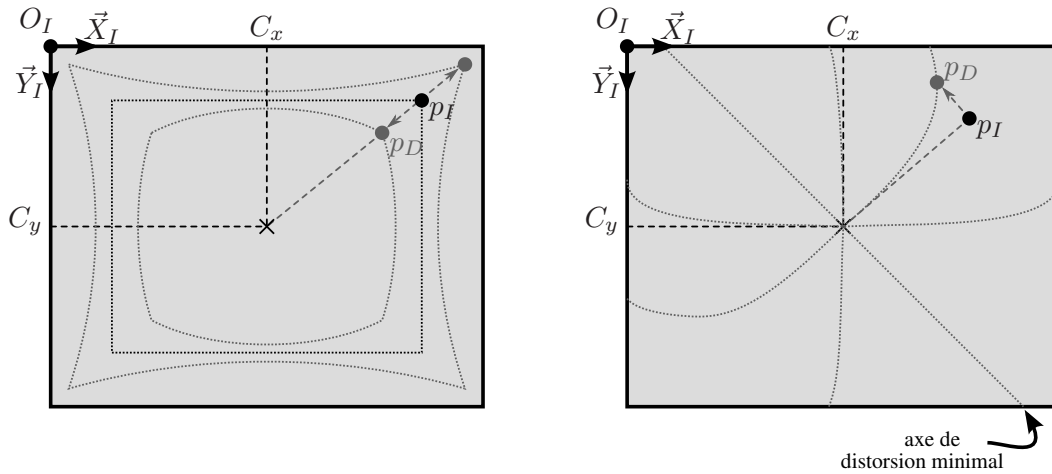


Figure 2.3 – **Gauche** : la distorsion radiale déplace un point le long du vecteur le reliant au point principal, bombant l'image vers l'intérieur ou l'extérieur selon le système de lentilles utilisé. **Droite** : la distorsion tangentielle déplace un point par rotation autour du point principal de l'image. L'amplitude de la distorsion dépend notamment de la distance angulaire du point par rapport à une droite le long de laquelle la distorsion est minimale.

où le vecteur de translation Δ_r est défini par :

$$\Delta_r = \bar{p}_D \sum_{i=1}^N \kappa_i r^{2i} \quad (2.10)$$

avec :

$$\bar{p}_D = \begin{bmatrix} \bar{x}_D \\ \bar{y}_D \end{bmatrix} = \begin{bmatrix} x_D - C_x \\ y_D - C_y \end{bmatrix} \quad \text{et} \quad r = \|\bar{p}_D\| \quad (2.11)$$

Les *distorsions tangentielles*, quant à elles, sont dues à un mauvais alignement des lentilles avec l'axe optique. Au niveau de l'image, cela se traduit par une légère rotation des points images par rapport au point principal. L'amplitude de cette rotation varie non seulement en fonction de la distance au point principal, mais également en fonction de la distance angulaire par rapport à une droite du plan image, appelée *axe de distorsion minimale*, le long de laquelle la distorsion tangentielle s'annule, comme l'illustre la figure 2.3-droite.

Comme précédemment, la distorsion tangentielle peut être modélisée par une simple translation en espace image :

$$p_I = p_D + \Delta_t \quad (2.12)$$

où le vecteur de translation Δ_t est défini ainsi :

$$\Delta_t = \left(1 + \sum_{i=3}^N \rho_i r^{2(i-2)} \right) \begin{bmatrix} r^2 + 2\bar{x}_D^2 & 2\bar{x}_D\bar{y}_D \\ 2\bar{x}_D\bar{y}_D & r^2 + 2\bar{y}_D^2 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad (2.13)$$

Les coefficients $(\kappa_1, \dots, \kappa_N)$ de distorsion radiale et (ρ_1, \dots, ρ_N) de distorsion tangentielle s'ajoutent alors au lot des paramètres intrinsèques définissant le modèle caméra.

Dans la pratique, il est souvent plus commode de travailler avec des images non distordues. Cela permet, en effet, de modéliser chaque vue à l'aide du modèle sténopé standard. Afin de satisfaire cette demande, bon nombre d'applications commencent donc par rectifier les images acquises à l'aide des coefficients de distorsions estimés.

■ 1.3 Calibrage

L'estimation des paramètres caméra à partir de données réelles dans le but de représenter un capteur existant constitue la phase de *calibrage caméra*. Il s'agit généralement d'algorithmes de minimisation qui nécessitent des correspondances 3D/2D entre certains points de la scène et leurs équivalents dans l'espace de l'image acquise par le capteur. De par leur nature, ces algorithmes d'étalonnage tendent à devenir moins précis sur l'estimation de chaque paramètre lorsque le nombre de ces derniers augmente. Ainsi, des simplifications sont souvent imposées au modèle. On peut citer, à titre d'exemple, les méthodes de Tsai [187] et de Zhang [210], qui ignorent toutes deux les distorsions tangentielles, les considérant comme négligeables, et qui ne représentent les distorsions radiales que par un seul coefficient (κ_1) pour la première, ou par deux coefficients (κ_1 et κ_2) pour la seconde.

Il existe de nombreux algorithmes de calibrage, certains étant dédiés à des situations spécifiques (courte focale, acquisition longue portée, etc.). Un survol des principaux développements dans ce domaine depuis les années quarante jusqu'à ce jour peut être consulté pour plus d'informations [30].

2 Lumière structurée par décalage de phase

■ 2.1 Triangulation optique et stéréovision active

Les systèmes de numérisation 3D par stéréovision s'apparentent, d'une certaine manière, au mécanisme de la vision humaine : ils exploitent l'information disponible dans plusieurs vues d'une même scène, acquises depuis des points de vue différents, pour retrouver l'information de profondeur manquante.

Pour ce faire, les systèmes stéréo exploitent le principe de la triangulation optique. Ce principe, illustré par la figure 2.4-gauche, est le suivant. Soient $\{V_i\}_{1 \leq i \leq N}$ un ensemble de vues d'une même scène, prises sous différents points de vue, et p un point physique visible depuis chacune d'elles et dont on cherche à déterminer la position dans l'espace. Ayant préalablement estimé les paramètres caméra d'une vue V_k , il est possible de construire une demi-droite partant du centre optique du capteur ayant servi à son acquisition et traversant le point image p_{I_k} sur lequel se projette p . Les coordonnées spatiales de p peuvent alors

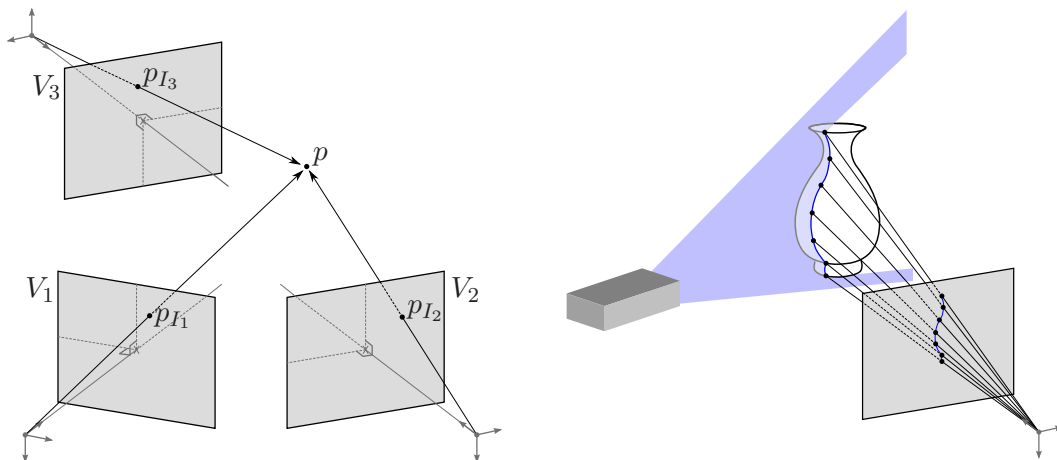


Figure 2.4 – **Gauche** : les coordonnées d'un point p visible depuis plusieurs images calibrées sont données par l'intersection des demi-droites partant des centres optiques des caméras d'acquisition et traversant les points image sur lesquels p se projette. **Droite** : extraction d'un profil de surface à l'aide d'un système de stéréovision active par franges.

être déterminées en cherchant l'intersection de toutes les demi-droites construites à partir de l'ensemble de vues initiales.

Lorsqu'il s'agit de faire de la mesure de formes avec des systèmes de stéréovision passive (systèmes ne travaillant qu'à partir d'images), la principale difficulté consiste alors à déterminer quels sont les pixels qui représentent le même point physique dans les différentes vues. Cette problématique de mise en correspondance est très ancienne dans le domaine de la stéréophotogrammétrie, et nous invitons le lecteur à explorer le survol publié par Dhond & Agarwal [40] sur le sujet.

Bien sûr, le principe de triangulation reste opérant lorsque le capteur associé à l'une des vues est remplacé par une source émettrice. La mesure consiste alors à projeter sur la scène, à l'aide de cette source, un motif lumineux qui va être ensuite identifié dans l'image acquise par le capteur. Les coordonnées 3D du point de la surface atteint par le motif peuvent, comme précédemment, être déterminées par l'intersection de deux demi-droites, l'une correspondant à la direction de projection lumineuse de la source, et l'autre au rayon quittant le centre optique du capteur et traversant le plan image à l'endroit où le motif a été détecté. Une technologie très répandue utilisant ce principe est la numérisation laser : le motif en question correspond au point lumineux généré sur la surface d'un objet lorsque celui-ci est intersecté par le faisceau. Ce type d'approches permet de s'affranchir des problèmes de mise en correspondance qui apparaissent lorsque l'on travaille exclusivement à partir d'images. On entre, dans ce cas, dans le domaine de la stéréovision active.

■ 2.2 Acquisition de franges et paramétrisation par phase

Certains dispositifs de vision active, tels que celui que nous avons employé, se basent sur la projection de franges pour effectuer une numérisation. Une frange lumineuse définit, en quelque sorte, un plan de coupe dans l'espace qui va permettre de récupérer tout un profil de l'objet à la fois plutôt qu'un point unique comme dans le cas du spot laser (voir la figure 2.4-droite). Pour acquérir l'ensemble d'une surface à la fois, ces dispositifs projettent alors plusieurs franges en même temps (idéalement, une par colonne de pixels d'un vidéoprojecteur). Il faut alors être capable de les distinguer l'une de l'autre au niveau du capteur, afin de savoir quelle frange est actuellement observée en un pixel donné. Toute l'idée consiste alors à définir, dans le domaine de projection, une paramétrisation permettant d'identifier chaque frange de manière unique. Il s'agit donc d'établir un codage lumineux cohérent de ces franges de manière à pouvoir recalculer, côté capteur, les valeurs de la paramétrisation uniquement à partir de l'information d'intensité lumineuse perçue dans les images.

Le scanner avec lequel nous avons travaillé utilise la phase ϕ d'une fonction sinusoïde $S(\phi) = I_m(1 + a \sin \phi)$ pour paramétrer les franges. Cette sinusoïde correspond à la fonction servant à encoder la variation lumineuse des franges qui sont projetées sur la scène, I_m représentant alors leur intensité moyenne et a l'amplitude de leur contraste. Lors de l'acquisition, l'intensité I perçue par le capteur dépend, bien entendu, de l'intensité de la frange (elle-même directement liée à la phase de la sinusoïde) mais également du matériaux et de l'orientation de la surface mesurée. Elle peut être décrite par l'équation suivante :

$$I = \rho S(\phi) \quad (2.14)$$

où ρ est un facteur dépendant de la valeur de réflectance de la surface par rapport aux positions relatives du projecteur et du capteur. La résolution des différentes inconnues intervenant dans cette égalité nécessite un système à plusieurs équations. Un tel système peut être obtenu grâce à l'acquisition de plusieurs images, chacune d'elles étant capturée en introduisant un déphasage $\delta\phi_k$ connu dans la sinusoïde projetée :

$$\begin{bmatrix} I_1 \\ \vdots \\ I_N \end{bmatrix} = \begin{bmatrix} \rho S(\phi + \delta\phi_1) \\ \vdots \\ \rho S(\phi + \delta\phi_N) \end{bmatrix} \quad (2.15)$$

Le codage des franges est donc à la fois spatial et temporel.

La résolution de ce système permet finalement de calculer la valeur de phase de la sinusoïde projetée en chaque pixel de l'image d'où les franges sont visibles.

■ 2.3 Calcul de profondeur par décalage de phase

Pour le calcul de la géométrie, les valeurs de profondeur peuvent être dérivées directement à partir des valeurs de phase mesurées. La figure 2.5 met en évidence la relation qui existe entre ces deux quantités. Sur ce schéma, θ représente l'angle entre la direction de projection

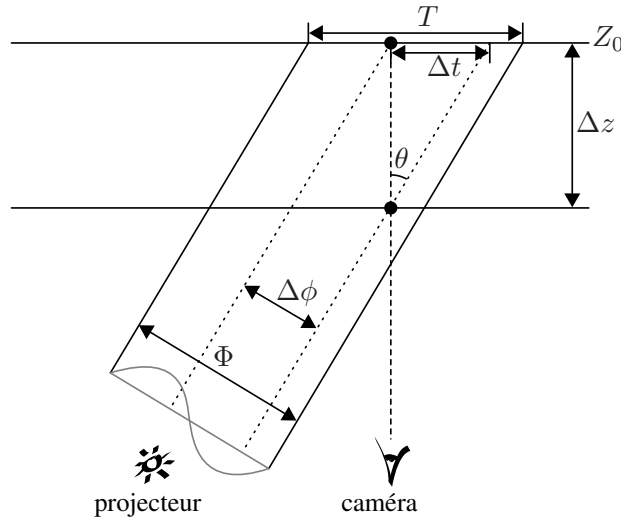


Figure 2.5 – Relation entre profondeur et phase.

et la direction d'observation, Φ la période de la sinusoïde projetée et T la période de la sinusoïde observée. Notamment, on peut observer que, lorsque la distance d'un point à la caméra change, la phase de la sinusoïde observée en ce point augmente ou diminue, selon qu'il s'en rapproche ou qu'il s'en éloigne. Connaissant la phase pour un plan de référence de profondeur Z_0 , il s'agit d'abord de calculer le déphasage $\Delta\phi$ engendré par le changement de profondeur :

$$\Delta\phi = 2\pi \frac{\cos\theta}{\Phi} \Delta t \quad (2.16)$$

où Δt représente le décalage de phase observé.

Le changement de profondeur Δz par rapport à Z_0 peut alors être calculé de la manière suivante :

$$\Delta z = \frac{\Delta t}{\tan\theta} = \Delta\phi \frac{\Phi}{2\pi \sin\theta} \quad (2.17)$$

Disposant de deux cartes de phases, l'une pour le plan de référence, préalablement déterminée par calibrage, et l'autre contenant l'objet à numériser, le déphasage $\Delta\phi$ peut être calculé par soustraction des deux images, et la distance de l'objet à la caméra peut alors être déterminée en chaque pixel grâce à la relation définie par l'équation 2.17. Pour une lecture plus aboutie sur la mesure de formes par décalage de phase, voir [80, 81, 160].

■ 2.4 Démodulation

Pour éviter toute ambiguïté dans le calcul de phase, la période Φ de la sinusoïde doit être au moins aussi grande que le domaine de projection, sans quoi les valeurs de phase peuvent se répéter dans l'image, modulo 2π . Malheureusement, la faible dynamique en niveau de gris des vidéoprojecteurs (seulement 256 valeurs en général) génère des paliers dans le signal de la sinusoïde qui rendent la qualité du calcul de phase très approximative, provoquant du bruit dans la reconstruction 3D.

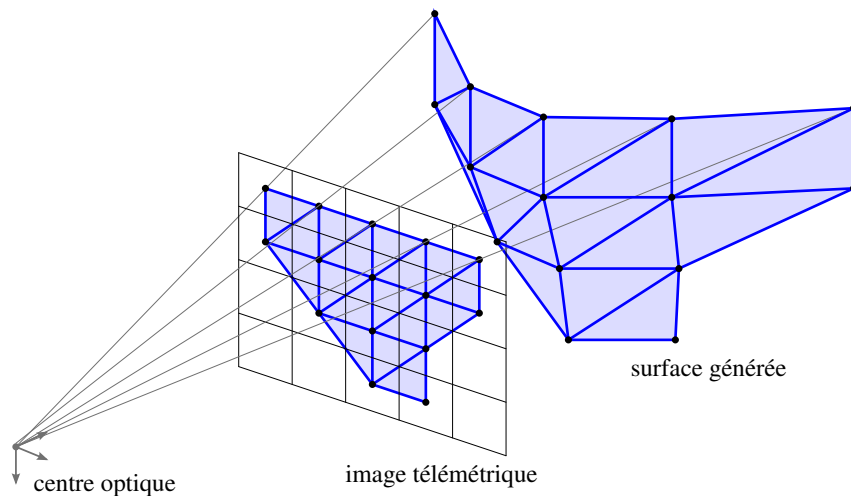


Figure 2.6 – Triangulation générée à partir de l’image télémétrique en considérant les pixels quatre par quatre. Les triangles sont créés de manière à relier les pixels contenant des points valides.

Pour limiter l’influence de ces paliers, il est nécessaire de choisir une période très petite. La phase obtenue doit alors être démodulée de manière à ce que chaque frange soit identifiée par une valeur unique. La démodulation d’une carte de phase peut se faire en consultant une autre carte, acquise à l’aide d’une fonction de période au moins deux fois plus grande. La numérisation se fait donc en projetant un signal sinusoïdal de période de plus en plus grande, chaque niveau permettant de démoduler le niveau précédent. Le processus s’arrête dès lors qu’une fonction de période suffisamment grande pour couvrir tout le domaine de projection a été projetée. Toute ambiguïté sur les valeurs de phase redondantes sont alors résolues et le calcul de géométrie décrit précédemment peut avoir lieu.

3 Représentation des données acquises

De nombreuses technologies de mesure de formes, y compris la nôtre, conservent la structuration 2D des données géométriques implicitement fournie par la grille de pixels de la caméra d’acquisition. L’information tridimensionnelle est donc fournie sous la forme d’une *image télémétrique*, c’est-à-dire une image dont chaque pixel contient un point 3D mesuré par rapport au repère local du capteur. On parle également, pour qualifier ce genre de données, de représentation $2^{1/2}$ D (deux dimensions et demi).

Par rapport à un nuage de points 3D totalement destructuré, cet agencement présente certains avantages, à commencer par le fait de posséder une connexité sous-jacente. De nombreux travaux [34, 159, 172, 173, 178, 189, 211] exploitent cette adjacence pour reconstruire, à moindre coût, une surface triangulée. Cette première approximation surfacique sert

souvent de base aux traitements ultérieurs que sont le recalage et l'intégration. Pour ce faire, l'image télémétrique est parcourue ligne par ligne, en considérant chaque carré de pixels adjacents, comme illustré sur la figure 2.6. Selon le nombre de points valides dans ces quatre pixels, des triangles sont générés ou non. Si les quatre pixels contiennent un point valide à la fois, la triangulation peut se faire selon deux schémas, en reliant l'une ou l'autre des diagonales. Le choix de la plus courte diagonale est généralement préféré, puisqu'il engendre des triangles d'aspect plus régulier.

Pour un point de vue donné, certaines parties de l'objet peuvent en masquer d'autres lors de l'acquisition. Du fait de ces problèmes d'occlusion, il est possible que deux pixels adjacents dans l'image télémétrique représentent des points de la surface très éloignés l'un de l'autre. Créer des triangles pour joindre de tels points n'est donc pas correct. Ce cas est traité de manière très simple, en fixant un seuil au-delà duquel deux points 3D ne sont plus considérés comme adjacents. Aucun triangle n'est alors créé entre ces points.

L'équivalence entre la triangulation obtenue par ce procédé et une triangulation de Delaunay a été démontrée par de précédents travaux [172], ce qui garantit une régularité optimale pour le maillage ainsi produit.

Puisque nous venons de montrer que les trois représentations sont disponibles moyennant quelques menus traitements, nous parlerons, par la suite, des données relatives à une acquisition en termes de *nuage de points*, d'*image télémétrique* ou de *surface* selon que l'agencement $2^{1/2}D$ a une importance ou non, ou que l'on a besoin d'une surface triangulée.

- Recalage géométrique -

Résolution du problème d'alignement de surfaces 3D

Dans le chapitre précédent, nous avons expliqué que l'outil de mesure de formes que nous utilisons ne fournit qu'une mesure partielle de l'objet à chaque acquisition, à savoir la partie visible depuis le point de vue courant, uniquement. La numérisation complète d'un objet doit donc se faire à l'aide de multiples acquisitions, résultant en un ensemble de surfaces disjointes et chacune définie dans son repère propre.

Nous allons donc aborder, dans ce chapitre, le problème du *recalage*, dont l'objectif est de retrouver les transformations permettant d'aligner ces surfaces les unes par rapport aux autres. Une fois de plus, il s'agit d'une problématique principalement étudiée dans le domaine de la vision, pour de nombreuses applications touchant à la reconnaissance de formes ou à la localisation spatiale de capteurs. Cependant, de nombreux travaux dans le domaine de l'informatique graphique se sont également penchés sur les problèmes posés par la numérisation 3D, dont le recalage. Il y a eu notamment quelques grands projets de déploiement sur site pour la numérisation du patrimoine culturel, comme le projet Michelangelo [98], la numérisation de la statue de David [19] et de la Pietà [12] à Florence, ou du Grand Bouddha de Kamakura [122] au Japon.

Intuitivement, l'utilisation de dispositifs permettant de déterminer la position du scanner à chaque acquisition [207], et ainsi de résoudre immédiatement les problèmes de recalage, est une idée qui semble assez séduisante. Mais ces dispositifs présentent cependant plusieurs inconvénients :

- il n'est pas rare que le recalage obtenu par ce biais n'atteigne pas la précision escomptée, pour diverses raisons, à commencer par les erreurs de calibrage (Chen *et al.* [27], par exemple, rapportent des erreurs d'alignement malgré l'usage d'une table rotative calibrée pour faire tourner l'objet d'une rotation connue devant le capteur),
- pour les plus précis d'entre eux, ces dispositifs peuvent également être très onéreux, ce qui les rend difficilement accessible au plus grand nombre,
- ces dispositifs apportent également des restrictions supplémentaires sur la mobilité du matériel d'acquisition, du fait des nécessités de calibrage à chaque déplacement. Pour des campagnes de numérisation comme celles énoncées plus haut, requérant un déploiement sur site, la mobilité du matériel est pourtant un critère important,

- enfin, le choix de la technologie de localisation dépend de la configuration du site en lui-même. On peut citer l'exemple de la numérisation de la Pietà, réalisée par Bernardini *et al.* [12], où le dispositif électromagnétique initialement prévu pour localiser le scanner dans l'espace a dû être abandonné, du fait des perturbations engendrées par la structure métallique présente dans les murs du musée.

Pour la plupart de ces raisons, aucun des projets précités de numérisation de grande envergure n'a utilisé de dispositifs de localisation, à l'exception, peut être, de cette tentative avortée de Bernardini *et al.*. Le recalage se fait le plus souvent manuellement, ce qui devient très vite fastidieux : lorsque le nombre de vues 3D est de l'ordre de plusieurs centaines, la résolution d'un tel puzzle peut demander quelques dizaines d'heures de travail humain. On comprend alors bien la nécessité d'algorithmes permettant l'automatisation de cette tâche.

Nous allons donc commencer par présenter le problème du calcul de transformations sous-jacent au recalage (section 1). Nous verrons ensuite comment cette transformation peut être déterminée de manière automatique grâce à la recherche de points communs entre les surfaces impliquées (section 2), puis nous présenterons des algorithmes de minimisation mis au point pour raffiner un premier recalage grossier, comme ceux obtenus par les algorithmes précédents ou à la suite d'un recalage manuel (section 3). Même si cela peut sembler être la méthode la plus immédiate pour recalcer un grand nombre de vues 3D, le fait de recalcer les surfaces deux à deux de manière incrémentale introduit souvent un biais. Nous discuterons alors de certaines approches plus globales qui prennent en compte l'ensemble des surfaces à la fois (section 4). Nous finirons par présenter, dans la section 5, les quelques approches génétiques qui, vu la complexité du problème de recalage, ont naturellement émergé pour tenter de le résoudre, puis nous tirerons des conclusions de l'ensemble de ces travaux (section 6).

1 Calcul de transformations rigides entre deux surfaces

■ 1.1 Énoncé du problème

Le passage d'une acquisition à l'autre se traduit par un repositionnement du capteur par rapport à l'objet. En théorie, les données acquises ne subissent ni déformation, ni changement d'échelle durant cette opération : toutes les distances sont conservées par rapport à la réalité. La transformation de recalage \mathbf{T} permettant de passer d'une position de capteur à une autre est donc une transformation rigide (isométrie), composée d'une matrice de rotation \mathbf{R} et d'un vecteur de translation \mathbf{t} . Dans la suite de ce document, la transformation de recalage sera donc représentée soit par la notation \mathbf{T} , soit par le couple (\mathbf{R}, \mathbf{t}) .

Si elle ne peut être fournie par une source extérieure (telle qu'un outil de localisation), la

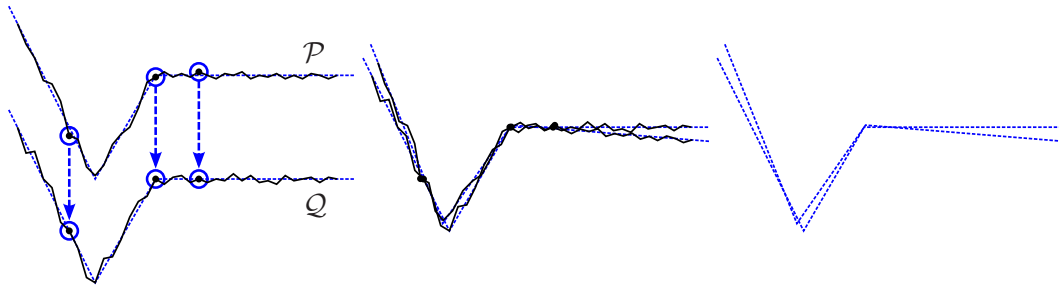


Figure 3.1 – Recalage dans le cas de données bruitées. N'utiliser qu'un nombre minimum de correspondances pour le calcul de la transformation rigide n'implique pas forcément un bon alignement entre les deux jeux de données.

transformation \mathbf{T} doit être estimée à partir des données elles-mêmes. Cela présuppose une certaine redondance dans les données : pour que deux surfaces puissent être recalées l'une par rapport à l'autre, elles doivent impérativement présenter du *recouvrement*, c'est-à-dire que les portions de l'objet physique qu'elles représentent doivent posséder une zone commune. Grâce à ce recouvrement, il est possible d'appareiller certains points deux à deux lorsque ceux-ci représentent le même point physique, générant ainsi un ensemble de *correspondances géométriques* entre les deux jeux de données. La rotation \mathbf{R} et la translation \mathbf{t} seront alors estimées à partir de cet ensemble de correspondances.

Dans le cas idéal où $\mathcal{P} = \{p_i\}_{1 \leq i \leq N}$ et $\mathcal{Q} = \{q_i\}_{1 \leq i \leq N}$ sont les ensembles de points appartenant à la zone de recouvrement commune à deux mesures distinctes, chaque couple (p_i, q_i) formant une correspondance, la transformation de recalage permettant de passer du repère de \mathcal{P} à celui de \mathcal{Q} peut être définie par :

$$q_i = \mathbf{R} p_i + \mathbf{t}, \quad \forall i \quad (3.1)$$

Comme \mathbf{R} est une matrice 3×3 et \mathbf{t} un vecteur à trois composantes, seules quatre correspondances (p_a, q_a) , (p_b, q_b) , (p_c, q_c) et (p_d, q_d) de points non coplanaires suffisent à les déterminer de manière unique. Mais ce calcul n'est correct que si l'ensemble \mathcal{Q} est l'exact transformé de \mathcal{P} , ce qui n'est jamais le cas dans la pratique :

- tout d'abord, deux points mesurés ne représentent jamais exactement le même point physique : la discrétisation de l'espace image du capteur implique que chaque point de mesure ne correspond pas à un unique point de l'objet, mais à un petit élément de surface correspondant à la projection du pixel dont il est issu. La probabilité que ces éléments de surface soient identiques pour deux échantillons différents est donc quasi nulle.
- ensuite, les données provenant de la numérisation 3D contiennent systématiquement du bruit, perturbant les positions des points de mesure.

Ainsi, calculer \mathbf{R} et \mathbf{t} à partir de seulement quatre correspondances risque de fournir une transformation qui n'est pas optimale vis-à-vis des deux surfaces prises dans leur globalité. Ce problème est illustré dans le cas 2D par la figure 3.1. Le calcul de la transformation doit donc tenir compte de toutes les correspondances à la fois.

■ 1.2 Solution optimale au sens des moindres carrés

La transformation rigide doit donc être estimée en recherchant les paramètres \mathbf{R} et \mathbf{t} qui minimisent la distance globale entre les deux ensembles de points \mathcal{P} et \mathcal{Q} . Cela revient à rechercher le minimum de la fonction d'erreur suivante :

$$\mathcal{E}(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N \|r_i(\mathbf{R}, \mathbf{t})\|^2 \quad (3.2)$$

où $\{r_i\}_{1 \leq i \leq N}$ est un ensemble de fonctions résiduelles telles que :

$$r_i(\mathbf{R}, \mathbf{t}) = q_i - \mathbf{R} p_i - \mathbf{t}, \quad \forall i \quad (3.3)$$

Arun *et al.* [2] présentent une approche permettant de résoudre ce problème grâce à une décomposition en valeurs singulières de la matrice de transformation.

Horn [78] propose, quant à lui, une formulation du problème utilisant un quaternion unitaire pour représenter la rotation. Le cheminement élégant de l'auteur montre, notamment, que les estimations des deux composantes de la transformation peuvent être découplées : la rotation est d'abord estimée à partir des ensembles de points exprimés localement par rapport à leurs centroïdes respectifs, puis la translation est finalement déduite une fois que la rotation est connue. Conscient du fait que bon nombre de chercheurs dans la communauté travaillent directement à partir de matrices, il étendra sa méthode ultérieurement pour fournir un résultat directement sous cette forme, sans avoir besoin de recourir à des conversions entre quaternions et matrices [79].

Walker *et al.* [191] proposent, eux aussi, une formulation à l'aide de quaternions, mais cette fois en nombres duaux. Sous cette forme, un quaternion est capable d'encoder à la fois la rotation et la translation. L'argument avancé par les auteurs pour justifier cette approche est que cette représentation conjointe permet d'estimer simultanément \mathbf{R} et \mathbf{t} , évitant ainsi la diffusion de l'erreur commise par Horn lors des calculs successifs de son découplage.

L'étude proposée par Lorusso *et al.* [105] montre, cependant, qu'il n'en est rien. En effet, les auteurs comparent les quatre méthodes précédentes selon plusieurs critères, notamment la précision et la vitesse d'exécution. Leurs résultats montrent que la décomposition en valeur singulière de Arun *et al.* et le quaternion unitaire de Horn semblent un peu plus précis que les autres méthodes, quelque soit le niveau de bruit présent dans les données. Le quaternion en nombres duaux de Walker *et al.* peut tout de même se targuer d'être le plus rapide à calculer lorsque la taille des ensembles \mathcal{P} et \mathcal{Q} devient importante ($> 10\,000$ points). Toutes ces différences restent néanmoins négligeables.

2 Recherche automatique de correspondances

Les méthodes précédentes de calcul de transformations rigides présupposent que les ensembles \mathcal{P} et \mathcal{Q} de points appareillés soient disponibles. Malheureusement, déterminer l'ensemble des correspondances de points entre deux jeux de données constitue justement le principal problème du recalage géométrique. En effet, la zone de recouvrement commune à deux mesures n'est généralement pas connue *a priori*, et les paires de points censés se superposer une fois le recalage accompli doivent être déterminées.

Il existe trois types d'approches pour trouver ces correspondances. La première consiste à utiliser des mires placées dans la scène [1]. Si ces mires ont l'avantage de fournir des correspondances immédiates, elles présentent également plusieurs inconvénients :

- il faut être sûr qu'un nombre minimum de mires se trouve dans le champs de vision du scanner lors de chaque acquisition,
- les mires risquent de masquer l'objet à l'endroit où elles sont posées, supprimant ainsi de l'information utile,
- si les mires doivent être posées sur l'objet lui-même, des risques de détérioration sont à redouter, surtout s'il s'agit d'œuvres d'art fragiles ou précieuses, comme dans notre cas.

Ce dernier point peut être levé en utilisant des mires projetées [12, 153], mais de nouveaux problèmes peuvent alors apparaître, comme par exemple des occlusions lorsque le scanner se trouve entre la source de projection et l'objet mesuré.

Dans le but de s'affranchir des contraintes imposées par l'usage de mires, les deux autres catégories de méthodes travaillent, quant à elles, directement sur les données pour en déduire les correspondances. Même si certains travaux s'y sont essayés [24], une recherche exhaustive visant à comparer chaque point d'une première acquisition à tous les points d'une seconde n'est absolument pas envisageable d'un point de vue combinatoire : les technologies actuelles de mesure de formes renvoient, en effet, plusieurs centaines de milliers (voire plusieurs millions) de points par acquisition. L'objectif est alors de restreindre l'espace de recherche en proposant des critères de similarité entre les points, ce qui permettra de rejeter très rapidement un grand nombre d'appariements invalides.

Ces approches se basent sur l'*extraction de caractéristiques géométriques* ou sur la définition d'*invariants par transformations rigides*. Nous allons les détailler ici.

■ 2.1 Extraction de caractéristiques géométriques

Le principe de l'extraction de caractéristiques est de rechercher les *éléments saillants* de la géométrie, tels que les coins ou les arêtes franches, dans la mesure où ceux-ci sont plus faciles à identifier mais également parce que le nombre de candidats potentiels pour la mise en correspondance est plus limité. En effet, rechercher l'homologue d'un point n'ayant aucune particularité distinctive revient quasiment à effectuer une recherche exhaustive. L'idée est

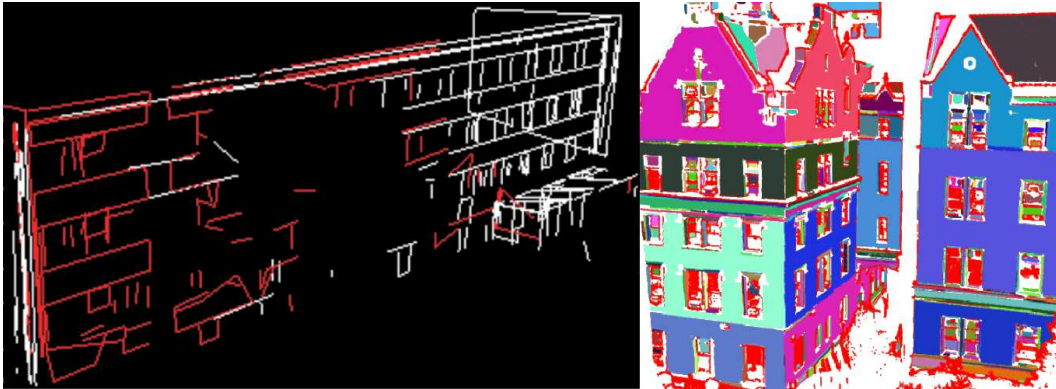


Figure 3.2 — Différents types de caractéristiques pouvant être extraits de données architecturales. **Gauche** : extraction d'arêtes franches (Zhang *et al.* [208]). **Droite** : extraction de régions planaires (Stamos *et al.* [174]).

donc de réduire la complexité combinatoire en recherchant les correspondances uniquement entre les points présentant les mêmes particularités géométriques.

Les méthodes entrant dans cette catégorie tirent souvent parti des spécificités de la scène. Ainsi, certaines d'entre elles [23, 174, 175, 208] ont été développées dans le cadre de la numérisation de scènes architecturales, pour lesquelles de telles caractéristiques sont souvent très présentes : grandes régions planaires, longues arêtes rectilignes et parallèles, etc. Zhang *et al.* [208], par exemple, proposent une méthode de recalage pour la numérisation en milieux urbains. Ils commencent par estimer des segments de droites à partir des points de forte discontinuité détectés dans l'image télémétrique acquise (figure 3.2-gauche). Lorsque deux de ces segments s'intersectent, un triplet $(s_1, s_2, p_{s_1 \leftrightarrow s_2})$ est formé, où s_1 et s_2 sont les deux segments et $p_{s_1 \leftrightarrow s_2}$ leur intersection. Riche d'informations telles que les longueurs de s_1 et s_2 ou encore l'angle qu'ils forment au point $p_{s_1 \leftrightarrow s_2}$, ces triplets sont alors utilisés lors des tests de concordance.

Stamos *et al.* [174, 175] considèrent le même genre de caractéristiques, mais à la dimension supérieure : chaque élément extrait est un triplet constitué d'un segment de droite et des deux régions planaires qui lui sont adjacentes (figure 3.2-droite). Des attributs supplémentaires, comme l'angle diédral entre les deux régions planaires ou leurs aires respectives, peuvent alors être pris en compte. Dans le but de concevoir un canevas complet pour l'extraction de caractéristiques de multiples natures, cette même équipe s'est également penchée, grâce aux travaux de Chen *et al.* [23], sur la mise en correspondance des éléments circulaires que l'on peut trouver sur certains types d'architectures.

De nombreux autres objets, comme des pièces industrielles par exemple, présentent des arêtes franches, mais celles-ci ne sont pas forcément rectilignes. Yang *et al.* [206] proposent de les représenter à l'aide de courbes paramétriques définies par des polynômes cubiques et exploitent les informations de courbure et de torsion pour tester leur similarité.

Outre la géométrie, il se peut que d'autres types de données puissent être exploités. De plus en plus, les outils de mesure de formes sont également capables de renvoyer une in-

formation de couleur associée à la géométrie. Dans le cas des scanners laser, par exemple, l'intensité du spot lumineux observé lors de la mesure peut être enregistrée en plus des coordonnées (x, y, z) du point. Certaines méthodes de recalage exploitent cette information supplémentaire pour extraire des caractéristiques d'ordre pictural à l'aide d'outils de traitement d'images, tels que des opérateurs de détection de coins ou d'arêtes [153, 201, 193]. La mise en œuvre de ces outils est très simple à réaliser, et elle permet d'extraire des éléments caractéristiques là où la géométrie seule n'y suffirait pas.

■ 2.2 Définition d'invariants par transformations rigides

Contrairement aux approches précédentes, qui extraient les éléments caractéristiques d'une surface directement à partir sa géométrie (ou de sa couleur), de nombreux autres travaux proposent de calculer des identifiants définis de manière indépendante de la pose de l'objet. Cette invariance par transformation rigide assure alors que deux points provenant de jeux de données différents mais représentant le même point physique se verront attribuer des identifiants similaires.

Ainsi, Li *et al.* [101] encodent dans le domaine fréquentiel la variation locale des normales dans le voisinage de chaque point à l'aide d'une transformée de Fourier discrète, et utilisent le vecteur de coefficients résultant comme invariant pour ce point.

Wyngaerd *et al.* [202], quant à eux, se basent sur les points bitangents, c'est-à-dire les paires de points appartenant à un même plan tangent de la surface, pour définir des paires de courbes fermées sur la surface de l'objet. Ces paires de courbes constituent alors un outil relativement général pour la problématique du recalage, puisqu'il s'agit en fait d'un invariant par transformation Euclidienne, transformation affine et transformation projective.

Dans le but d'utiliser des attributs chromatiques, Wyngaerd *et al.* [201] utilisent le contour généré par l'intersection avec la surface d'une sphère de rayon donné, centrée sur le point considéré. Le contour est discrétisé en un ensemble de points, et le vecteur contenant les couleurs de ces points est utilisé pour définir l'invariant.

D'autres travaux utilisent des bases définies localement en chaque point et permettant de représenter tous les autres points de la surface de manière indépendante de la pose de l'objet. Johnson *et al.* [85] partent d'un ensemble de points munis de leurs normales pour définir un nouveau système de coordonnées. Ainsi, au point p_i de normale n_i , tout autre point $p_j, j \neq i$, peut s'exprimer à l'aide d'un couple de coordonnées (α, β) tel que :

$$\begin{aligned}\alpha &= \sqrt{\|p_j - p_i\|^2 - \beta^2} \\ \beta &= (p_j - p_i) \cdot \vec{n}_i\end{aligned}$$

où α représente la distance de p_j à la droite de vecteur directeur n_i passant par p_i et β la distance orthogonale de p_j au plan tangent en p_i (voir figure 3.3-gauche). Le domaine des coordonnées (α, β) est alors discrétisé de manière à construire une image, appelée *image spin*. Initialement, les pixels de cette image ont tous une valeur nulle. Tous les points $p_j, j \neq i$,

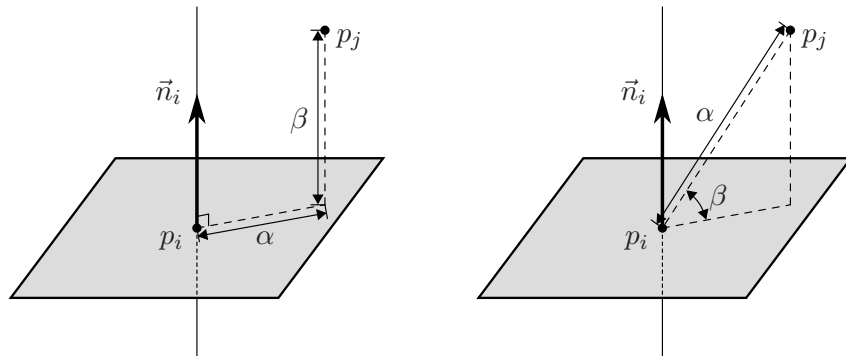


Figure 3.3 – *Gauche* : base locale utilisée lors de la génération de l’image spin [85] associée au point p_i . De nouvelles coordonnées (α, β) sont alors calculées pour tous les autres points $p_j, j \neq i$, du modèle. *Droite* : base proposée par Yamany *et al.* [205] sur le même principe.

de la surface autres que p_i sont alors projetés sur cette image par rapport à leurs coordonnées (α, β) , et chaque pixel atteint voit sa valeur augmentée de un. L’image spin constitue donc une sorte d’histogramme 2D de la surface, défini localement au point p_i . Une telle image est construite pour chaque point de chaque surface. Étant donné l’invariance de la base par transformation rigide, l’image spin associée à chaque point reste la même quelque soit la pose de l’objet. Les tests de compatibilité entre les points de deux mesures différentes peuvent alors être faits simplement à l’aide d’algorithmes de corrélation d’images.

Il est à noter, cependant, que si la densité d’échantillonnage n’est pas la même dans les jeux de données comparés, les images spin de points censés être appariés risquent de ne plus correspondre, du fait que le nombre de points se projetant en chaque pixel dépend nécessairement de cette densité. Carmichael *et al.* [21] ont montré que ce problème pouvait être résolu en projetant non plus les points, mais directement les triangles du maillage. La valeur de chaque pixel est alors augmentée de l’aire de la portion de surface qui s’y projette, rendant ainsi le calcul de l’image spin indépendant de la densité d’échantillonnage.

Yamany *et al.* [205] proposent une autre base, dans laquelle tout point p_j du modèle est représenté localement par rapport à un point p_i par :

$$\alpha = \|p_j - p_i\| \quad (3.4)$$

$$\beta = \cos^{-1} \left(\frac{p_j - p_i}{\|p_j - p_i\|} \cdot \vec{n}_i \right) \quad (3.5)$$

α représente alors la distance entre les deux points et β l’angle entre la droite qui les traverse et le plan tangent à p_i (figure 3.3-droite). La construction de l’image se fait cette fois en stockant, en chaque pixel, la moyenne des courbures des points qui s’y projettent, ce qui la rend moins sensible aux problèmes d’échantillonnage exposés plus haut.

■ 2.3 Caractéristiques et invariants multi-échelle

De manière générale, on peut remarquer que les éléments caractéristiques d'un objet correspondent à des points de gradient extrémal (gradient de couleur ou courbure géométrique). Cependant, dire si un extremum est pertinent ou non pour caractériser une surface est totalement dépendant de l'échelle sous laquelle cette dernière est considérée. Certaines méthodes [54, 91, 101, 177] s'attachent donc à définir des invariants ou à extraire des caractéristiques pour différents niveaux d'échelle afin de prendre en compte plusieurs ordres de grandeur à la fois lors de la mise en correspondance.

Pour détecter des caractéristiques géométriques à différents niveaux, Li *et al.* [101] proposent d'appliquer successivement des noyaux de lissage Gaussien de tailles croissantes sur la surface mesurée. Chaque noyau définit un niveau d'échelle, et les points qui sont les plus affectés lors de l'application d'un filtre sont considérés comme éléments caractéristiques pour le niveau correspondant. Cette méthode a également été utilisée en appliquant les filtres Gaussiens sur les attributs chromatiques plutôt que sur la géométrie [91], afin d'extraire des caractéristiques multi-échelle d'ordre pictural.

Gelfand *et al.* [54] considèrent une sphère centrée en chaque point et calculent le pourcentage de son volume qui est occupé par l'objet. Ainsi, les régions planes, concaves et convexes sont respectivement représentées par des pourcentages égaux, supérieurs et inférieurs à 50%, ce qui permet d'identifier les points de forte courbure. Des sphères de tailles croissantes sont alors utilisées pour construire les différents niveaux d'échelle.

Enfin, Sun *et al.* [177] proposent d'utiliser les contours générés en chaque point par projection sur le plan tangent d'un ensemble de cercles géodésiques concentriques de rayons croissants. Ces contours, dont la forme varie en fonction de la courbure de la surface, définissent en quelque sorte une *empreinte digitale* pour chaque point.

■ 2.4 Détermination de la transformation de recalage

Dès lors que des caractéristiques ont été extraites, que ce soit de manière purement géométrique, chromatique, ou par l'intermédiaire d'un invariant, un degré de similarité peut alors être déterminé entre deux points. Ce degré n'est cependant qu'un indice quand au fait que les deux points représentent le même point physique, mais certainement pas une réponse claire et définitive. Il permet seulement de réduire la complexité du problème en évitant de tester les paires de points pour lesquelles le degré de similarité tombe en deçà d'un certain seuil.

Un ensemble de transformations rigides probables peut donc être déterminé à partir des appariements les plus fiables. La plupart des méthodes précédentes utilisent alors une tactique de vote majoritaire pour déterminer laquelle de ces transformations correspond à la transformation de recalage recherchée : chacune des transformations est testée, et le nombre de points de la première surface amenés à proximité de la seconde est comptabilisé. La transformation remportant le plus haut score est choisie comme transformation de recalage.

Tester ainsi toutes les transformations probables pouvant s'avérer très coûteux en temps de calcul, Tarel *et al.* [180] proposent d'effectuer une classification floue de cet ensemble de transformations à l'aide d'un algorithme de type *K-means*. La transformation de recalage est alors choisie comme étant le noyau de la classe la plus grosse.

Enfin, pour être sûr de lever toute ambiguïté possible, le choix de la transformation finale peut également être laissé à l'utilisateur dans le cas d'outils semi-automatiques d'aide au recalage [22].

■ 2.5 Caractérisation globale

Toutes les méthodes qui ont été présentées jusqu'ici consistent exclusivement à caractériser chaque point indépendamment. Comme nous venons de le voir, cela ne permet pas de déterminer immédiatement la transformation de recalage, mais uniquement de restreindre l'ensemble des possibilités. Il existe cependant des méthodes cherchant à caractériser l'ensemble d'une surface à la fois. Ainsi, dès lors que la mise en correspondance est achevée, la solution finale est obtenue.

Zhang *et al.* [209] utilisent, par exemple, des cartes harmoniques pour déplier l'objet 3D sur un disque. Une image est alors générée en attribuant à chaque sommet du maillage déplié un niveau de gris dépendant de sa courbure dans le maillage initial. Comme dans le cas des images spin, des algorithmes de corrélation d'images peuvent être utilisés, mais cette fois-ci pour une mise en correspondance globale et non plus point à point. Les surfaces à recalage doivent cependant avoir la même topologie qu'un disque pour que l'approche fonctionne, ce qui n'est malheureusement pas toujours le cas dans la pratique.

On peut également trouver, dans cette catégorie de méthodes, des techniques basées sur la paramétrisation de maillages sur une sphère. L'objectif est alors de réduire le nombre de degrés de liberté en supprimant la composante translationnelle du problème de détermination de pose. On peut ainsi trouver des travaux sur les *images gaussiennes étendues (EGI)* [77, 89] ou sur d'autres types de paramétrisations sphériques [75]. Cependant, ces techniques ont été principalement développées dans le cas de la reconnaissance de formes, c'est-à-dire lorsqu'un morceau de surface acquis doit être recalé sur un modèle CAO complet de l'objet. Le problème est différent en ce sens que la paramétrisation de l'une des deux surfaces couvre l'ensemble du domaine, et l'extension au recalage de deux surfaces partielles n'est pas toujours triviale. De plus, seuls les objets ayant la topologie d'une sphère peuvent être traités grâce à de telles paramétrisations, restreignant une fois encore la classe des objets représentables.

3 Résolution par minimisation itérative

Dans cette section, nous allons présenter des algorithmes de minimisation proposés dans le but de déterminer incrémentalement la transformation de recalage. Ces algorithmes donnent généralement des résultats très précis, mais, comme la plupart des méthodes de minimisation, ils peuvent tomber dans des minimums locaux, et sont donc très sensibles aux conditions initiales. Ainsi, ils sont généralement utilisés pour raffiner un premier recalage, obtenu soit manuellement, soit par l'intermédiaire des méthodes présentées dans la section précédente.

Nous allons commencer par décrire l'algorithme en lui-même, tel qu'il a été initialement défini dans la littérature (section 3.1). Nous verrons ensuite que les sources d'incertitude liées à la numérisation 3D peuvent nuire au bon déroulement de cet algorithme. Nous présenterons donc certaines variantes proposées dans le but d'en améliorer la convergence et de le rendre plus robuste face à ces sources d'erreurs (section 3.2). Comme cela sera expliqué ci-dessous, les algorithmes de recalage par minimisation se basent sur la notion de *plus proches voisins* pour appairer des points entre eux. Or, la recherche du plus proche voisin d'un point dans un jeu de données est un problème complexe d'un point de vue combinatoire, et qui constitue le véritable goulet d'étranglement de ces algorithmes. Nous présenterons donc quelques méthodes mises au point pour accélérer cette recherche (section 3.3). Enfin, nous verrons que des attributs supplémentaires, tels que de l'information chromatique associée à la géométrie, peuvent être utilisés pour guider davantage le processus de minimisation vers le minimum global (section 3.4).

■ 3.1 Description de l'algorithme

Les deux travaux fondateurs par rapport aux algorithmes de recalage par minimisation sont ceux de Chen & Medioni [27] et de Besl & McKay [13]. Ces deux approches ne diffèrent qu'à un calcul métrique près, comme nous le verrons plus loin dans cette section, mais la description générale de l'algorithme est la même.

Considérons les deux jeux de données à recaler \mathcal{P} et \mathcal{Q} . À chaque itération, l'algorithme considère le plus proche voisin $q_i \in \mathcal{Q}$ d'un point $p_i \in \mathcal{P}$, par rapport à un critère métrique donné, comme étant la meilleure approximation de correspondance pour ce point. L'ensemble \mathcal{P} est alors recalé sur \mathcal{Q} en conséquence, ayant pour effet d'attirer progressivement les deux surfaces l'une vers l'autre à mesure que le nombre d'itérations augmente. Le processus est alors répété jusqu'à convergence.

L'algorithme peut donc être résumé par les quatre étapes suivantes :

1. **Recherche de plus proches voisins** : pour un critère métrique $d(\cdot, \cdot)$ donné, déterminer pour tout point $p_i \in \mathcal{P}$ le point $q_i \in \mathcal{Q}$ qui soit tel que :

$$d(p_i, q_i) \leq d(p_i, q_j), \quad \forall j \neq i. \quad (3.6)$$

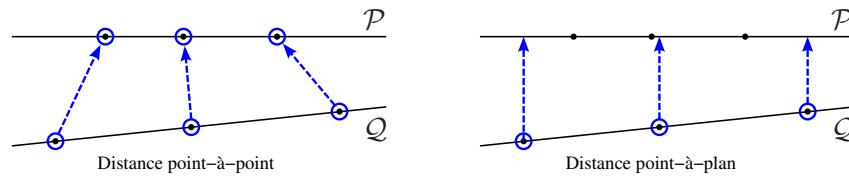


Figure 3.4 – Stabilité de la distance au plan tangent de Chen & Medioni [27] par rapport à la distance point-à-point utilisée par Besl & McKay [13].

2. **Calcul de la transformation rigide** : déterminer la transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ minimisant la distance entre les deux surfaces au sens des moindres carrés, représentée par la fonction d'erreur suivante :

$$\mathcal{E}(\mathbf{T}, \mathcal{P}, \mathcal{Q}) = \sum_{i=1}^{\text{card}(\mathcal{P})} d(q_i, \mathbf{R}p_i + \mathbf{t})^2 \quad (3.7)$$

3. **Recalage** : appliquer la transformation \mathbf{T} à l'ensemble \mathcal{P} .
4. **Test de convergence** : répéter les étapes 1 à 3 jusqu'à ce que le critère de convergence proposé soit atteint.

Besl & McKay ont introduit le nom d'*Iterative Closest Point (ICP)* pour décrire leur algorithme. Ce nom est devenu une dénomination très largement employée dans la communauté pour décrire, dans le cas général, tous les algorithmes de recalage par minimisation itérative. Nous l'utiliserons donc également dans la suite de ce document lorsque nous parlerons d'algorithmes de ce genre.

Comme cela a été dit plus haut, les deux travaux fondateurs précités se distinguent l'un de l'autre par le critère métrique employé : Besl & McKay utilisent une distance Euclidienne point-à-point, alors que Chen & Medioni utilisent la distance du point source au plan tangent de son homologue. Si la méthode de Besl & McKay est plus simple à mettre en œuvre et plus rapide à calculer, la distance point-à-plan est néanmoins reconnue comme étant plus stable, et permettant une convergence plus rapide.

La différence de stabilité entre les deux méthodes est illustrée par la figure 3.4, qui montre l'exemple du recalage de deux surfaces planes ayant des densités d'échantillonnage différentes. La distance point-à-point amène à un résultat oscillant, dans la mesure où le plus proche voisin sera tantôt d'un côté, tantôt de l'autre, alors que la distance point-à-plan ne se soucie que de tirer la surface \mathcal{P} vers la surface \mathcal{Q} dans la direction orthogonale au plan.

Concernant le critère de convergence de ce genre d'algorithmes, la plupart des travaux se contentent de fixer une valeur plancher pour la fonction \mathcal{E} à minimiser. Un critère d'arrêt alternatif a été proposé [53], visant à tester si la transformation de recalage \mathbf{T} devient proche de la transformation identité. L'avantage de ce critère est qu'il ne dépend pas de l'ordre de grandeur des jeux de données considérés.

■ 3.2 Amélioration de la convergence et de la robustesse

Les algorithmes de type ICP sont bien loin de se placer dans un cas idéal lorsqu'ils sont appliqués au recalage de données mesurées : la numérisation introduit du bruit dans les données, comme nous l'avons déjà évoqué, mais également des distortions plus globales sur la surface acquise (écrasement, étirement, etc.), du fait des imprécisions de calibrage. Toutes ces sources d'incertitude peuvent engendrer des instabilités dans le processus de minimisation, et donc altérer la qualité du recalage obtenu.

Afin de prendre en compte le bruit de numérisation, Dorai *et al.* [41] proposent d'introduire, dans la fonction de mesure d'erreur à minimiser, un poids σ_i dépendant de la variance locale de l'erreur pour chaque correspondance (p_i, q_i) :

$$\mathcal{E}(\mathbf{T}, \mathcal{P}, \mathcal{Q}) = \sum_{i=1}^{\text{card}(\mathcal{P})} \left(\frac{d(q_i, \mathbf{T}p_i)}{\sigma_i} \right)^2 \quad (3.8)$$

D'autres pondérations de ce genre ont été introduites afin de tenir compte d'autres sources d'erreur potentielles, comme, par exemple, le fait que les parties de la surface acquises sous des angles rasants sont davantage sujettes à l'incertitude que les parties numérisées de front [58, 158].

Les distortions globales, en revanche, sont plus difficiles à prendre en compte que le simple bruit de numérisation. En effet, si celles-ci sont trop importantes, une transformation rigide ne suffit plus à définir le recalage entre deux surfaces. Il faut dans ce cas faire appel à des méthodes *non-rigides* [17, 44, 84], qui appliquent des déformations aux surfaces pendant le recalage, dans le but de les faire coïncider le plus exactement possible. Ainsi, Fledmar *et al.* [44] proposent un algorithme qui estime une transformation affine au lieu d'une simple isométrie, et Ikemoto *et al.* [84] proposent de découper les surfaces engendrant les erreurs de recalage les plus grandes en morceaux plus petits, et d'effectuer un recalage pour chacun de ces morceaux comme s'il s'agissait d'une surface indépendante.

Un autre problème, pour le recalage rigide, vient du fait que le calcul de la transformation se fasse de manière consensuelle, comme nous l'avons vu dans la section 1, de manière à proposer un recalage qui contente au mieux toutes les paires de points fournies. Cela implique que la forme générale de l'objet a une contribution beaucoup plus importante dans la détermination de cette transformation que les fins détails de sa géométrie. Ainsi, si la forme générale ne permet pas de contraindre convenablement l'ensemble des degrés de liberté (objets fortement planaires, surfaces de révolution, etc.), le recalage peut alors aboutir à deux surfaces apparemment bien recalées, mais dont les fins détails géométriques ne coïncident absolument pas, comme l'illustre la figure 3.5-gauche.

Pour résoudre cette instabilité, Rusinkiewicz *et al.* [158] proposent de sélectionner les points en se basant sur leurs normales plutôt que de travailler sur un échantillonnage uniforme (ou sur la totalité) de la surface : les points sont alors choisis de manière à représenter un échantillonnage uniforme sur la sphère de Gauss. Ainsi, toutes les orientations que prend la surface sont équitablement représentées, qu'elles correspondent à des régions de grande

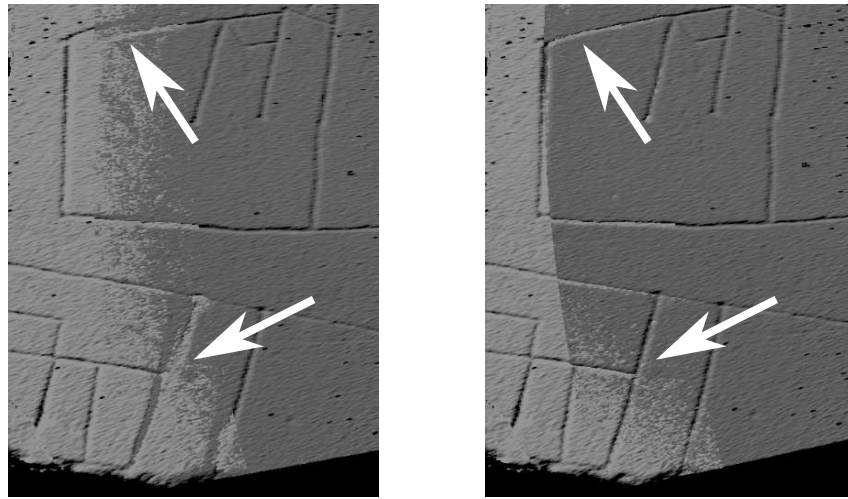


Figure 3.5 – Recalage de deux acquisitions du *Forma Urbis Romae*. **Gauche** : même lorsque la forme générale de l'objet semble bien recalée, les fins détails de sa géométrie peuvent ne pas coïncider parfaitement. **Droite** : résultat obtenu grâce à l'échantillonnage uniforme de l'espace des normales proposé par Rusinkiewicz *et al.* [158] pour résoudre ce problème.

taille ou non. Gelfand *et al.* [55] proposent une méthode permettant de détecter quels sont les degrés de liberté de la transformation à estimer qui ne sont pas bien contraints par les données, et ainsi de ne sélectionner que les correspondances permettant de préserver la bonne contriction du problème.

Ces deux méthodes donnent d'excellents résultats, fournissant un recalage dont les défauts d'alignement pour les détails géométriques sont parfaitement corrigés, comme on peut le voir sur la figure 3.5-droite.

■ 3.3 Problème du plus proche voisin

Par défaut, l'algorithme ICP est très gourmand en temps de calcul. En effet, l'étape de recherche des plus proches voisins constitue un véritable goulet d'étranglement : naïvement, trouver le plus proche voisin de $p_i \in \mathcal{P}$ dans l'ensemble $\mathcal{Q} = \{q_i\}_{1 \leq i \leq N}$ est un calcul en $\mathcal{O}(N)$, ce qui veut dire que le fait de trouver les homologues de tous les points de \mathcal{P} est un problème en $\mathcal{O}(N^2)$ (en considérant que les deux ensembles soient de cardinalités similaires). Or, comme nous l'avons déjà évoqué, les scanners 3D actuels peuvent renvoyer jusqu'à plusieurs centaines de milliers (voire plusieurs millions) de points par acquisition. Une telle solution n'est donc pas envisageable. Nous allons donc présenter ici des méthodes qui ont été développées dans le but de réduire la complexité combinatoire de ce problème.

Une première méthode simple, la *méthode d'Elias* [31], consiste à répartir les points de l'ensemble de recherche \mathcal{Q} dans une partition (régulière ou non) de l'espace. Les sous-espaces engendrés par cette partition sont alors parcourus dans l'ordre croissant de leur distance au point requête p_i . Pour chacun d'eux, le plus proche voisin de p_i est mis à jour

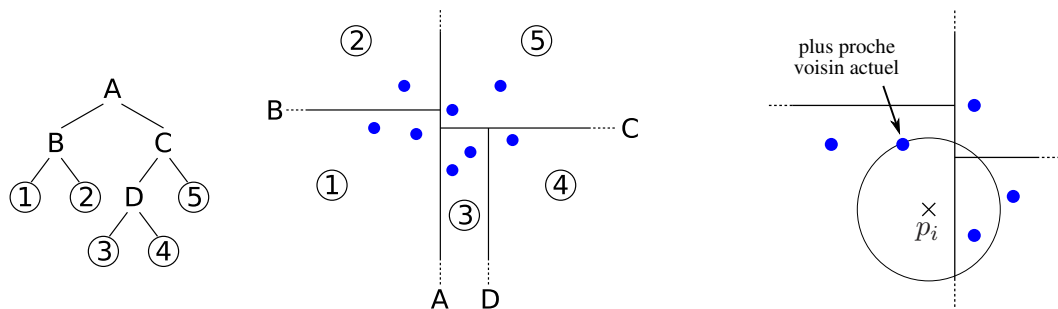


Figure 3.6 — Gauche : découpage spatial engendré par un kD -arbre. Droite : la sphère centrée sur le point requête et incidente à son plus proche voisin actuel est utilisée lors de la recherche pour détecter si d'autres feuilles doivent être visitées.

par une recherche exhaustive dans l'ensemble de points qu'il contient. Dès lors que le sous-espace actuellement examiné se trouve plus éloigné du point p_i que ne l'est son plus proche voisin actuel, le parcours s'arrête.

Bentley [9] est le premier à proposer le kD -arbre, qui généralise les arbres binaires de recherche à des espaces de dimensionnalité quelconque. Chaque nœud interne de cet arbre définit un hyperplan découpant les données en deux sous-ensembles selon l'une des dimensions. Chaque nœud feuille correspond alors à la région convexe de l'espace issue du découpage récursif induit par la descente dans l'arbre, et contient l'ensemble des points de \mathcal{Q} inclus dans ce sous-espace (figure 3.6-gauche). La recherche dans un tel arbre se fait alors en deux étapes :

- Une première descente est effectuée jusqu'à atteindre un nœud feuille, comme dans le cas des arbres binaires de recherche standards. Une recherche exhaustive parmi les points qu'elle contient permet alors de trouver une première estimation du plus proche voisin.
- L'arbre est ensuite remonté en vérifiant si le plan de coupe défini en chaque nœud intersecte ou non la sphère centrée sur le point requête et passant par son plus proche voisin actuel (figure 3.6-droite). Si c'est le cas, le sous-arbre de ce nœud qui n'a pas encore été inspecté est parcouru à son tour, et le plus proche voisin est mis à jour à chaque fois qu'une nouvelle feuille est atteinte.

Dans la version originale de Bentley, la dimension de partitionnement est définie cycliquement sur les niveaux de l'arbre. Friedman *et al.* [50] ont démontré plus tard qu'il était préférable de couper systématiquement au milieu de la dimension de plus grande dispersion. L'algorithme du kD -arbre est certainement le plus populaire en ce qui concerne la résolution du problème du plus proche voisin. Chaque requête se fait approximativement en $\mathcal{O}(\log N)$, ce qui en fait une solution performante.

Pour les algorithmes de type ICP, à part lors des toutes premières itérations, la position d'un point ne varie pas de manière drastique d'une itération à la suivante. De ce fait, le plus proche voisin d'un point n'est lui-même pas très éloigné de celui trouvé à l'itération précédente. Des mécanismes de cache pour l'algorithme du kD -arbre ont ainsi été développés [132, 168] afin d'exploiter cette cohérence temporelle pour accélérer encore la recherche. De même, Greenspan *et al.* [62] proposent un prétraitement consistant à déterminer un voisinage trié en chaque point de l'ensemble de recherche. La recherche commence alors dans le

voisinage du plus proche voisin trouvé à l'itération précédente. Si cette première recherche s'avère infructueuse, une méthode plus traditionnelle est finalement employée.

Plutôt que de rechercher les plus proches voisins exacts, certains auteurs se contentent d'approximations, souvent plus rapide à déterminer. Celles-ci ne sont alors utilisées qu'au cours des premières itérations, lorsque l'algorithme ne nécessite pas encore une précision trop importante. Ainsi, Greenspan *et al.* [63] utilisent un k D-arbre dont l'étape de remontée est ignorée. Dans les dernières itérations, ils reviennent tout de même à une recherche exacte afin de garantir une bonne convergence. L'approche de Yamany *et al.* [204] utilise une grille 3D dont chaque voxel contient la distance et la direction du point de Q le plus proche de son centre. Cette méthode permet donc une recherche en temps constant, mais présente trois inconvénients, qui sont le temps de construction d'une telle grille, sa consommation mémoire, et le fait que la précision de la recherche dépende de sa résolution. Concernant ce dernier point, les auteurs mettent en place une grille hiérarchique qu'ils affinent aux abords de la surface Q .

Lorsque le dispositif d'acquisition fournit les deux jeux de données \mathcal{P} et Q sous la forme d'images télémétriques, c'est-à-dire sous la forme d'images de points 3D (voir section 3), il est possible d'exploiter les informations de calibrage du capteur pour déterminer une approximation du plus proche voisin en temps constant [14, 157]. Connaissant les paramètres intrinsèques du capteur et la transformation de recalage obtenue à l'itération précédente, tout point requête $p_i \in \mathcal{P}$ peut être reprojété sur le plan de l'image télémétrique de Q , et le point contenu dans le pixel atteint est alors utilisé pour l'appariement. Ainsi, le problème du plus proche voisin n'est pas considéré en termes de vraie distance Euclidienne, mais en termes de plus courte distance le long des droites d'observation du capteur. Benjemaa *et al.* [8] reprennent ce principe en utilisant des tampons de profondeurs générés à partir de points de vue virtuels. Ils n'ont ainsi pas besoin de connaître les paramètres de calibrage pour effectuer la reprojektion. Comme l'expliquent Blais *et al.* [14], ces méthodes sont intéressantes dans la mesure où l'approximation par reprojektion tend vers la solution exacte à mesure que la distance entre les deux surfaces diminue.

■ 3.4 Utilisation d'attributs additionnels

Nous avons vu dans la section précédente, concernant la recherche automatique de correspondances (section 2), que des attributs complémentaires pouvaient être associées à la géométrie, qu'il s'agisse de données calculables à partir de cette dernière ou de données chromatiques directement fournies par le scanner. Nous allons donc voir comment ces informations peuvent alors être utilisées à bon escient pour orienter le processus de minimisation vers la solution optimale, et réduire ainsi les chances de voir l'algorithme converger vers un minimum local.

Nishino *et al.* [130], par exemple, travaillent avec un scanner associant à chaque point 3D l'intensité du spot laser perçue lors de sa mesure. Ils proposent alors de comparer l'intensité lumineuse des points pour trouver la correspondance la plus pertinente parmi plusieurs candidats potentiels. Ainsi, non pas un, mais k plus proches voisins sont récupérés, et celui

dont l'intensité lumineuse est la plus semblable de celle du point considéré est conservé.

De leur côté, Godin *et al.* [57] suggèrent d'utiliser prioritairement les points dont le vecteur d'attributs (couleurs, invariants, etc.) se retrouve simultanément dans les deux jeux de données. Ces points ont en effet plus de chance de se trouver dans la zone de recouvrement commune aux deux acquisitions, et donc d'avoir un homologue.

Sappa *et al.* [163] proposent de combiner l'algorithme ICP avec une méthode d'extraction de caractéristiques géométriques. Leur minimisation itérative ne travaille donc qu'avec une représentation simplifiée de l'objet, à savoir les seuls points situés sur les arêtes franches. L'avantage de cette combinaison est double : ne travailler que sur les points caractéristiques réduit les chances de tomber dans un minimum local, mais réduit également les temps de calcul. Cependant, cette méthode nécessite que la taille de la zone de recouvrement soit très importante.

Un autre moyen d'utiliser des attributs pour améliorer le recalage est de les incorporer directement dans la métrique de détermination du plus proche voisin. Pour ce faire, Yang *et al.* [206] utilisent un facteur dépendant des valeurs de courbure calculées en p_i et q_i pour pondérer la distance Euclidienne qui les sépare. Dans le cas plus général de vecteurs d'attributs quelconques $a(p_i)$ et $a(q_i)$ définis en ces deux points, Sharp *et al.* [166] proposent la métrique suivante :

$$d(p_i, q_i) = \|p_i - q_i\|^2 + \alpha \|a(p_i) - a(q_i)\|^2$$

où le poids α sert à moduler l'influence du vecteur d'attributs dans le calcul de la distance. Les auteurs montrent, notamment, que cette métrique augmente la probabilité d'atteindre le minimum global dans le processus de minimisation. Cependant, utilisée avec des invariants géométriques calculés sur un voisinage très localisé, et donc très sensibles au bruit, cette métrique peut entraîner une perte de précision lorsque l'algorithme commence à converger. La valeur de α est donc modifiée au cours de l'algorithme de manière à ce qu'elle soit forte dans les premières itérations, lorsque le recalage est encore assez grossier, et très faible lorsque l'algorithme commence à converger et qu'une précision maximale est requise.

Une métrique très similaire est définie par Johnson *et al.* [86] mais, cette fois, dans le cas plus particulier d'attributs de couleur représentés dans des espaces trichromatiques quelconques.

Utiliser de telles métriques, tenant compte d'attributs supplémentaires, revient à augmenter la dimensionalité de l'espace de recherche (3D pour les coordonnées géométriques + n D pour les attributs). Or, l'étude menée par Nene *et al.* [126] montrent que les méthodes traditionnelles de recherche de plus proches voisins, telles que le k D-arbres, s'effondrent lorsque le nombre de dimensions devient trop important. On peut ainsi noter que ce constat a abouti au développement d'algorithmes de recherche de plus proches voisins spécifiquement dédiés aux espaces de grande dimensionalité [3, 126].

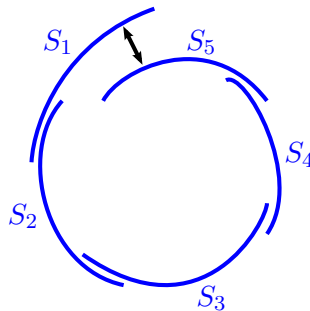


Figure 3.7 — Mauvais alignement induit au niveau global par une accumulation de l'erreur lors de recalages par paires successifs.

4 Recalage global

Dans cette section, nous allons discuter d'algorithmes de recalage développés dans le cas où plus de deux surfaces sont à considérer à la fois. En effet, jusqu'à présent, nous n'avons abordé le problème du recalage que dans le cas de deux jeux de données devant être alignés l'un par rapport à l'autre. Or, une campagne complète de numérisation demande généralement bien plus que deux acquisitions pour couvrir l'intégralité de la surface d'un objet, surtout si la géométrie de ce dernier est complexe (nombreuses concavités, trous, etc.).

Une approche naïve pour replacer un ensemble d'acquisitions dans un même repère est de procéder de manière incrémentale. Une première vue 3D est choisie comme référentiel. Toutes les autres vues sont alors recalées par rapport à celle-ci, en commençant par celles qui lui sont immédiatement adjacentes. On procède ainsi de proche en proche, en prenant comme nouveau référentiel les vues fraîchement recalées, et ainsi de suite jusqu'à ce que l'ensemble des acquisitions aient été traitées. On appelle cela un *recalage chaîné*.

Même si cette approche produit un résultat, elle présente tout de même l'inconvénient d'accumuler l'erreur de recalage à mesure que le processus incrémental avance. En effet, si le recalage n'est considéré qu'à un niveau local, rien n'empêche que des erreurs d'alignement surviennent à un niveau plus global. La figure 3.7 illustre l'exemple d'un recalage chaîné, où chaque acquisition est recalée par rapport à la précédente. Même si chacun des recalages locaux, considéré indépendamment des autres, peut sembler parfait, une accumulation progressive de l'erreur provoque tout de même un décalage entre la première (S_1 sur la figure) et la dernière (S_5 sur la figure) acquisition de la chaîne.

Il est important de remarquer que le problème illustré dans l'exemple précédent se produit de manière systématique : toutes les méthodes de recalage chaîné produisent des décalages, aussi mineurs soient-ils. C'est pour palier à ce genre de problèmes que les techniques que nous allons présenter, dites de *recalage global*, ont été développées.

■ 4.1 Arbre de recouvrement minimal

Le premier type d'approches pour aborder ce problème consiste à considérer tous les recalages possibles en prenant les acquisitions deux à deux, et à ne conserver que ceux engendrant l'erreur la plus faible.

Pour ce faire, un graphe complet est construit, dans lequel chaque nœud représente une surface, et chaque arête la transformation de recalage calculée entre les deux nœuds qu'elle relie. La qualité du recalage entre chaque paire est évaluée, et permet de définir un poids sur l'arête correspondante. Un arbre de recouvrement minimal est finalement calculé sur ce graphe afin d'éliminer les recalages de moindre qualité.

Cette approche est simple à mettre en œuvre, et cela explique certainement pourquoi de nombreux travaux l'utilisent [82, 176, 201, 208]. Il suffit, en effet, de disposer d'un algorithme de recalage par paire, comme ceux présentés dans les sections précédentes, ainsi que d'un moyen de mesurer la qualité de chacun des recalages obtenus. Ce dernier point est cependant crucial, et mérite une attention particulière si l'on souhaite qu'une telle approche fournisse de bons résultats. À ce sujet, Huber *et al.* [82] ont proposé des mesures de consistance très robustes pour évaluer la qualité d'un recalage par paires dans le cadre de tels algorithmes.

Avec une bonne mesure de consistance, les approches par arbres de recouvrement minimal sont capables de reconstruire un objet à partir de plusieurs vues 3D de manière totalement automatique. Elles présentent tout de même un gros défaut : la première étape, consistant à calculer un recalage pour toutes les paires de vues possibles, a une complexité en $\mathcal{O}(n^2)$, où n est le nombre de vues en question, ce qui entraîne une explosion combinatoire lorsque ce nombre est trop important. Huber *et al.* [82] reconnaissent d'ailleurs eux-mêmes que ces approches deviennent inexploitables dès lors que $n > 50$. Si la séquence est disponible, on peut éventuellement s'aider de l'ordre dans lequel les acquisitions ont été capturées pour réduire le coût combinatoire, comme cela est proposé par Pingi *et al.* [139].

■ 4.2 ICP global

La seconde catégorie d'approches pour le recalage global est composée des méthodes travaillant par minimisation itérative [53, 130, 145]. Il s'agit en fait d'algorithmes de type ICP dont l'étape de recherche de plus proches voisins est modifiée : au lieu de ne rechercher les correspondances qu'entre les acquisitions prises deux à deux, celles-ci sont recherchées entre chaque surface et toutes les autres en même temps (figure 3.8). La transformation de recalage calculée minimise alors l'erreur de manière global, en tenant compte de toutes les surfaces adjacentes à la fois.

Puisqu'il s'agit d'algorithmes de minimisation, ces méthodes nécessitent, de disposer d'un premier recalage pour pouvoir fonctionner. Elles ont néanmoins l'avantage de réellement chercher à minimiser une fonction d'erreur globale, contrairement aux méthodes par arbre de recouvrement minimal, qui se contentent de choisir la « moins pire » des solutions parmi

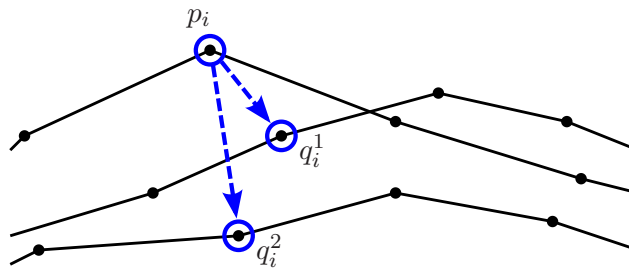


Figure 3.8 – ICP global. Pour chaque point d'une acquisition, les correspondances sont recherchées dans toutes les autres acquisitions.

l'ensemble des transformations préalablement fournies par le recalage par paires.

5 Algorithmes génétiques de recalage

Vue la complexité du problème de recalage dans le cas général, des algorithmes de type évolutionnaire ont naturellement vu le jour pour tenter de le résoudre de manière automatique, sans nécessiter la moindre connaissance *a priori*. Il faut cependant savoir que les approches évolutionnaires restent très marginales par rapport à l'ensemble de la littérature concernant le recalage géométrique. Les quelques travaux que nous allons présenter, même s'ils sont rares, montrent tout de même que ces approches suscitent un certain intérêt dans la communauté, peut être du fait de la simplicité de leur mise en œuvre.

Chaque algorithme génétique se différencie des autres au travers de l'information représentée par les chromosomes qu'il manipule. Ainsi, Lomonosov *et al.* [103] utilisent des chromosomes à six gènes représentant les transformations de recalage sous la forme de trois angles d'Euler et trois coordonnées de translation. Ils axent leur méthode non pas sur la précision mais sur l'efficacité. Pour ce faire, les six paramètres sont codés sur des valeurs entières, afin d'éviter les calculs en virgule flottante plus coûteux, et la qualité de chaque chromosome est évaluée en ne testant la transformation de recalage qu'il représente que sur un sous-échantillonnage très grossier des jeux de données initiaux.

Silva *et al.* [167] utilisent également des chromosomes représentant des transformations rigides, mais ils formulent le problème de manière à pouvoir traiter le cas du recalage global.

De leur côté, Brunnstrom *et al.* [18] définissent chaque chromosome comme étant un vecteur de paires de points correspondants. Ils utilisent alors une fonction d'évaluation permettant de tester la validité de ces correspondances uniquement à partir de critères invariants par transformations rigides. Contrairement aux deux méthodes précédentes, cela leur permet d'évaluer la qualité des chromosomes sans avoir à effectuer un recalage complet pour chacun d'eux et à chaque génération.

Cette phase d'évaluation constituant, en effet, l'un des points critiques des méthodes de recalage évolutionnaires, Robertson *et al.* [148] proposent de répartir l'exécution de cette étape spécifique sur plusieurs processeurs à l'aide d'un algorithme génétique distribué.

Le recalage produit par la plupart de ces méthodes n'a pas pour vocation d'être précis : il s'agit avant tout de fournir, de manière totalement automatique, une première bonne estimation. Celle-ci peut toujours être raffinée par la suite à l'aide d'algorithmes de types ICP, comme c'est le cas de Lomonosov *et al.* [103] et Brunnstrom *et al.* [18]. Les résultats avancés par Robertson *et al.* [148] montrent néanmoins une très grande précision, permettant largement de se passer de ce genre de post-traitements.

6 Bilan

Concernant le recalage, nous pouvons rappeler nos contraintes initiales :

- *l'automatisme* : aucune intervention manuelle, même visant à fournir un premier alignement grossier, ne doit être requise. Ces traitements deviennent en effet rapidement fastidieux lorsque le nombre d'acquisitions devient important.
- *le contrôle interactif* : la qualité des acquisitions doit pouvoir être contrôlée au moment même de la mesure. Cela nécessite des traitements rapides afin de permettre une visualisation immédiate de résultats intermédiaires.
- *préservation des modèles* : l'utilisation de mires physiques est inacceptable dans notre cas, sachant que les mesures sont destinées à être accomplies sur site, à partir d'œuvres d'art potentiellement très fragiles.

Parmi toutes les méthodes qui ont été présentées, rares sont celles qui répondent à l'ensemble de ces critères. Les algorithmes de type ICP ne peuvent clairement pas prétendre à l'automatisme, puisqu'ils demandent impérativement de disposer d'un premier alignement. Les algorithmes basés sur la recherche automatique de correspondances fonctionnent tous sur le principe d'une recherche exhaustive. Le domaine de recherche est bien entendu réduit grâce aux caractéristiques ou aux invariants extraits, mais la détermination de la transformation est néanmoins souvent coûteuse. Il en va bien sûr de même pour les algorithmes génétiques, qui sont eux-mêmes des méthodes exhaustives affirmées. Pour ce qui est de l'utilisation d'attributs supplémentaires, il ne s'agit pas d'une solution en soi, mais plutôt d'un moyen de guider la recherche de manière plus efficace et plus rapide. Cela peut amener à plus d'interactivité, mais ne suffit pas pour autant à rendre une méthode totalement automatique, bien que ces attributs permettent de lever certaines ambiguïtés pour les cas pathologiques d'objets présentant des symétries dans leur forme.

Il faut cependant constater que les algorithmes de type ICP fournissent des résultats très précis dans la mesure où il s'agit d'algorithmes de minimisation. Pour peu que l'on soit capable de fournir l'alignement initial requis, il peut être avantageux d'utiliser de tels algorithmes pour raffiner les transformations obtenues. Enfin, une étape de recalage globale

semble incontournable. Elle permet notamment d'éviter des artefacts visuels sur la reconstruction finale de l'objet, imputables aux décalages importants entre deux acquisitions incorrectement alignées. Les approches par ICP global sont préférables à celles par arbre de recouvrement minimal dans la mesure où, une fois de plus, il s'agit de minimiser une erreur, et non de sélectionner la possibilité la moins défavorable parmi un ensemble restreint de transformations.

- Intégration et filtrage -

Fusion de surfaces 3D et élimination du bruit de numérisation

Dans ce chapitre, nous allons discuter des différents traitements géométriques nécessaires à la génération de la copie numérique finale. Ces traitements considèrent que le recalage géométrique a été préalablement effectué, et travaillent donc à partir d'une description géométrique de l'objet consistant en un ensemble de surfaces disjointes, alignées les unes par rapport aux autres.

Le premier de ces traitements consiste justement à reconstruire une surface unique (et potentiellement fermée, selon l'algorithme choisi) en fusionnant ces différentes acquisitions partielles. En effet, si le recalage a pu les aligner, ces dernières n'en demeurent pas moins déconnectées les unes des autres. Cette étape de fusion, appelée *intégration*, est donc primordiale, d'autant plus qu'elle ne se contente pas uniquement de construire un maillage unifié, mais elle propose également de résoudre certains problèmes d'alignement non résolus par le processus de recalage. Ces erreurs d'alignement, si elles ne sont pas correctement prises en compte, risquent en effet de générer des artefacts visuels qui nuisent au bon déroulement des algorithmes de visualisation que nous présenterons dans le chapitre 5. De même, certains projets de numérisation n'ont pas comme unique but la visualisation : dans le projet concernant la statue de David [19], les auteurs présentent des simulations physiques d'écoulements d'eau sur la copie numérique pour déterminer les zones les plus exposées aux intempéries, et donc les plus sujettes à une dégradation liée à une exposition en milieu extérieur. Pour de telles simulations, l'obtention d'un maillage unique et fermé est indispensable. Toutes ces raisons justifient donc l'usage des processus d'intégration que nous allons présenter dans la section 1.

Le second traitement géométrique dont il va être question dans ce chapitre est, quant à lui, d'ordre plus esthétique : il s'agit de filtrer la surface de l'objet acquis afin d'en éliminer le bruit de numérisation. Ce processus de filtrage peut donc permettre d'aboutir, au moins au niveau visuel, à une plus grande fidélité de la copie numérique par rapport à l'objet réel dont elle est issue. Nous débattons des approches existantes, dont la plupart proviennent initialement du domaine du traitement du signal, dans la section 2. Nous verrons, notamment, que l'une des principales problématiques à considérer est de savoir différencier le bruit de numérisation des détails géométriques de l'objet, sans quoi ceux-ci risquent de disparaître

en même temps que le bruit au moment du filtrage.

On peut éventuellement noter que si l'ordre d'application de ces deux traitements semble, d'un premier abord, inversible, le filtrage est d'ordinaire effectué après l'intégration, histoire d'éviter de filtrer chaque morceau de surface indépendamment avant que ceux-ci n'aient été corrigés (en termes d'alignement) et fusionnés par le processus d'intégration.

1 Intégration

La reconstruction d'une surface à partir d'un échantillonnage de points n'est pas un problème simple dans le cas général : un ensemble de points ne possède en soi aucune information de topologie ou d'orientation. De nombreux travaux ont cependant tenté de le traiter [11, 76], et un bon survol de ces méthodes généralistes peut être trouvé dans l'article de Mencl *et al.* [117].

Si certaines d'entre elles pourraient bien être utilisées dans notre cas, elles restent néanmoins quelque peu hors propos : les données géométriques mesurées par un dispositif tel que le nôtre sont renvoyées sous la forme d'images télémétriques, et possèdent donc une structure sous-jacente qui permet de reconstruire certaines informations sur l'échantillonnage, telle que la connexité entre les points (implicitement fournie par la paramétrisation 2D de l'image) et leurs orientations (lorsque la connexité est connue, les normales peuvent aisément être reconstruites). Le problème de la reconstruction ne se pose donc plus ici dans le cas général, puisqu'une grande quantité d'information *a priori* est potentiellement disponible. Nous ne nous intéresserons donc, dans cet état de l'art, qu'aux méthodes spécifiquement conçues pour les données de numérisation.

Ces méthodes, en plus de fournir une surface unique, ont pour objectif d'éliminer la redondance dans les données, redondance dont la présence est à peu près systématique du fait de son absolue nécessité lors la phase précédente de recalage. Mais cette élimination doit se faire intelligemment : que ce soit à cause du bruit, d'erreurs de calibrage, ou d'un recalage trop grossier, les différentes acquisitions impliquées dans une zone de recouvrement ne sont jamais parfaitement alignées. Une suppression pure et simple des données redondantes risque donc de générer des artefacts dans la géométrie du modèle finale. Ces algorithmes doivent donc tenir compte des différentes sources de données afin de générer une *surface consensuelle* visant à corriger ces défauts, comme cela est illustré sur la figure 4.1.

À l'époque des premiers travaux de numérisation en laboratoire, les chercheurs utilisaient presque systématiquement des tables rotatives calibrées pour résoudre les problèmes de recalage. De ce mode opératoire a émergé l'un des premiers algorithmes d'intégration [27], consistant à reparamétriser l'ensemble du modèle selon une fonction cylindrique dont l'axe coïncide avec celui de la table tournante et qui, à tout couple de coordonnées (θ, h) , associe la distance de la surface. Dans les zones de recouvrement, cette distance est déterminée par

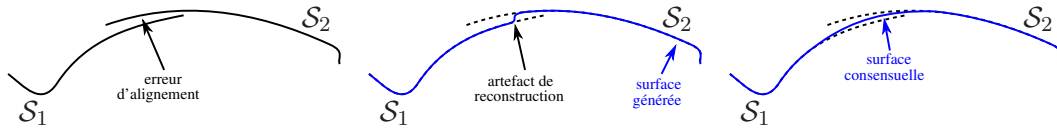


Figure 4.1 – **Gauche** : les deux acquisitions S_1 et S_2 à fusionner. L'alignement de ces surfaces dans la zone de recouvrement n'est pas parfait. **Milieu** : résultat d'une intégration avec suppression simple de la redondance. Des artefacts géométriques peuvent apparaître. **Droite** : intégration par calcul d'une surface consensuelle.

une moyenne pondérée sur les différentes surfaces rencontrées, garantissant des transitions douces et sans artefact. Une surface triangulée est finalement générée dans ce nouvel espace de coordonnées. Cette méthode historique a rapidement présenté ses limites, restreignant la classe des objets restructurables à ceux topologiquement équivalents à une sphère et ne présentant aucune auto-occlusion.

D'autres algorithmes, cette fois utilisables dans le cas général, ont donc été développés. Ceux-ci se distinguent principalement selon deux grandes catégories :

- les algorithmes *basés maillages*, présentés dans la section 1.1, qui considèrent chaque acquisition comme une surface partielle, et qui tentent alors de les relier entre elles après avoir supprimé les zones de recouvrement,
- les algorithmes *volumiques*, qui expriment la géométrie dans un espace de représentation alternatif, à savoir une surface implicite définie sur une grille 3D, dans lequel opérations de fusion et de calcul de la surface consensuelle peuvent être effectuées beaucoup plus simplement. Un opérateur inverse est finalement appliqué à cet espace de représentation pour revenir à une surface triangulée. Nous présenterons ces méthodes dans la section 1.2.

Comme nous allons le voir, les approches basées maillages travaillent directement à partir des points de mesure initiaux, contrairement aux approches volumiques, qui perdent totalement cette information du fait du passage dans un autre espace de représentation. Nous discuterons des avantages et des inconvénients de ces deux façons d'approcher le problème dans le bilan de ce chapitre, en section 3.

■ 1.1 Intégration basée maillages

Les algorithmes de cette catégorie travaillent généralement de manière incrémentale, en ne considérant que deux surfaces à la fois : une première, généralement appelée «modèle», qui est le résultat de l'intégration des acquisitions précédentes, et une seconde correspondant à la nouvelle vue 3D à intégrer.

La fusion de ces deux surfaces se déroule alors en quatre étapes :

1. **Détection du recouvrement** : l'ensemble des triangles se trouvant dans la zone commune aux deux surfaces est déterminé.
2. **Calcul d'une surface consensuelle** : cet ensemble de triangles redondants est exploité de manière à parfaire l'alignement entre les deux surfaces, évitant ainsi les artefacts de reconstruction au niveau des jonctions.

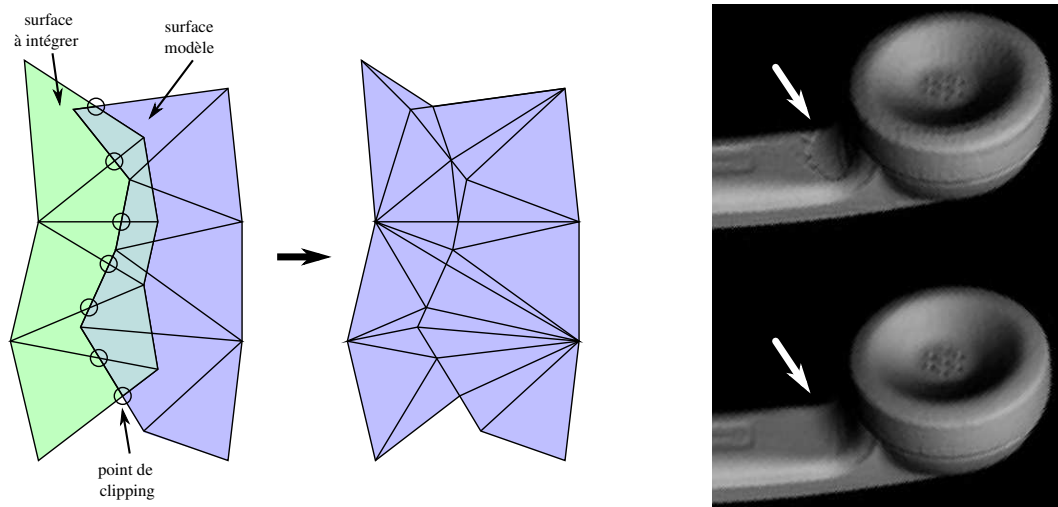


Figure 4.2 – Gauche : processus d'intégration de Turk *et al.* [189]. Des triangles sont supprimés jusqu'à ce que les deux surfaces n'aient plus qu'une bande d'une largeur d'un triangle en commun. Cette bande est alors fusionnée par *clipping*. Droite : exemple d'artefacts de reconstruction corrigés par le calcul de la surface consensuelle proposé par cette même méthode.

3. **Élimination du recouvrement** : les triangles redondants sont éliminés de manière à ce que la surface à intégrer et le modèle soient strictement distincts.
4. **Liaison topologique** : la surface à intégrer est finalement reliée au modèle par une étape de *couture topologique*, résultant en une unique surface.

Pour Turk *et al.* [189], un triangle est considéré comme redondant si les distances entre ses trois sommets et leurs plus proches voisins dans l'autre surface sont inférieures à un seuil donné¹. Ils proposent alors de supprimer ces triangles en érodant simultanément les deux surfaces, jusqu'à ce que la zone commune ne soit plus réduite qu'à une bande d'une largeur d'un triangle. Cette érosion est guidée par un indice de confiance qui tient compte du fait que les surfaces acquises sous un angle rasant et les points situés à proximité des sauts de profondeur dans l'image télémétrique sont sources d'incertitude. Ainsi, les triangles les moins fiables sont supprimés en priorité. La bande commune restante est finalement fusionnée par *clipping*, en découpant toutes les arêtes s'intersectant, et en générant les triangles appropriés. Ce processus est illustré par la figure 4.2-gauche. Le calcul d'une surface consensuelle est alors effectué *a posteriori*, en projetant chaque point du maillage final le long de sa normale sur les surfaces des acquisitions initiales. La nouvelle position est obtenue grâce à une moyenne pondérée des intersections trouvées, les poids impliqués correspondant alors aux indices de confiance définis précédemment. La figure 4.2-droite montre un exemple de résultats obtenus par cette méthode. On peut d'ailleurs clairement y voir le genre d'artefacts qui surviennent lorsqu'aucune correction d'alignement n'est appliquée.

Sun *et al.* [178] proposent de détecter le recouvrement en reprojétant les triangles du modèle

¹Cette méthode de détection est d'ailleurs reprise par Zhou *et al.* [211], mais en considérant les centroïdes des triangles plutôt que leurs sommets.

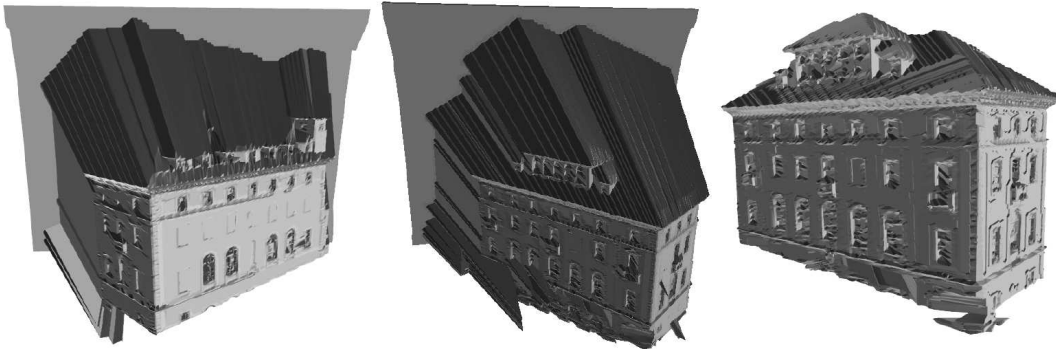


Figure 4.3 — Intégration proposée par Reed *et al.* [147]. Chaque image télémétrique est extrudée le long des lignes d’observation du capteur afin de construire un modèle partiel (*gauche et milieu*). Tous les modèles obtenus sont alors fusionnés à l’aide d’opération de type CSG (*droite*).

dans le domaine de l’image télémétrique à intégrer, et en effectuant les tests dans cet espace 2D. Les triangles de l’image dont au moins un sommet se trouve dans le cercle circonscrit du triangle requête sont alors considérés comme redondants. Notons \mathcal{T} l’ensemble de ces triangles. Comme précédemment, la suppression est basée sur un indice de confiance : la fiabilité du triangle requête est comparée à la fiabilité moyenne de \mathcal{T} et, selon le cas, c’est soit le triangle requête qui est éliminé, soit tous les triangles de \mathcal{T} à la fois. Cette approche est plutôt agressive en ce sens que la région détectée comme redondante risque de s’étendre bien au delà de la zone de recouvrement réelle, du fait que le cercle circonscrit de triangles très étirés peut parfois être très grand.

Pour palier à ce genre de problème, des méthodes plus précises de détection du recouvrement ont été proposées [140, 162], donnant une épaisseur aux surfaces pour tenir compte des erreurs d’alignement en construisant, par extrusion, un prisme autour de chaque triangle. La détection du recouvrement se fait alors par des tests d’inclusion d’un point dans un prisme, ou par de véritables tests d’intersections triangle/polyèdre, ce qui est malheureusement plus coûteux. Dans le même ordre d’idées, Reed *et al.* [147] proposent une approche CSG (*Constructive Solid Geometry*) qui opère sur les volumes construits en extrudant chaque acquisition le long des lignes d’observation du capteur (voir figure 4.3). Même si Pito *et al.* [140] proposent de détecter des erreurs de mesure grâce à des critères de visibilité, aucune de ces trois méthodes ne s’attache réellement au calcul de la surface consensuelle.

À l’inverse, Rutishauser *et al.* [159] considèrent ce problème avec intérêt, en proposant une méthode robuste qui utilise une matrice de covariance déterminée sur la base d’un bruit Gaussien pour définir l’incertitude en chaque point. Ils appliquent alors les règles du filtrage de Kalman [194] sur ces matrices pour corriger les données redondantes. Concernant la fusion, les auteurs adoptent une approche par *clipping* similaire à celle de Turk *et al.* [189].

Soucy *et al.* [173] proposent, quant à eux, une approche totalement différente des méthodes précédentes : ils considèrent chaque acquisition comme un ensemble de données, et représentent les relations de recouvrement à l’aide d’un diagramme de Venn. Chaque cellule de

ce diagramme représente donc exclusivement la portion commune à k images télémétriques (k pouvant être égal à 1 pour les régions n'étant visibles que d'une seule acquisition). Pour chaque cellule, une droite de projection est déterminée par moyennage des directions sous lesquelles les acquisitions qu'elle contient ont été capturées. L'ensemble des données est alors projeté le long de cette droite sur une grille 2D, en moyennant les informations redondantes lorsque plusieurs triangles se projettent sur un même pixel de cette grille 2D. Cette algorithmique revient, en quelque sorte, à utiliser le diagramme de Venn pour générer de nouvelles images télémétriques, cette fois disjointes les unes des autres, et définissant chacune une surface consensuelle. Ces surfaces sont finalement cousues entre elles à l'aide d'un algorithme de Delaunay contraint, à l'instar de Sappa *et al.* [162].

Enfin, on notera les travaux de Li *et al.* [102] qui proposent d'intégrer des nuages de points plutôt que des surfaces triangulées, prétextant que les opérations impliquées pour ce faire sont plus simples. Les auteurs proposent donc une chaîne complète de traitements, avec élimination de la redondance, filtrage de l'objet et simplification du nuage de points final, qui exploite tout de même la structure sous-jacente des images télémétriques. Malheureusement, pour reconstruire, *in fine*, la surface de l'objet, des algorithmes de reconstruction à partir de nuages de points déstructurés doivent tout de même être utilisés, avec toutes les difficultés que cela implique.

■ 1.2 Intégration volumique

L'autre type d'approches largement répandu dans la communauté pour fusionner les acquisitions provenant de différents points de vue se base sur l'usage de surfaces implicites pour distinguer quelles sont les régions de l'espace qui sont incluses dans l'objet. Cette représentation implicite est obtenue à l'aide d'une grille 3D dont chaque voxel est labellisé en fonction de sa position par rapport aux surfaces à intégrer. Une fois que tous les voxels ont été classifiés, un algorithme de détournement [104] est utilisé pour extraire une surface unique correspondant à l'interface entre les régions intérieures et extérieures, comme l'illustre la figure 4.4-gauche.

La classification des voxels est un point essentiel de l'algorithme. Pulli *et al.* [146] partent du constat que l'espace situé entre le scanner et la surface qu'il mesure est nécessairement libre. Ainsi, les voxels sont projetés sur chaque image télémétrique, et seuls ceux qui se trouvent derrière toutes les surfaces à la fois par rapport à leurs points de vue d'acquisition sont marqués comme faisant partie de l'intérieur de l'objet (figure 4.4-droite).

Cette classification binaire intérieur/extérieur présente l'inconvénient de ne pas permettre une reconstruction très précise. Il suffit de voir les illustrations de la figure 4.5 pour s'en convaincre. De nombreux travaux préfèrent utiliser un champ de distances signées : il s'agit d'une fonction de $\mathbb{R}^3 \rightarrow \mathbb{R}$ qui associe à chaque point p de l'espace sa plus courte distance $d_{\min}(p, \mathcal{S})$ à la surface \mathcal{S} . Cette distance peut être soit négative, soit positive, selon que p se situe devant ou derrière la surface, respectivement. Celle-ci peut alors être reconstruite par extraction de l'iso-contour de valeur nulle. De nombreux travaux se sont attachés aux calculs de champs de distances. Le papier de Frisken *et al.* [51] en est un exemple, et l'état

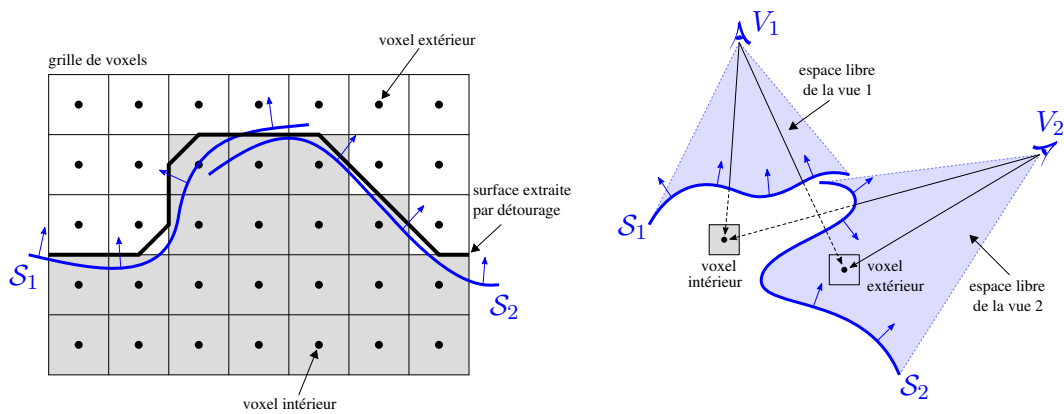


Figure 4.4 — *Gauche* : Intégration à l'aide d'une surface implicite. Chaque voxel d'une grille 3D est classifié selon son appartenance à l'intérieur de l'objet. Un algorithme de détourage permet alors d'extraire une surface unique. *Droite* : classification des voxels selon Pulli *et al.* [146], par test de présence dans l'espace libre situé entre chaque mesure et son point de vue d'acquisition.

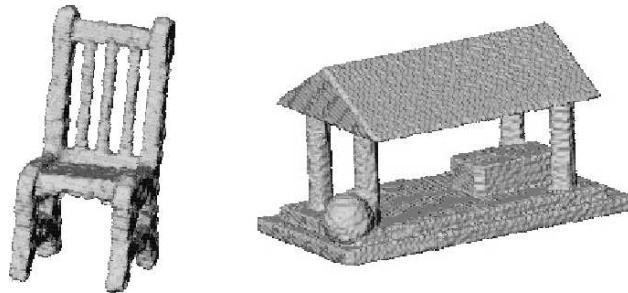


Figure 4.5 — Modèles obtenus à l'aide d'une classification binaire intérieur/extérieur des voxels d'une grille 3D [146]. Un tel modèle de représentation ne permet pas de reconstruire des surfaces lisses.

de l'art qu'il propose couvre un large spectre dans ce domaine. Appliqué à l'intégration, le problème consiste alors à construire une telle structure à partir de N morceaux de surface disjoints et potentiellement bruités (ou mal alignés).

Pour cela, Curless *et al.* [34] proposent de calculer la distance en chaque voxel v_i comme une combinaison linéaire des distances minimales à chacune des acquisitions \mathcal{S}_j :

$$d_{min}(v_i, \{\mathcal{S}_j\}_{1 \leq j \leq N}) = \frac{\sum_{j=1}^N w_{min}(v_i, \mathcal{S}_j) d_{min}(v_i, \mathcal{S}_j)}{\sum_{j=1}^N w_{min}(v_i, \mathcal{S}_j)} \quad (4.1)$$

où le poids $w_{min}(v_i, \mathcal{S}_j)$ correspond à l'indice de confiance du point de la surface \mathcal{S}_j le plus proche de v_i . L'indice en question est le même que celui défini dans la méthode de Turk *et al.* [189] présentée précédemment. Cette combinaison linéaire a en fait pour effet de déplacer l'iso-contour de valeur nulle vers la surface consensuelle, résolvant ainsi les problèmes d'alignement.

Wheeler *et al.* [197] adoptent une approche similaire :

- Pour un voxel donné, l'ensemble $\{p_j\}_{1 \leq j \leq N}$ des points les plus proches de son centre dans chaque acquisition est d'abord construit.
- La position de chaque p_j est ensuite recalculée de manière consensuelle en cherchant ses plus proches voisins dans toutes les autres acquisitions, et en moyennant ceux qui sont compatibles entre eux en termes de position et d'orientation (selon un seuil fixé).
- La distance au plus proche des p_j corrigés est alors utilisée pour définir la distance signée du voxel.

Les auteurs vendent leur méthode comme étant plus robuste que la précédente, du fait qu'ils appliquent un filtrage (tests de compatibilité) avant de combiner les acquisitions.

On peut remarquer l'approche de Neugebauer *et al.* [127, 128], qui se distingue des autres méthodes d'intégration volumique par le fait que le calcul de la surface consensuelle est dé-couplé du problème d'intégration en lui-même : ce calcul est effectué *avant* que le champ de distances ne soit effectivement construit, à l'aide d'une méthode basée maillages. Le principe est de corriger chaque acquisition en déplaçant ses points le long des lignes d'observation du capteur. La nouvelle position d'un point correspond alors à la moyenne des intersections du rayon d'observation qui le traverse avec les autres surfaces. Cette correction le long des lignes de visée se base sur un postulat couramment avancé dans le domaine de la numérisation, selon lequel l'incertitude est plus forte le long de ces directions.

Vue les structures de données volumiques nécessaires à ces méthodes d'intégration, on comprend vite que leur principale limitation est la consommation mémoire, d'autant plus que la résolution de la grille doit être très fine si l'on souhaite obtenir une reconstruction précise.

Pour réduire ce coût, Curless *et al.* [34] ne calculent le champ de distances que pour une fine bande de voxels autour de la surface. La grille 3D est alors considérée comme un ensemble de tranches, chacune étant compressée par un algorithme RLE et décompressée à la demande. On peut également noter quelques approches par subdivision non régulière de l'espace, qui utilisent la structure hiérarchique d'un arbre octal pour affiner le découpage uniquement à proximité de la surface [146, 161].

L'autre inconvénient majeur, bien entendu, est la nécessité d'un algorithme de détournement pour revenir de la représentation implicite à une représentation surfacique du modèle.

2 Filtrage

La numérisation 3D, comme n'importe quel procédé de mesure, introduit du bruit dans les données qu'elle acquiert. Ce bruit engendre un déplacement aléatoire des points de mesure par rapport à leurs positions théoriques, ce qui se traduit visuellement par un effet de granulosité comme celui que l'on peut voir sur l'image de gauche de la figure 4.6. L'effet du filtrage est donc principalement d'ordre esthétique, visant à lisser une surface pour la rendre plus plaisante à l'œil, et surtout plus fidèle à l'objet réel dont elle est issue (au moins

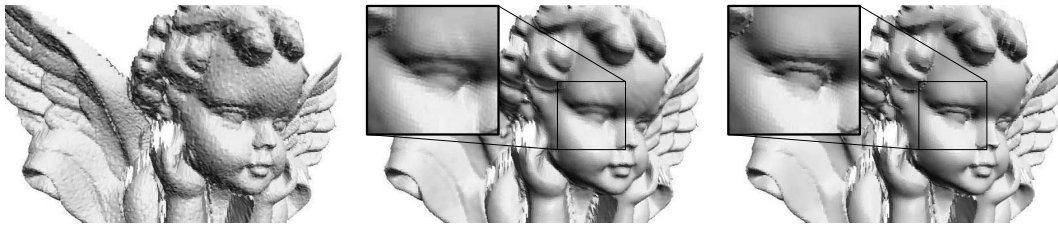


Figure 4.6 – **Gauche** : aspect granuleux de la surface induit par le bruit de numérisation. **Milieu** : application d'un filtre simple, type moyenneur. En plus d'éliminer le bruit, certaines caractéristiques géométriques sont altérées. **Droite** : application d'un filtre adaptatif. Le bruit est éliminé, mais les caractéristiques géométriques sont préservées. (Images tirées de [112].)

au niveau visuel).

Comme nous allons le voir, la plupart des algorithmes développés pour le filtrage de surfaces 3D correspondent à l'adaptation de filtres ayant initialement vu le jour dans la communauté du traitement du signal. La principale problématique inhérente au processus de filtrage est d'éviter d'éliminer les fins détails de la géométrie en même temps que le bruit. En effet, la géométrie d'un objet peut présenter trois cas de figure qui nécessitent des traitements spécifiques :

- les *zones planes*, ou basses fréquences, qui représentent les zones de faible courbure de l'objet,
- les *caractéristiques*, ou hautes fréquences, qui représentent les zones à forte courbure,
- les *discontinuités*, qui correspondent aux arêtes franches de l'objet, au niveau desquelles l'orientation de la surface change brutalement.

Pour les zones planes, des filtres simples peuvent être utilisés, comme des équivalents du filtre moyenneur en traitement d'images, par exemple. En revanche, pour les deux autres cas, l'application de tels filtres risque d'altérer la géométrie de façon significative.

On peut ainsi distinguer deux grandes catégories de filtres :

- les *filtres simples*, qui correspondent à des filtres moyenneurs (passe-bas) appliqués aux maillages 3D. Ces filtres ne font aucune distinction entre les trois cas de figure présentés plus haut : le filtrage est appliqué de manière égale sur l'ensemble de la surface, sans chercher à en préserver les détails éventuels, comme on peut le voir sur la figure 4.6-milieu.
- les *filtres adaptatifs*, qui sont des filtres plus évolués développés dans le but de préserver les fins éléments de la géométrie, comme dans l'exemple de la figure 4.6-droite. Leurs paramètres sont modifiés au cours du filtrage en fonction de la variation locale de la géométrie. On peut ainsi distinguer deux sous-catégories :
 - les filtres adaptatifs qui modulent la puissance de filtrage,
 - les filtres adaptatifs qui modulent la forme du masque du filtre.

Nous verrons que cette première sous-catégorie est plus adaptée à la préservation des caractéristiques, alors que la seconde a été prioritairement développée dans le but de préserver les discontinuités.

Nous allons donc commencer par présenter les filtres simples, puisqu'ils sont historique-

ment les premiers à être apparus. Le premier dont nous allons discuter est le filtre Laplacien, décrit dans la section 2.1, qui est le premier des opérateurs de filtrage à avoir été développé pour les maillages, et nous verrons notamment comment l’outil puissant qu’est la transformée de Fourier a été définie sur les maillages à partir de ce filtre. Nous verrons ensuite d’autres filtres simples, en section 2.2, clairement liés à des opérateurs de traitement d’images existants avec lesquels nous ferons le lien. Nous discuterons ensuite, dans les sections 2.3 et 2.4 respectivement, des filtres adaptatifs à puissance et à masque variable. Nous illustrerons les différences constatables entre ces deux approches. Enfin, nous parlerons brièvement d’autres types d’approches pouvant être utilisées pour le filtrage, bien que ce ne soit pas forcément l’objectif premier de ces travaux connexes (section 2.5), avant de conclure par le bilan général de ce chapitre.

■ 2.1 Premier filtre simple : le filtre Laplacien

Le filtre Laplacien [182, 190] est un filtre moyenneur qui utilise l’opérateur Laplacien $\vec{\Delta}p_i$ défini en chaque sommet p_i d’un maillage 3D par l’équation 4.2 :

$$\vec{\Delta}p_i = \frac{1}{\text{card}(\mathcal{N}(p_i))} \sum_{p_j \in \mathcal{N}(p_i)} p_j - p_i \quad (4.2)$$

où $\mathcal{N}(p_i)$ est l’ensemble des points du maillage adjacents à p_i . Cet opérateur représente le déplacement qui sépare le point p_i du centre de l’ombrelle définie par l’ensemble de ses triangles voisins, comme le montre la figure 4.7-gauche. Le filtre Laplacien calcule alors les nouvelles coordonnées p'_i du sommet p_i par un déplacement le long de son Laplacien d’une amplitude κ définissant la puissance du filtre :

$$p'_i = p_i + \kappa \vec{\Delta}p_i \quad (4.3)$$

Le lissage s’effectue donc par un aplatissement de l’ensemble des ombrelles qui composent le maillage.

Ce filtre présente cependant un défaut bien connu : il tend à faire rétrécir les objets. Étant donné que l’application d’un filtre se fait généralement de manière itérative, afin de permettre un lissage plus ou moins prononcé du modèle selon l’intensité du bruit à traiter, ce rétrécissement sera, par là même, de plus en plus flagrant. Cet effet indésirable est illustré par la figure 4.7-droite, où l’on constate clairement une perte progressive de volume à mesure que le nombre d’itérations augmente.

Vollmer *et al.* [190] ont proposé de modifier ce filtre de manière à éviter cet effet de rétrécissement. L’idée est alors de filtrer l’objet tout en contraignant les points à rester proches de leurs positions initiales, en combinant au résultat du Laplacien les coordonnées du point à l’itération précédente et à l’itération zéro (coordonnées initiales).

Taubin [182, 183, 184] rappelle que l’origine de ce défaut est que le filtre Laplacien n’est en fait pas un véritable filtre passe-bas. Il propose alors, comme solution, d’aborder le problème du filtrage sous une approche fréquentielle, en généralisant la transformée de Fourier aux

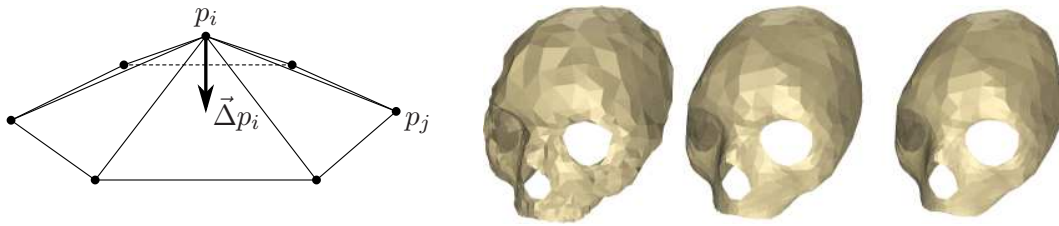


Figure 4.7 – **Gauche** : opérateur Laplacien défini en un sommet p_i d'un maillage triangulaire. **Droite** : trois itérations du filtre Laplacien appliquées à un modèle 3D bruité. On peut noter le rétrécissement progressif du modèle, principal défaut de ce filtre (images tirées de [190]).

maillages triangulaires, de manière à pouvoir définir de tels filtres passe-bas. En représentant un maillage par le vecteur $\mathbf{p} = (p_1, \dots, p_N)$ de ses sommets, l'opérateur Laplacien peut être simplement réécrit sous la forme matricielle suivante :

$$W = (w_{ij})_{1 \leq i, j \leq N}, \quad \text{où } w_{ij} = \begin{cases} \frac{1}{\text{card}(\mathcal{N}(p_i))} & \text{si } p_j \in \mathcal{N}(p_i) \\ 0 & \text{sinon} \end{cases} \quad (4.4)$$

ce qui permet de reformuler l'application du filtre Laplacien par la relation suivante :

$$\mathbf{p}' = (I - \kappa W) \mathbf{p} \quad (4.5)$$

où le vecteur $\mathbf{p}' = (p'_1, \dots, p'_N)$ correspond à l'ensemble des sommets du maillage filtré.

Cette représentation matricielle est alors décomposée en un ensemble de vecteurs propres pouvant être vus comme une décomposition fréquentielle du maillage, les vecteurs propres associés aux plus petites valeurs propres représentant alors les basses fréquences, et *vice versa*. L'auteur utilise alors cette représentation pour appliquer simultanément deux filtres Laplaciens à des niveaux fréquentiels différents : l'un positif ($\kappa > 0$), sur les hautes fréquences, pour lisser la surface, et l'autre négatif ($\kappa < 0$), sur les basses fréquences, pour compenser le phénomène de rétrécissement engendré par l'application du premier filtre.

Cette approche très élégante a permis de montrer que des techniques de traitement du signal autres que la simple application d'opérateurs linéaires de filtrage pouvaient être adaptées dans le but de traiter des surfaces.

■ 2.2 Autres filtres simples

Concernant les filtres simples, deux filtres bien connus en traitement d'images, le filtre moyenneur et le filtre médian, ont été adaptés au traitement de maillages 3D par Yagou *et al.* [203].

La différence, par rapport au filtre Laplacien et ses variantes présentés dans la section précédente, est que ces deux adaptations filtrent les normales des triangles au lieu des positions des sommets. Ainsi, ces deux filtres considèrent, pour chaque triangle t , un voisinage $\mathcal{N}(t)$

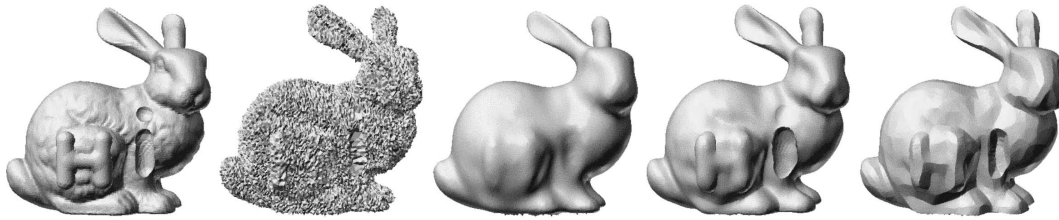


Figure 4.8 – De gauche à droite : 1) maillage original, 2) maillage bruité artificiellement, 3) résultat du filtre Laplacien, 4) résultat du filtre moyennneur, 5) résultat du filtre médian (images tirées de [203]).

composé de tous les triangles ayant un sommet ou une arête en commun avec t , et recalculent la normale \vec{n}_t de t à partir de ce voisinage.

Dans le cas du filtre moyennneur, la nouvelle normale est simplement la moyenne des normales de $\mathcal{N}(t)$. Dans le cas du filtre médian, l'angle formé entre $\vec{n}(t)$ et la normale en chacun des triangles de $\mathcal{N}(t)$ est calculé. La normale de $\mathcal{N}(t)$ associée à la valeur médiane de tous ces angles est alors choisie comme nouvelle normale pour le triangle t .

Bien entendu, filtrer les normales ne modifie pas la géométrie. Pour que le filtrage devienne effectif, les sommets doivent tout de même être déplacés *a posteriori* de manière à ce que les orientations des triangles satisfassent les normales mises à jour. Les auteurs proposent, pour ce faire, une méthode qui équivaut à appliquer une rotation à chaque triangle par rapport à son centre de gravité, l'avantage étant que cette approche empêche l'effet de rétrécissement observé dans le cas du filtre Laplacien de se produire.

Les résultats obtenus par Yagou *et al.* [203] pour ces deux filtres sont, bien évidemment, à mettre en parallèle avec ceux observés pour leurs homologues du traitement du signal :

- le filtre moyennneur est plus adapté aux objets dénués de discontinuités ou de caractéristiques trop prononcées. Dans le cas contraire, les éléments hautes fréquences sont lissés sans être préservés.
- à l'inverse, le filtre médian préserve bien les discontinuités mais a tendance à faire apparaître des aplats dans les zones arrondies mais de faible courbure.

On peut également noter, en regardant la figure 4.8, que le phénomène de «sur-lissage», qui est un autre problème du filtre Laplacien, ne se pose plus dans le cas de ces deux filtres.

Ces résultats mettent en évidence la nécessité des algorithmes adaptatifs dont il va être question à partir de la section suivante. En effet, un objet composé à la fois de zones homogènes de faibles courbures et de zones caractéristiques ou de discontinuités ne pourra pas être correctement traité par l'un ou l'autre de ces filtres simplistes.

■ 2.3 Filtres adaptatifs à puissance variable

Les filtres adaptatifs sont chronologiquement apparus après les filtres simples, ce qui est normal dans la mesure où ils constituent une évolution logique de ces premiers filtres. Les filtres adaptatifs de la première sous-catégorie que nous allons voir modulent la puissance

du filtrage en fonction des variations géométriques de la surface sous-jacente.

Le filtre MMSE (*Minimum Mean Squared Error*) [59, 112], par exemple, se base sur la variance σ_{local}^2 des normales dans un voisinage donné pour quantifier les variations locales de la géométrie. L'idée est alors de comparer cette variance à la variance σ_{bruit}^2 du bruit (censée être connue et fournie par l'utilisateur) afin de régler la puissance du filtre. Le filtrage opère sur les normales, à l'instar de Yagou *et al.* [203], avec le même mécanisme de mise à jour des positions des sommets dont nous avons discuté précédemment.

Le filtre MMSE calcule donc la nouvelle normale \vec{n}'_t d'un triangle t par interpolation linéaire entre sa normale initiale \vec{n}_t et une normale \vec{n}_{moy} résultant de l'application du filtre moyennneur de Yagou *et al.* :

$$\vec{n}' = \begin{cases} \vec{n}_{moy} & \text{si } \sigma_{bruit}^2 > \sigma_{local}^2 \\ \left(1 - \frac{\sigma_{bruit}^2}{\sigma_{local}^2}\right) \vec{n} + \left(\frac{\sigma_{bruit}^2}{\sigma_{local}^2}\right) \vec{n}_{moy} & \text{sinon} \end{cases} \quad (4.6)$$

Ainsi, si la variance locale est plus importante que celle du bruit, on peut dire que les variations géométriques représentent de l'information utile et, dans ce cas, un filtrage de faible puissance est appliqué (l'interpolation penche en faveur de la normale initiale). Dans le cas contraire, la variance est trop faible pour représenter des caractéristiques distinguables du bruit et, dans ce cas, un filtrage beaucoup plus fort est appliqué (le filtre moyennneur prend plus d'importance). La figure 4.6-droite illustre ce que peut donner l'application de ce filtre.

Comparé aux filtres Laplacien, médian et moyennneur, le filtre MMSE donne de très bons résultats pour la préservation des caractéristiques. En ce qui concerne les discontinuités, le filtre échoue parfois à éliminer le bruit, du fait que, dans ces zones, la variance locale devient beaucoup trop importante par rapport à celle du bruit, ce qui implique un filtrage systématiquement trop faible.

Pour permettre de mieux gérer ces discontinuités, le filtre bilatéral, initialement proposé en traitement d'images par Tomasi *et al.* [185], a été adapté aux maillages 3D par Fleishman *et al.* [47]. Ce filtre, travaillant cette fois sur les positions des sommets, peut être vu comme la composition de deux filtres Gaussiens :

- le premier s'appliquant à la simple distance Euclidienne entre un point p et son voisinage, donnant ainsi moins d'importance dans le calcul du filtrage aux points les plus éloignés, comme le ferait un filtre Gaussien standard,
- le second s'appliquant à la distance du voisinage au plan tangent en p . En effet, si l'on considère ce plan comme une approximation locale de la surface, les sommets du voisinage qui se trouvent très éloignés de ce plan représentent alors des discontinuités dans la géométrie. Ce filtre permet donc de leur donner moins d'importance, et de se limiter aux seuls points proches de cette approximation locale de la surface.

Les résultats montrent, en effet, une bonne préservation des discontinuités. On peut noter qu'une autre approche pour l'adaptation du filtre bilatéral aux maillages 3D peut également être trouvée dans le papier de Jones *et al.* [87]. Cette seconde approche semble cependant préserver moins efficacement les discontinuités que celle de Fleishman *et al.* [47]. Les résultats de ces deux variantes peuvent être comparés à l'aide de la figure 4.9.

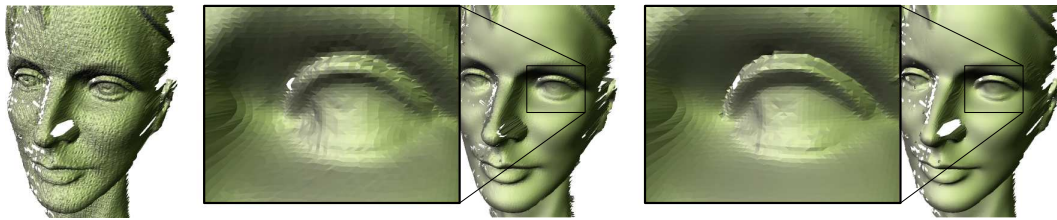


Figure 4.9 – Deux adaptations du filtre bilatéral pour le débruitage de maillages 3D (images tirées de [47]). **Gauche** : maillage avant filtrage. **Milieu** : version de Jones *et al.* [87]. **Droite** : version de Fleishman *et al.* [47].

Si le filtre bilatéral gère mieux les discontinuités que le filtre MMSE, c'est parce qu'il tend à exclure du calcul de filtrage les éléments du voisinage qui divergent trop de l'approximation locale de la surface. Cette idée est en fait très proche des filtres dont il va être question dans la section suivante, c'est-à-dire ceux qui modifient la forme de leur masque en fonction de la géométrie sous-jacente.

■ 2.4 Filtres adaptatifs à masque variable

Pour garantir une bonne préservation des discontinuités, certains filtres modifient la forme de leur masque : plutôt que de travailler sur le voisinage complet d'un sommet ou d'un triangle à filtrer, ce voisinage est tronqué de manière à éliminer ce qui se trouve au-delà d'une discontinuité, et ainsi permettre de mieux la préserver. Ces filtres constituent notre seconde sous-catégorie.

Le filtre SUSAN (*Smallest Univalued Segment Assimilating Nucleus*), par exemple, est bien connu en traitement d'images pour exploiter ce principe. L'idée est de centrer un masque circulaire sur chaque pixel, et de ne considérer que les pixels appartenant à ce masque dont l'intensité est proche de celle du pixel central. Un filtre classique peut finalement être appliqué sur les seuls pixels retenus. Une adaptation de ce filtre pour les maillages 3D a été proposé par Mao *et al.* [110] en utilisant les normales des triangles en lieu et place des intensités lumineuses, avec l'avantage d'éliminer du masque les triangles dont l'orientation diverge trop de celle du triangle filtré, et ainsi de bien préserver les arêtes franches.

Un autre exemple est le filtre trilatéral, proposé par Choudhury *et al.* [28], qui étend le filtre bilatéral précédent de manière à préalablement supprimer du masque les triangles dont la normale diffère de trop de celle du triangle considéré.

Il est important de noter que les deux sous-catégories de filtres adaptatifs ne sont pas mutuellement exclusives mais, au contraire, sont tout à fait complémentaires : une fois que la forme du masque a été adaptée de manière à ignorer ce qui se trouve au-delà d'une discontinuité, la puissance du filtre appliqué aux éléments restants peut toujours être modulée afin de préserver les caractéristiques géométriques. C'est le cas de Ohtake *et al.* [135], qui définissent un filtre Gaussien dont le masque est tronqué au niveau des arêtes franches, mais avec la propriété supplémentaire d'adapter sa taille (et donc sa puissance) en fonction de la

courbure locale de la géométrie, rapetissant dans les régions de forte courbure et s'élargissant dans les régions planes. Ainsi, ce filtre agit aussi bien sur les zones de fortes courbures que sur les discontinuités de la surface, en traitant de manière distincte chacun de ces deux cas de figure.

■ 2.5 Travaux connexes

D'autres travaux connexes à la problématique du filtrage peuvent être trouvés dans le domaine de la reconstruction de modèles CAO à partir de nuages de points, où il est particulièrement intéressant de pouvoir obtenir de belles surfaces lisses avec des arêtes franches préservées. On peut ainsi citer Fleishman *et al.* [46], qui reconstruisent un modèle continu par morceaux en estimant des surfaces paramétriques grâce à des méthodes statistiques permettant de tenir compte du bruit.

D'autres modes de représentation de haut niveau peuvent être utilisés, comme celui de Oh-take *et al.* [134], qui proposent de construire, à partir d'un nuage de points, un ensemble de petits éléments de surface, chacun représenté par une quadrique. L'une des applications qui illustrent l'intérêt de cette représentation est justement le filtrage d'objets 3D.

On peut aussi noter que certains travaux [68, 179, 211] définissent le filtrage comme un problème d'optimisation, en définissant une fonction de coût visant à minimiser la courbure moyenne pour la globalité de l'objet.

Bien entendu, dans le domaine du traitement géométrique, de nombreux autres travaux peuvent se référer, plus ou moins directement, au problème du filtrage. Nous avons voulu en citer quelques uns pour montrer, dans cet état de l'art, que le débruitage de maillages 3D ne se limite pas forcément aux opérateurs de filtrage présentés dans les sections précédentes. Des références vers d'autres travaux du même genre peuvent être trouvées dans ces quelques articles.

3 Bilan

Concernant l'intégration, les méthodes volumiques semblent beaucoup plus simples à mettre en œuvre : la fusion des différentes acquisitions en une surface unique est immédiate, ce qui n'est pas le cas des algorithmes basés maillages, où l'étape de couture n'est pas un procédé trivial. Il en va de même pour le calcul d'une surface consensuelle, où la combinaison linéaire de plusieurs champs scalaires semble largement suffire à générer des modèles sans artefact [189].

Cependant, ces méthodes volumiques ont également des défauts. En premier lieu, la qualité

du maillage reconstruit dépend de la finesse de la grille. Une grille trop grossière risque, en effet, de faire disparaître certains détails de la géométrie mais, à l'inverse, si la grille est trop fine, un maillage contenant une quantité de triangles beaucoup trop importante sera généré. Déterminer la résolution optimale n'est donc pas chose triviale. Ensuite, comme nous l'avons déjà dit, ces méthodes consomment généralement beaucoup de mémoire. Même si des solutions existent, lorsque la précision requise est grande ou lorsque les modèles sont volumineux, cela n'en demeure pas moins problématique. L'implantation de mécanismes hors mémoire devient alors impératifs. Enfin, l'utilisation d'algorithmes de détournement est indispensable pour revenir à une représentation surfacique. La surface générée n'a d'ailleurs plus rien à voir avec les maillages issus des différentes acquisitions, puisqu'elle est directement liée à la discrétisation de la grille de voxel. Il est d'ailleurs impossible de revenir à l'information initiale des points de mesure, contrairement aux méthodes basées maillages qui, à quelques rares exceptions près, travaillent directement sur ces points.

Le choix entre intégration volumique ou intégration basée maillages dépend donc de l'application, et notamment du fait que l'on ait besoin ou non de conserver les échantillons de mesure. Il faut également prendre en compte le fait que la représentation volumique permet d'effectuer certains traitements à moindre coût : par exemple, filtrer le maillage peut se faire par l'application de filtres linéaires 3D sur la grille de voxels avant l'extraction de la surface, et différents niveaux d'échelle de l'objet (maillage avec plus ou moins de triangles) peuvent être reconstruits en faisant simplement varier la résolution de la grille.

Concernant le filtrage, il est clair que les filtres adaptatifs surpassent largement les autres types de filtres, ce qui est normal dans la mesure où ils constituent une évolution logique de ces premiers filtres plus basiques. L'intérêt qu'ils présentent est d'autant plus important que la géométrie des objets numérisables peut être tout à fait quelconque. Il faut cependant noter que la plupart de ces filtres demandent à l'utilisateur de saisir des paramètres, comme la variance du bruit dans le cas du filtre MMSE [112]. Le choix de ces valeurs n'a généralement rien d'intuitif, et le bon jeu de paramètres n'est souvent obtenu que par tâtonnement, à l'issue de nombreux essais.

De plus, ces filtres ignorent certaines caractéristiques propres aux données issues d'une numérisation par télémétrie optique :

- ils ne tiennent pas compte du fait que le bruit est généralement plus fort dans la direction des lignes d'observation du capteur,
- contrairement au processus d'intégration, les algorithmes de filtrage n'exploitent pas la redondance disponible dans les zones de recouvrement, alors que cette redondance pourrait permettre, justement, d'estimer certains des paramètres si difficiles à apprécier manuellement.

- Capture et restitution de l'apparence -

Représentation de l'apparence en synthèse d'images

et utilisation de données réelles

L'apparence d'un objet ne se définit pas simplement par des assertions telles que «cet objet est bleu» ou «cet objet est brillant», mais par les lois régissant l'interaction entre la lumière et la matière, et par les propriétés optiques des matériaux qui le composent. Dans ce chapitre, nous allons donc commencer, dans la section 1, par définir ce qu'est la *réflectance* d'un matériau, élément fondamental permettant d'en caractériser le comportement face à la lumière. Nous détaillerons alors certains des modèles proposés dans la littérature pour représenter une telle quantité, ainsi que certaines techniques permettant d'en faire l'acquisition.

Ces modèles et ces techniques ne sont cependant utilisables que si les objets considérés ne sont constitués que d'un unique matériau uniforme, ce qui est rarement le cas pour des objets réels. En effet, les objets de la vie courante sont souvent composés de matériaux présentant une *variation spatiale*, c'est-à-dire que chaque point de leurs surfaces possède sa propre réponse photométrique. L'apparence ne peut donc pas être caractérisée pour l'ensemble de cette surface à l'aide d'un unique jeu de paramètres, ni même acquise en un unique point. Nous discuterons donc, dans la section 2, des moyens matériels permettant d'acquérir les matériaux de telles surfaces, où la variation spatiale est présente, ainsi que des problématiques que cela soulève.

Comme nous allons le voir, ce que caractérise la réflectance est en fait la dépendance de l'apparence d'un matériau aux conditions d'éclairage, mais également aux conditions d'observation : en effet, pour des surfaces brillantes, de nombreux reflets suivent le déplacement de l'observateur, même lorsque l'éclairage reste fixe. Tenir compte à la fois de cette double dépendance et de la variation spatiale implique de nombreuses difficultés, autant au niveau de l'acquisition que de la conception de modèles permettant la visualisation de tels matériaux.

Ainsi, nous présenterons dans la section 3 quelques modèles significatifs qui ont été proposés dans le but de tenir compte de tous ces paramètres à la fois. Nous discuterons ensuite

de modèles simplifiés, travaillant soit en éclairage fixe, soit en point de vue fixe (section 4). L'objectif est, bien entendu, de réduire le nombre de paramètres des modèles en éliminant l'une des deux dépendances, quitte à limiter les possibilités de visualisation pour les objets ainsi acquis. Enfin, nous présenterons des méthodes permettant l'acquisition d'une texture diffuse, c'est-à-dire qu'un degré de simplification supplémentaire est franchi en ne considérant cette fois qu'une information chromatique fixe, ne tenant finalement compte d'aucun effet lié à l'illumination (section 5). Il faut d'ailleurs savoir que, à l'heure actuelle, ce sont ces dernières méthodes qui sont employées lors des grandes campagnes de numérisation, c'est-à-dire lorsque les conditions de mesures en laboratoire ne sont plus disponibles.

1 Définition de l'apparence

■ 1.1 La fonction de réflectance

Le comportement d'un matériau par rapport à la lumière est décrit par sa fonction de *réflectance* f_r , qui exprime la capacité de ce matériau à réfléchir l'énergie lumineuse incidente.

Considérons la géométrie de la figure 5.1, représentant l'élément infinitésimal dS d'une surface opaque et non émissive, illuminée par une luminance¹ incidente $L_i(\theta_i, \phi_i)$ arrivant selon une direction (θ_i, ϕ_i) au travers d'un angle solide infinitésimal $d\omega_i$. L'éclairement $dE(\theta_i, \phi_i)$ de la surface dS induit par l'illumination provenant de $d\omega_i$ est alors défini par l'équation 5.1 :

$$dE(\theta_i, \phi_i) = L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i \quad (5.1)$$

Si $L_r(\theta_r, \phi_r)$ est la luminance lumineuse quittant la surface dS dans la direction (θ_r, ϕ_r) , la fonction de réflectance peut être calculée comme étant le rapport entre la luminance lumineuse réfléchie et l'éclairement :

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{L_r(\theta_r, \phi_r)}{dE(\theta_i, \phi_i)} = \frac{L_r(\theta_r, \phi_r)}{L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i} \quad (5.2)$$

En d'autres termes, la fonction de réflectance permet de représenter quelle est la quantité d'énergie lumineuse arrivant sur une surface depuis une direction donnée qui est renvoyée dans une autre direction. Elle caractérise donc effectivement le comportement d'un matériau par rapport à la lumière.

Du fait de la double dépendance aux directions d'incidence et de réflexion, on parle du caractère *bidirectionnel* de la réflectance. Nous emploierons donc par la suite, sans distinction, les termes *réflectance*, *fonction de réflectance*, *fonction de réflectance bidirectionnelle* ou l'abréviation *BRDF* (de l'anglais : *Bidirectional Reflectance Distribution Function*).

¹Voir le glossaire photométrique en annexe A.

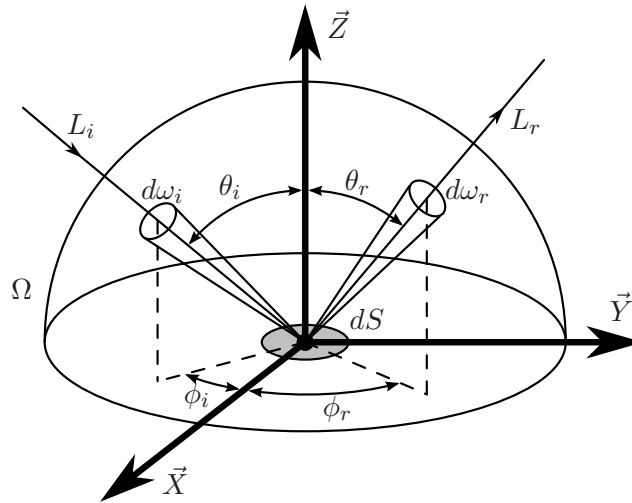


Figure 5.1 – La réflectance est définie à partir du flux E d'énergie lumineuse arrivant sur la surface selon une direction d'incidence (θ_i, ϕ_i) et du flux L d'énergie quittant cette surface dans la direction de réflexion (θ_r, ϕ_r) .

Ainsi, si l'élément de surface dS placé dans un environnement lumineux quelconque est composé d'un matériau dont la réflectance est connue, la luminance lumineuse émise dans la direction (θ_r, ϕ_r) par dS est obtenue en intégrant le transfert d'énergie sur l'ensemble des directions d'incidence lumineuse possibles :

$$L_r(\theta_r, \phi_r) = \int_{\Omega} f_r(\theta_i, \phi_i, \theta_r, \phi_r) L_i(\theta_i, \phi_r) \cos \theta_i d\omega_i \quad (5.3)$$

$$= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} f_r(\theta_i, \phi_i, \theta_r, \phi_r) L_i(\theta_i, \phi_r) \cos \theta_i d\theta_i d\phi_i \quad (5.4)$$

où Ω correspond à l'hémisphère défini au-dessus de la surface dS (ayant posé l'hypothèse que dS était une surface opaque, seul l'hémisphère supérieur peut apporter sa contribution à l'éclairage). En synthèse d'images, l'équation 5.4 est connue sous le nom d'*équation d'illumination locale*. Sa dérivation exacte à partir des quantités photométriques standards peut être retrouvée dans l'article original de Kajiya [88].

L'apparence d'une surface quelconque pour n'importe quelle direction d'observation et n'importe quelles conditions d'illumination peut donc être déterminée, sous réserve que la réflectance de cette surface soit connue.

Il est à noter que la fonction de réflectance obéit à deux propriétés physiques fondamentales :

- La *conservation de l'énergie*, selon laquelle la somme des quantités d'énergie respectivement transmise², absorbée et réfléchi est égale au total de l'énergie incidente,
- La *réciprocité de Helmholtz*, selon laquelle la valeur de la fonction de réflectance est identique si les directions d'incidence et de réflexion sont interverties.

²La transmission ne se produit que dans le cas de surfaces translucides, hypothèse que nous avons éliminée ici.

■ 1.2 Mesure de réflectances

Mesurer la réflectance de surfaces réelles nécessite un appareillage très perfectionné, appelé *gonioréfectomètre*. Cet appareil est composé d'un *photomètre*³, d'une source lumineuse, ainsi que d'un mécanisme permettant le positionnement de ces deux éléments sur l'hémisphère des directions avec une très grande précision. La source lumineuse doit se rapprocher le plus possible d'une source ponctuelle de manière à réduire autant que faire se peut l'angle solide incident $d\omega_i$. La mesure se fait alors en posant l'échantillon plan d'un matériau au centre de l'hémisphère et en faisant varier les positions du photomètre et de la source de manière à collecter des échantillons de réflectance pour différents couples de directions d'incidence et d'observation $\{(\theta_{i,0}, \phi_{i,0}, \theta_{r,0}, \phi_{r,0}), \dots, (\theta_{i,N}, \phi_{i,N}, \theta_{r,N}, \phi_{r,N})\}$.

Vu la précision demandée, un appareillage de ce genre est extrêmement coûteux, d'autant plus si l'on souhaite que la mesure se fasse de manière automatique, auquel cas un montage robotisé est indispensable. Ainsi, on trouve dans la littérature des dispositifs alternatifs développés dans le but de permettre l'acquisition d'une telle information à moindre coût. Ward [192] propose une glace sans tain de forme hémisphérique positionnée au-dessus de l'échantillon à mesurer. Juste à côté de l'échantillon se trouve une caméra numérique munie d'une lentille *fish-eye*. L'échantillon est éclairé à travers le miroir grâce à une source lumineuse ponctuelle. Lors de la mesure, la radiance émise par l'échantillon se reflète sur l'hémisphère et vient heurter la lentille, capturant en une seule image toutes les directions d'observation à la fois. Ainsi, seule la source a besoin d'être positionnée, ce qui peut être accompli manuellement en des temps raisonnables. L'auteur rapporte que seules quelques minutes sont nécessaires à la capture d'une BRDF.

Marschner *et al.* [111] proposent un dispositif encore moins onéreux, qui utilise la géométrie des échantillons pour réduire les degrés de liberté nécessaires lors de l'acquisition. Seules une caméra numérique et une source lumineuse ponctuelle sont nécessaires. La caméra capture des clichés d'un échantillon non planaire dont la géométrie est connue (soit parce qu'elle est assez simple pour être décrite par une surface paramétrique, soit parce qu'elle a été numérisée), et les positions relatives de la source et de la caméra par rapport à cet échantillon peuvent être extraites pour chaque cliché à l'aide de mires placées dans la scène. Pour chaque pixel correspondant à un point de la surface, les directions d'incidence et d'observation exprimées dans le repère local de ce point seront différentes. Ainsi, chaque cliché permet d'extraire un grand nombre d'échantillons de réflectance à la fois. La géométrie de l'échantillon doit néanmoins présenter le moins de concavités possible, afin d'éviter les problèmes d'interréflexion (réflexion de la lumière par la géométrie sur elle-même).

■ 1.3 Modèles de réflectances en synthèse d'images

La réflectance est une information complexe : en nous plaçant, comme nous l'avons fait jusqu'à présent, dans le cas d'une réflectance monochromatique (indépendance à la longueur d'onde) et en omettant la polarisation, il nous faut tout de même quatre dimensions pour

³Appareil mesurant les grandeurs lumineuses en fonction de la courbe de sensibilité de l'œil.

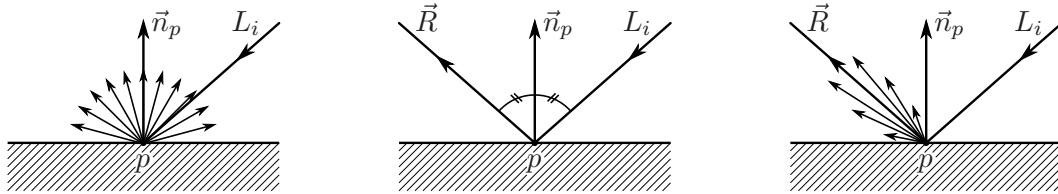


Figure 5.2 — **Gauche** : réflexion Lambertienne. L'énergie lumineuse incidente est renvoyée en égale proportion dans toutes les directions. **Milieu** : réflexion spéculaire idéale. L'énergie est intégralement réfléchi dans la direction \vec{R} de réflexion idéale. **Droite** : réflexion spéculaire dans le cas général. L'énergie est principalement renvoyée dans la direction de réflexion idéale, mais peut être dispersée du fait de la rugosité de la surface.

la définir (deux pour la direction d'incidence, et deux pour la direction d'observation). En synthèse d'images, la polarisation est généralement ignorée, et la dépendance à la longueur d'onde est approximée en discrétisant le spectre lumineux en bandes (modèle trichromatique RVB, par exemple) et en définissant une fonction de réflectance pour chaque bande. Nous allons donc considérer dans la suite de ce document cette représentation 4D comme une approximation valide.

La mesure d'une information 4D par les méthodes que nous avons vues plus haut fournit donc une quantité de données très importante, et difficile à exploiter directement, notamment dans le cas d'un rendu temps réel. Différents modèles mathématiques ont donc été proposés pour représenter la fonction de réflectance de manière analytique.

Dans la réalité, on peut observer deux types de réflexion : la réflexion *diffuse* et la réflexion *spéculaire*. La réflexion diffuse caractérise les matériaux dits mats, c'est-à-dire ceux dont l'apparence ne dépend que de l'orientation relative de la surface par rapport à la source. La réflexion spéculaire, quant à elle, caractérise les matériaux brillants, et correspond aux reflets qui bougent sur la surface en même temps que l'observateur se déplace.

La réflexion diffuse a été décrite il y a plusieurs siècles par le modèle de Lambert considérant que, pour de telles surfaces, la lumière incidente était réfléchi en égale quantité dans toutes les directions (figure 5.2-gauche). Ce modèle est ainsi décrit par une fonction de réflectance constante $f_r(\theta_i, \phi_i, \theta_r, \phi_r) = \rho_d$. Oren *et al.* [137] ont cependant montré que la réflexion purement Lambertienne ne permettait pas toujours de modéliser le comportement exact de surfaces réelles, du fait qu'elle ne tenait pas compte de la rugosité de la surface à un niveau microgéométrique. En utilisant une distribution aléatoire de microfacettes Lambertiennes pour représenter cette microgéométrie, et en considérant l'orientation relative des facettes par rapport aux directions d'incidence et d'observation, ainsi que le masquage et l'ombrage des différentes facettes entre elles, les auteurs ont pu élaborer un modèle de réflectance pour les matériaux diffus donnant des résultats visuels beaucoup plus proches de ce que l'on peut observer dans la réalité.

Concernant la réflexion spéculaire, elle est décrite dans le cas de surfaces parfaitement lisses par les lois de Snell-Descartes, selon lesquelles tout rayon lumineux incident est réfléchi dans la direction opposée par rapport à la normale à la surface (direction de réflexion

idéale), comme l'illustre la figure 5.2-milieu. Dans le cas concret, toutes les surfaces ne sont évidemment pas parfaitement spéculaires. La lumière est principalement réfléchiée dans la direction de réflexion idéale, mais la rugosité sous-jacente engendre une dispersion de la lumière dans un certain cône autour de cette direction (figure 5.2-droite). Phong [138] propose un modèle permettant de représenter la composante spéculaire de surfaces qui ne sont pas des réflecteurs parfaits. Ce modèle considère la distance angulaire entre la direction d'observation et la direction de réflexion idéale pour modéliser la dispersion lumineuse sous la forme d'un lobe :

$$f_r(\vec{v}, \vec{l}) = \rho_d + \rho_s \cos^n \widehat{\vec{v} \vec{l}_r} = \rho_d + \rho_s (\vec{v} \cdot \vec{l}_r)^n \quad (5.5)$$

où :

- $\vec{v} = (\sin \phi_r \cos \theta_r, \sin \phi_r \sin \theta_r, \cos \phi_r)^T$ est le vecteur unitaire orienté vers l'observateur,
- $\vec{l} = (\sin \phi_i \cos \theta_i, \sin \phi_i \sin \theta_i, \cos \phi_i)^T$ est le vecteur unitaire orienté vers la source lumineuse,
- \vec{n} est la normale à la surface au point considéré,
- $\vec{l}_r = 2\vec{n}(\vec{l} \cdot \vec{n}) - \vec{l}$ est la direction de réflexion idéale,
- n est le paramètre servant à moduler la forme du lobe. Plus n est grand, plus le lobe est fin, et plus la surface semble brillante.

Le modèle de Phong a cependant deux inconvénients :

- tous les matériaux qu'il représente ont une apparence plastique,
- du fait que ce modèle ait été construit de manière purement empirique, il ne respecte ni la propriété de conservation de l'énergie, ni le principe de réciprocité.

Cook *et al.* [33] proposent alors, de la même manière que Oren *et al.* [137], un modèle basé sur une distribution aléatoire de microfacettes pour représenter la rugosité de la surface et dériver un terme de réflexion spéculaire. Leur modèle permet aussi bien de représenter des matériaux plastiques que des matériaux métalliques. De plus, les auteurs expliquent que le modèle proposé respecte, cette fois, le principe de réciprocité⁴.

Les modèles de BRDFs précédents sont tous isotropes, c'est-à-dire que la valeur de la réflectance en un point ne change pas lorsque la source lumineuse et l'observateur tournent en même temps autour de la normale de ce point. En d'autres termes, la fonction de réflectance, dans le cas isotrope, ne dépend que de trois paramètres : θ_r , θ_i et $\phi_r - \phi_i$. Les modèles isotropes ne permettent donc pas de représenter tous les types de matériaux. En effet, certains textiles, comme le velours ou la soie, ou les surfaces en métal brossé présentent des caractéristiques anisotropes. Ward [192] et Ashikhmin [4] proposent tous deux des modèles de réflectance permettant de tenir compte du comportement de tels matériaux. Ward démontre notamment l'applicabilité de son modèle pour l'estimation à partir de données mesurées.

Dans le but de proposer un modèle général, capable de représenter n'importe quel type de matériaux, Lafortune *et al.* [93] proposent une généralisation du modèle de lobes spéculaires de Phong. Ce modèle respecte à la fois la réciprocité et la conservation, et peut tenir compte

⁴Pour ce qui est de la conservation de l'énergie, Lewis [99] montrera plus tard qu'elle ne peut être obtenue, du fait que les rebonds secondaires des rayons lumineux sur la microgéométrie sont ignorés par le modèle.

de l'anisotropie. Le produit scalaire entre le vecteur d'observation et la direction de réflexion idéale est d'abord représenté sous la forme matricielle suivante :

$$f_r(\vec{l}, \vec{v}) = \rho_s(\vec{l}^T M \vec{v})^n \quad (5.6)$$

Les auteurs montrent que si l'on se place dans un repère local dont les deux axes tangents sont alignés avec les principales directions d'anisotropie, la matrice M est diagonale et l'expression du lobe se simplifie comme dans l'équation 5.7 :

$$f_r(\vec{l}, \vec{v}) = \rho_s(C_x v_x l_x + C_y v_y l_y + C_z v_z l_z)^n \quad (5.7)$$

où C_x , C_y et C_z représentent les trois éléments de la diagonale de M . Une version du modèle généralisée à plusieurs lobes est finalement présentée :

$$f_r(\vec{l}, \vec{v}) = \sum_{k=1}^N (C_{k,x} v_x l_x + C_{k,y} v_y l_y + C_{k,z} v_z l_z)^{n_k} \quad (5.8)$$

Sa simplicité et le fait qu'il puisse être estimé à partir de données réelles ont rendu ce modèle très populaire dans la communauté de la synthèse d'images, notamment pour ce qui est du rendu temps réel.

Il existe, bien entendu, de nombreux autres travaux portant sur les modèles de représentation de BRDFs. Nous nous sommes cependant limités, dans notre description, à ceux qui sont les plus populaires, ou qui nous seront utiles par la suite. Les références bibliographiques fournies dans les articles [4, 33, 137, 192] peuvent être consultées pour plus d'informations sur ce sujet.

2 Acquisition de l'apparence à partir d'objets réels

Les techniques d'acquisition présentées dans la section précédente sont dédiées à la mesure très précise d'échantillons composés d'une unique BRDF, c'est-à-dire d'une fonction de réflectance donnée ne permettant de caractériser qu'un matériau donné. Or, les objets de la vie courante étant rarement composés d'un unique matériau uniforme, il est nécessaire, pour caractériser l'apparence de tels objets dans son ensemble, de considérer chaque point de la surface comme étant constitué de son matériau propre. Ainsi, l'apparence d'un objet est décrite par une fonction dépendant à la fois de la direction d'illumination, de la direction d'observation et de la position sur la surface. En plus des quatre dimensions de la fonction de réflectance, il faut alors ajouter deux nouvelles dimensions pour décrire les coordonnées paramétriques de variation spatiale.

En considérant que la géométrie d'un objet réel ait été préalablement capturée, l'acquisition de son apparence nécessite un échantillonnage de cette fonction 6D. Cela se fait généralement à l'aide de plusieurs clichés photographiques, capturés depuis des points de vue

différents, et pour des conditions variables d'illumination. Ainsi, pour chacun des clichés acquis, il faut pouvoir :

- localiser chaque point de vue par rapport à la géométrie,
- localiser la source lumineuse par rapport à la géométrie.

De plus, afin de pouvoir associer les valeurs de luminance perçues par la caméra à chaque primitive géométrique du modèle, les paramètres intrinsèques de la caméra doivent également être connus. Ces paramètres peuvent être déterminés soit par un calibrage préalable, soit par le procédé de localisation. On parle alors plutôt de *recalage de point de vue* que de localisation. L'association de la luminance à la géométrie se fait alors en projetant des morceaux d'images sur le modèle 3D⁵. Pour cette raison, le terme *texture* est souvent utilisé en informatique graphique pour décrire l'information photométrique/chromatique associée à une surface.

Déterminer la position de la caméra ou de la source lumineuse ne constitue évidemment pas un problème dans le cas où un dispositif de localisation est utilisé [35, 37, 69, 92, 97, 114]. Cependant, nous avons déjà évoqué, dans la section 3 sur le recalage géométrique, que de tels dispositifs engendraient de nombreuses contraintes dans le cas d'un déploiement sur site, et ne sont finalement utilisables que pour des mesures en laboratoire. Nous allons donc discuter des méthodes alternatives couramment utilisées pour résoudre ces problèmes. Nous allons également voir que les textures recalées de manière naïve sur un modèle à partir de plusieurs points de vue ne sont pas toujours parfaitement alignées. Nous allons donc également discuter de leur ajustement.

■ 2.1 Recalage de points de vue

La localisation de la caméra d'acquisition s'apparente, bien entendu, à un problème de calibrage caméra : s'il est possible d'établir des correspondances entre l'espace image d'une photographie et le modèle 3D, les paramètres caméra associés à cette photographie peuvent facilement être déterminés (voir chapitre 2).

Ainsi, le recalage peut être fait manuellement, en demandant à l'opérateur de spécifier, pour chaque image, un nombre minimum de correspondances avec le modèle 3D [129, 150]. Bien entendu, dès lors que le nombre de photographies augmente, cette approche devient très vite longue et fastidieuse, malgré les outils spécifiquement développés pour l'aide au recalage manuel d'images [49]. Il est également possible, comme dans le cas du recalage géométrique, d'utiliser des mires pour établir automatiquement ces correspondances [26, 60]. Cependant, selon les dimensions de l'objet à mesurer ou l'agencement du site sur lequel se déroule l'acquisition, le bon positionnement de ces mires est un problème ardu, voire insoluble.

Lensch *et al.* [95] et Neugebauer *et al.* [129] ont proposé des algorithmes de localisation basés sur la mise en correspondance de silhouettes. Connaissant le contour de l'objet dans l'image, l'idée est alors de déterminer le point de vue pour lequel l'écart entre cette sil-

⁵Inversement, le processus peut être vu comme une paramétrisation du modèle dans l'espace image.

houette et celle du modèle 3D projeté sur le plan image de la caméra est minimum. Alors que Neugebauer *et al.* demandent qu'une première approximation du point de vue soit fournie, la méthode de Lensch *et al.* produit un résultat de manière entièrement automatique, ce qui en démontre l'utilisabilité. Cependant, ces méthodes posent deux problèmes :

- l'objet doit pouvoir être correctement dissocié du fond dans la photographie. Cela nécessite un environnement d'acquisition contrôlé (objet placé devant un fond uniforme) qu'il n'est pas toujours possible de mettre en place dans le cas d'une mesure sur site.
- les objets présentant des symétries sont généralement difficiles à gérer, du fait de la non unicité de la solution dans de tels cas.

On peut également noter que certains travaux utilisent comme caméra d'acquisition la caméra embarquée dans le scanner 3D [10, 12, 122]. Une photographie est prise à chaque fois qu'une image télémétrique est capturée, ce qui permet de résoudre la détermination des points de vue en même temps que le recalage géométrique. Cette approche est d'ailleurs exploitée par Bernardini *et al.* [10] à double sens, en raffinant le recalage géométrique à l'aide des attributs chromatiques afin d'obtenir une grande précision, autant sur la géométrie reconstruite que sur sa mise en corrélation avec les photographies. Cette méthode à l'avantage de simplifier considérablement le problème, mais elle interdit l'acquisition des deux informations (géométrie et texture) à des résolutions différentes.

■ 2.2 Localisation de sources lumineuses

La localisation de la source lumineuse est un problème plus complexe que celui de la localisation du point de vue. En effet, dans ce dernier cas, il s'agit de localiser un capteur, c'est-à-dire un dispositif fournissant une mesure de la scène par rapport à laquelle sa position doit être déterminée. Dans le cas d'une source lumineuse, nous avons affaire à un émetteur, incapable, par définition, de nous fournir de lui même une quelconque information. L'estimation de la position de la source doit donc se faire au travers de l'image perçue par la caméra d'acquisition.

Une fois de plus, le problème peut être résolu à l'aide de mires. Masselus *et al.* [113], par exemple, disposent plusieurs sphères blanches mates de dimensions connues dans la scène. En reprojétant la géométrie de la sphère dans l'espace image de la caméra d'acquisition, chaque pixel atteint peut se voir associer la normale et l'intensité lumineuse du point correspondant sur la sphère. En considérant que la réflectance de la sphère est décrite par un modèle Lambertien, la position de la source peut être estimée au sens des moindres carrés grâce à l'ensemble de ces pixels.

D'autres techniques d'estimation de la position d'illuminants dans une scène filmée existent. Celles-ci, au lieu de travailler sur des mires de géométrie et de matériau parfaitement connus, utilisent directement l'objet d'intérêt [16, 67, 118]. Le principe est alors le même que dans le cas de la sphère diffuse, si ce n'est que la géométrie considérée est arbitraire (donc potentiellement sujette aux interférences lumineuses) et que la réponse photométrique de son matériau est inconnue. Pour aboutir à un résultat, ces méthodes estiment alors conjointement la réflectance en chaque point et la position de l'illuminant. Si cela peut

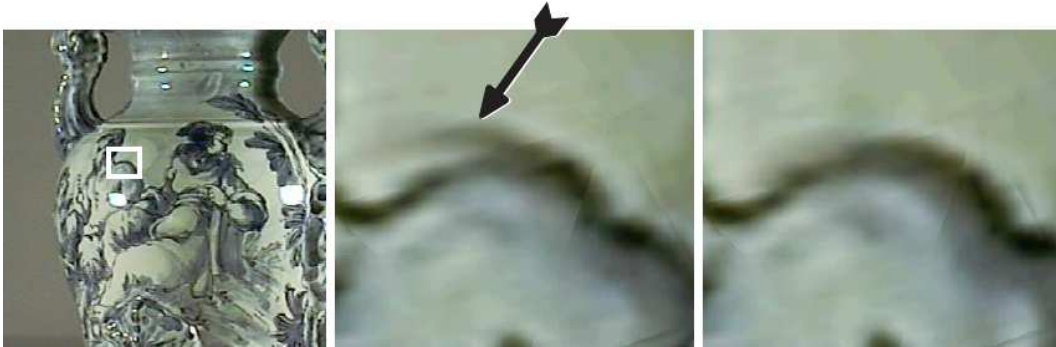


Figure 5.3 — Problème d'ajustement des textures provenant de différents points de vue (images tirées de [150]). **Gauche** : l'objet numérisé. **Milieu** : artefact généré par le mauvais alignement de deux textures. **Droite** : résultat obtenu après ajustement.

sembler attrayant, il faut cependant remarquer que plusieurs heures de calcul sont souvent nécessaires pour traiter une seule image, et que la précision du résultat dépend beaucoup de la scène analysée.

Dans certains cas très spécifiques, le problème de localisation est réglé de manière très simple en fixant une ou plusieurs sources à la caméra à l'aide d'une armature rigide [151, 154, 155, 156].

En dehors de ces quelques approches, il n'existe pas, à notre connaissance, d'autres moyens d'estimer la position d'une source lumineuse par rapport à une scène.

■ 2.3 Ajustement de textures

Dès lors que chaque photographie a été recalée sur la géométrie, par détermination des paramètres caméra associés, l'information photométrique peut être récupérée pour chacune d'elles en projetant le modèle 3D dans l'espace image. On obtient ainsi une texture pour chaque point de vue. Même dans le cas où le calibrage caméra est très précis, ces différentes textures ne s'ajustent pas toujours très bien entre elles, et cela peut générer des artefacts visuels dans la texture finale, comme dans l'exemple de la figure 5.3. Il existe plusieurs raisons à cela, notamment le fait que la géométrie subit tellement de traitements avant d'aboutir au modèle final que celui-ci ne correspond jamais exactement à l'objet réel, et ne se projette donc pas dans l'image exactement là où il le devrait. Il y a également des distorsions engendrées par la parallaxe : une même surface observée depuis un point de vue frontal et un point de vue rasant ne se projettera pas sur le même nombre de pixels dans l'image, ce qui va engendrer des dissimilarités entre les deux textures.

L'ajustement est généralement réalisé à l'aide d'un algorithme de corrélation d'images travaillant sur les voisinages, en espace texture, des projections d'un même point dans les différentes photographies [10, 150, 151]. La corrélation permet ainsi de modifier les coordonnées de projection de manière à améliorer l'alignement de l'information photométrique sur la surface de l'objet lorsqu'elle est redondante.

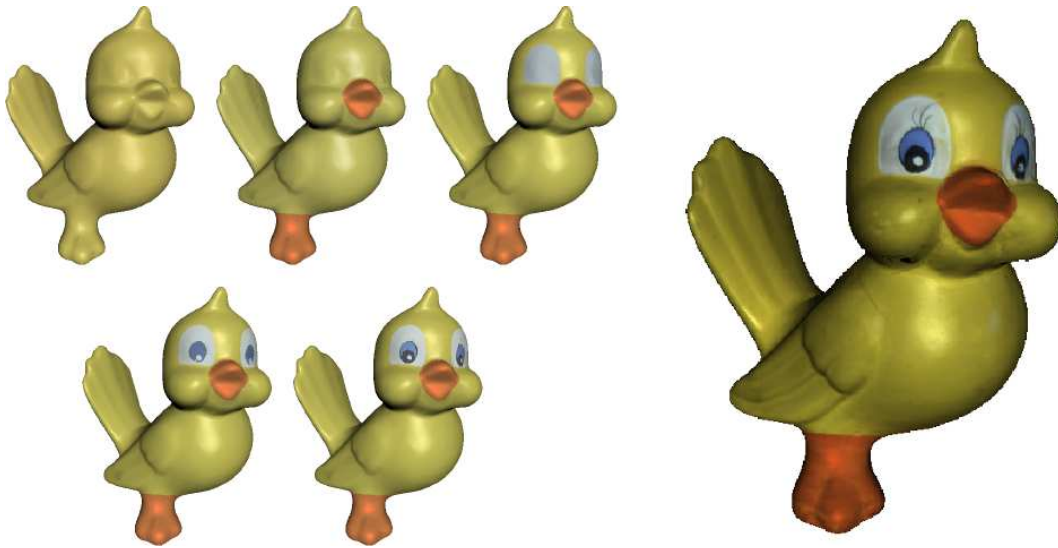


Figure 5.4 – Reconstruction de texture bidirectionnelle par la méthode de Lensch *et al.* [96]. **Gauche** : extraction progressive des principaux matériaux de l'objet. **Droite** : résultat final, après reconstruction des variations spatiales internes à chaque matériau.

3 Reconstruction de textures bidirectionnelles

Une texture de BRDFs, définissant la fonction 6D des matériaux à variation spatiale exposée précédemment, peut aisément être générée et manipulée dans le cas de données synthétiques [90, 116], mais sa reconstruction à partir de données réelles est autrement plus délicate. Sato *et al.* [164] et Erdem *et al.* [43] ont tenté d'apporter une réponse à ce problème, mais leurs solutions proposent toutes deux d'estimer un modèle de BRDF dont seule la couleur de la composante diffuse varie sur la surface, la partie spéculaire étant considérée comme uniforme sur tout l'objet.

Lensch *et al.* [96] proposent une représentation de la variation spatiale du matériau très compacte et très précise. L'algorithme commence alors par estimer un modèle de Lafortune f_r unique pour tout l'objet à partir de tous les échantillons extraits des photographies initiales. Les paramètres obtenus représentent, en quelque sorte, la réflectance moyenne de l'objet (tout en gardant à l'esprit que l'information bidirectionnelle qu'elle représente contient forcément des perturbations, dues aux interréllections sur la géométrie). Un processus de classification intervient alors pour extraire de nouveaux matériaux à partir de cette première estimation :

- L'estimation de f_r est obtenue grâce à l'algorithme de Levenberg-Marcquart (optimisation non linéaire), lequel fournit la matrice de covariance des échantillons par rapport aux paramètres estimés. Le vecteur propre \vec{e} associé à la plus grande valeur propre λ de cette matrice indique donc dans quelle direction les paramètres du modèle divergent le plus des échantillons. Deux nouvelles BRDFs sont alors créées à partir de f_r de manière à tenir

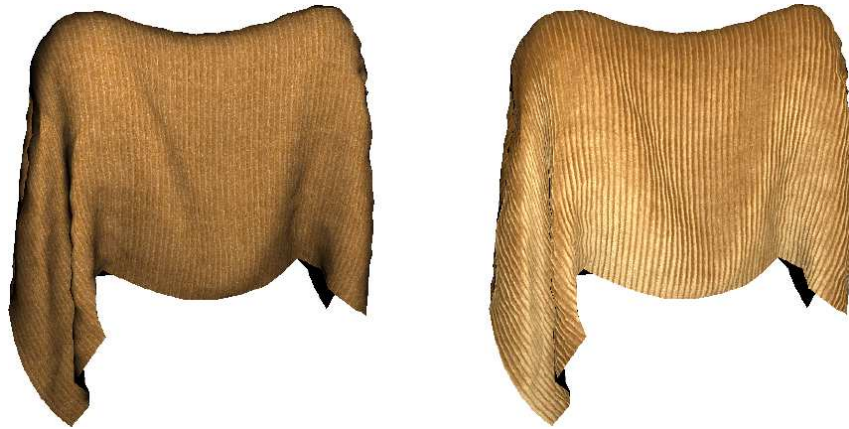


Figure 5.5 – Comparaison entre une texture traditionnelle (une simple photographie) à gauche, et une BTF à droite. La géométrie mésoscopique est restituée uniquement à partir d’images, simulant les phénomènes complexes de masquage, d’ombrage et d’interréflexion qui s’y produisent (images tirées de [125]).

compte de cette divergence :

$$f_{r,1} = f_r - \lambda \vec{e} \quad \text{et} \quad f_{r,2} = f_r + \lambda \vec{e}$$

- Chaque échantillon est alors associé à celle de ces deux nouvelles BRDFs qui le représente le mieux, selon un certain critère métrique.
- Ce processus de subdivision est répété jusqu’à ce qu’un nombre donné de classes ait été atteint.

Une fois que les matériaux principaux de l’objet ont été extraits, les petites variations qui peuvent être observées pour chacun d’eux sur la surface sont obtenues en construisant, pour chaque classe, une base de plusieurs BRDFs. Chaque pixel de la texture stocke alors les poids permettant de reconstruire sa propre BRDF par combinaison linéaire de celles formant la base de la classe à laquelle il appartient. Cette méthode est élégante, et les résultats produits sont réellement impressionnants, comme le montre la figure 5.4, mais elle nécessite de nombreuses heures de calcul pour la reconstruction complète de la texture.

Nous avons vu, un peu plus tôt, que l’allure de la fonction de réflectance était due aux interactions lumineuses se produisant au niveau microgéométrique de la surface sous-jacente. Certaines surfaces peuvent également présenter une géométrie fine, mais cependant visible, qui rend difficile leur restitution à l’aide d’une texture de BRDFs à variation spatiale. C’est le cas, par exemple, pour une écorce d’arbre, pour des mailles de tricot, où pour la surface d’une éponge. Cette géométrie *mésoscopique* entraîne, en effet, des phénomènes de masquage, d’ombrage, d’interréflexion et de diffusion interne qui sont beaucoup trop complexes pour pouvoir être estimés et découplés de la réflectance seule.

Ainsi, Dana *et al.* introduisent les *fonctions de textures bidirectionnelles* (ou *BTF*, de l’anglais *Bidirectional Texture Function*), qui sont des textures complexes capturant, uniquement de manière photométrique, les variations induites par la géométrie mésoscopique d’une surface pour toutes les conditions possibles d’éclairage et d’observation. Cette texture permet alors de simuler la mésostructure de l’objet uniquement à partir d’images, avec tous

les phénomènes complexes d'illumination qui s'y produisent (comme on peut le voir sur la figure 5.5), sans avoir à la capturer réellement d'un point de vue géométrique. À l'instar d'une BRDF à variation spatiale, une BTF est une fonction 6D, mais son interprétation sémantique est différente en ce sens qu'elle ne représente pas uniquement la réflectance en chaque point de la surface.

L'acquisition d'une BTF demande cependant un matériel de précision qu'il est difficile de déployer ailleurs qu'en laboratoire. Elle requiert également la capture de plusieurs centaines de photographie, ce qui la rend très gourmande en mémoire (plusieurs Go pour un petit échantillon de surface). Même si quelques travaux se sont intéressés à la compression de ces textures particulières [92, 120, 124], la complexité du processus d'acquisition et la taille des données sont telles que les BTFs sont systématiquement employées pour la capture de petits échantillons de surface, et jamais pour des modèles entiers.

4 Reconstruction de textures monodirectionnelles

Gérer le cas bidirectionnel étant relativement complexe, autant en ce qui concerne l'acquisition (problème du positionnement des sources) que la reconstruction et la visualisation de la texture, certaines catégories de méthodes choisissent de simplifier le problème en fixant soit le point de vue, soit l'éclairage, restreignant ainsi le nombre de degrés de liberté.

Bien entendu, les contraintes imposées au niveau de la capture le seront également au moment de la restitution : dans le premier cas, l'observateur ne pourra voir l'objet que sous l'unique point de vue d'acquisition, alors que dans l'autre cas, l'objet sera acquis dans son propre environnement lumineux et ne pourra donc pas être rééclairé différemment par la suite.

■ 4.1 Point de vue fixe

Fixer le point de vue permet en fait de s'absoudre totalement de la géométrie : les données en entrée consistant en un ensemble de photographies représentant la même prise de vue, mais sous des conditions d'illumination différentes, l'observateur est forcément contraint de se limiter à ce seul point de vue lors de la visualisation, rendant le modèle 3D obsolète. Ce type d'approches est d'ailleurs communément appelé *rééclairage d'images* [37, 52, 69]. En revanche, le problème de la localisation des sources reste toujours à traiter.

Hawkins *et al.* [69] mettent en œuvre une telle technique pour l'acquisition et l'étude d'artefacts culturels. Leurs travaux sont motivés par le fait que la géométrie est une information difficile à acquérir, voire impossible lorsque l'objet est constitué de matériaux transparents, hautement diffusants, ou très brillants. Ils estiment donc que, même si le point de vue est

contraint, rééclairer une image de manière réaliste peut déjà permettre de donner beaucoup d'informations à l'observateur. Ils présentent alors un dispositif d'acquisition constitué d'une rampe rotative de sources lumineuses, tournant autour de l'objet d'intérêt, ce qui leur permet de prendre des photos pour un grand nombre de directions d'illumination différentes. L'image peut ainsi être rééclairée pour des environnements lumineux complexes en combinant toutes ces photographies. Le dispositif d'acquisition présenté n'est malheureusement pas très mobile.

Malzbender *et al.* [109] proposent de représenter la variation lumineuse induite en chaque pixel par le déplacement d'une source ponctuelle à l'aide d'un polynôme quadratique fonction de la direction locale d'incidence lumineuse \vec{l} :

$$L_r(\vec{l}) = a_1 l_x^2 + a_2 l_y^2 + a_3 l_x l_y + a_4 l_x + a_5 l_y + a_6 \quad (5.9)$$

où l_x et l_y sont les coordonnées de \vec{l} dans le plan tangent du pixel. Ce modèle à l'avantage d'être très compact, puisque seulement six coefficients par pixel sont nécessaires. En revanche, il présente l'inconvénient de ne représenter que l'hémisphère supérieur d'incidence, et ne permet donc pas, comme la méthode de Hawkins peut le faire, de tenir compte de l'éclairage provenant de l'arrière dans le cas de matériaux diffusants. De plus, ce modèle ne permet qu'une interpolation douce, ce qui le rend peu adapté à la représentation de matériaux très brillants.

On peut noter les travaux de Dellepiane *et al.* [39] et Mudge *et al.* [123] utilisant ces textures polynomiales dans le cas d'applications relatives à l'archivage du patrimoine culturel.

■ 4.2 Éclairage fixe — approches purement photométriques

Fixer l'éclairage lors de l'acquisition de l'apparence revient à contraindre la restitution de la copie numérique à un environnement lumineux donné, tout en permettant à l'observateur de circuler librement autour de cette copie. Cela peut permettre, par exemple, de restituer une œuvre d'art avec l'apparence qu'elle présente sous les conditions d'éclairage du musée dont elle est issue.

L'environnement lumineux n'a donc besoin ni d'être connu, ni d'être contrôlé, ce qui permet de mettre au point des mécanismes d'acquisition plus légers que dans le cas du rééclairage d'images, puisque le problème de la localisation de la source disparaît. Seuls les points de vue ont besoin d'être connus.

Il existe deux types d'approches travaillant en éclairage fixe :

- les approches purement photométriques, qui ne travaillent qu'à partir d'images sans utiliser la géométrie de l'objet,
- les approches surfaciques, qui reconstruisent une texture dépendante du point de vue sur la base d'un modèle 3D, comme nous le verrons dans la sous-section suivante.

Il existe dans la littérature de nombreuses méthodes permettant d'interpoler des photographies afin de générer de nouveaux points de vue [25, 195], mais l'approche la plus connue en synthèse d'images, initiée par Levoy *et al.* [97] et Gortler *et al.* [60], est certainement

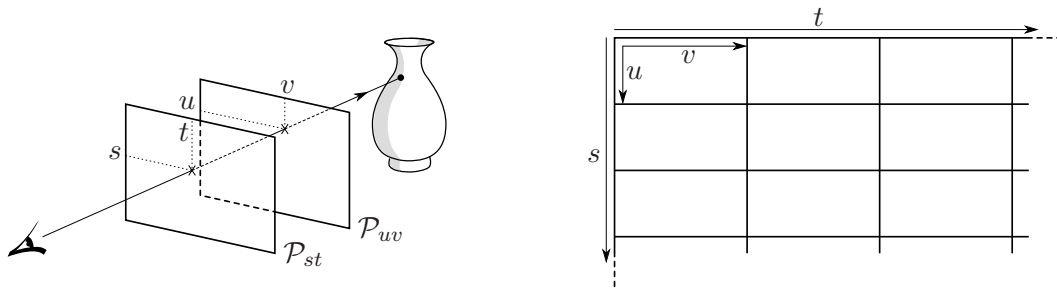


Figure 5.6 — Représentation d'un champ lumineux par une paramétrisation des rayons d'observation dans un espace 4D.

celle des *champs lumineux* (*light fields*), qui propose un moyen élégant de synthétiser de nouveaux points de vue.

Cette technique se base sur une paramétrisation de l'ensemble des rayons d'observation. La paramétrisation en question représente chaque rayon par quatre coordonnées (s, t, u, v) , correspondant aux intersections de ce rayons avec deux plans de l'espace \mathcal{P}_{st} et \mathcal{P}_{uv} parallèles entre eux, comme l'illustre la figure 5.6.

La luminance observée est donc représentée par une fonction dans un espace à quatre dimensions. Cet espace est alors discrétisé selon une grille 4D, et le champ lumineux est construit à partir de photographies en retrouvant, grâce à la paramétrisation (s, t, u, v) , l'élément de la grille correspondant au rayon passant en chaque pixel de la caméra d'acquisition. De la même manière, lors de la visualisation, de nouveaux points de vue peuvent être générés en cherchant les intersections avec les plans \mathcal{P}_{st} et \mathcal{P}_{uv} des rayons d'observation associés à chaque pixel de la caméra virtuelle, et ainsi retrouver les valeurs de luminance correspondantes. La figure 5.7 donne des exemples d'images générées à partir de champs lumineux. Deux plans parallèles ne permettant pas, à eux seuls, de paramétrer l'ensemble des rayons d'observation possibles, un cube est construit autour de l'objet d'intérêt, dont chacune des six faces est associée à une paramétrisation à deux plans afin de garantir une navigation libre autour de l'objet.

Les champs lumineux, même s'ils sont élégants, sont tout de même extrêmement gourmands en mémoire (plusieurs Go par modèle), ce qui est tout à fait normal compte tenu du fait qu'il s'agit d'échantillonner une fonction 4D. Des techniques de compression spécifiques permettent néanmoins d'en réduire drastiquement la taille [108, 107]. Un autre inconvénient, inhérent à toutes les méthodes d'interpolation de vues, vient de l'absence de gestion de la parallaxe : des effets de *ghosting*⁶ peuvent en effet apparaître lors de la visualisation du fait que la géométrie est ignorée, comme l'illustre la figure 5.8. La version de Gortler *et al.* propose justement de tenir compte de la géométrie, lorsque celle-ci est connue, pour corriger ce genre de défauts.

⁶Dédoublage de certains éléments de l'image lors de l'interpolation entre plusieurs vues, souvent dû à un échantillonnage trop grossier ou à une mauvaise mise en correspondance des vues impliquées.



Figure 5.7 — Exemples de vues synthétisées à l'aide de champs lumineux (images tirées de [60]).

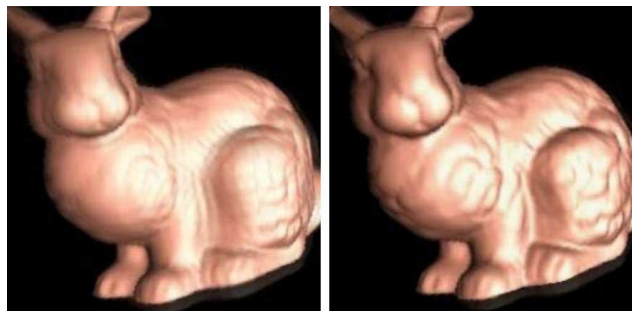


Figure 5.8 — **Gauche** : flou lié aux effets de *ghosting* introduits par l'absence de gestion de la parallaxe. **Droite** : correction apportée par Gortler *et al.* [60], tenant compte de la géométrie sous-jacente.

■ 4.3 Éclairage fixe — approches surfaciques

Par définition, les approches surfaciques, contrairement aux approches photométriques vues ci-dessus, nécessitent de connaître la géométrie de l'objet, ce qui n'est pas une information toujours très simple à acquérir. En revanche, elles ont l'avantage de ne représenter l'information que là où elle est réellement utile, c'est-à-dire sur la surface même de l'objet d'intérêt, ce qui peut conduire à une réduction du volume de données.

Debevec *et al.* [38] proposent de stocker, pour chaque face du modèle, l'ensemble des images dans lesquelles elle est visible. Une structure de données idoine permet de déterminer, au moment du rendu, les trois images dont les directions locales d'observation sont les plus proches du point de vue virtuel, et celles-ci sont alors interpolées pour générer, à la volée, la texture de la face considérée. L'inconvénient évident de cette approche est qu'elle demande impérativement de conserver toutes les photographies initiales, ce qui entraîne une consommation mémoire proportionnelle au nombre de prises de vue. Ainsi, la méthode n'est pas forcément plus économique qu'une approche purement photométrique. Cette dépendance peut éventuellement être levée grâce à la méthode de Nishino *et al.* [131], qui n'utilisent qu'un nombre K fixé d'images, construites à l'aide d'une analyse en composantes principales sur l'ensemble des photographies initiales.

Miller *et al.* [121] sont les premiers à formuler le problème des textures dépendantes du point à l'aide d'une paramétrisation 4D semblable à celle des champs lumineux. Ce modèle,

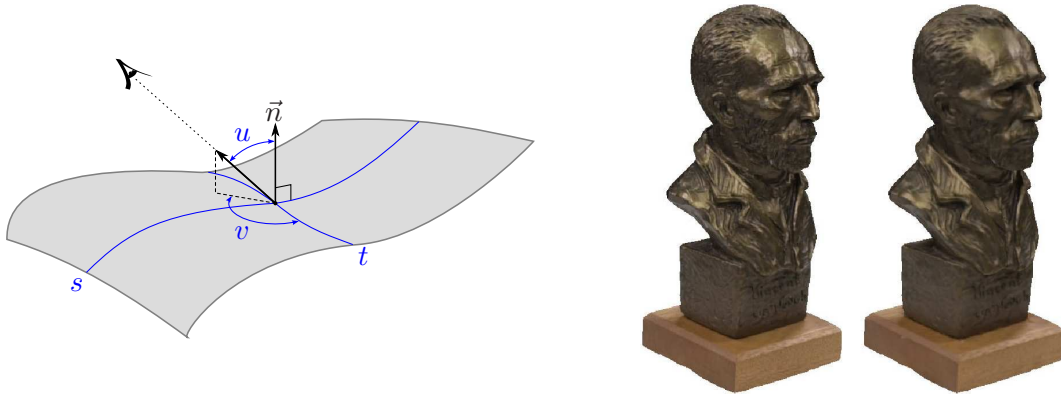


Figure 5.9 – Gauche : paramétrisation proposée par Miller *et al.* [121] pour définir les champs lumineux sur des surfaces. Droite : photographie d'un objet réel, et vue synthétisée à partir du champ lumineux surfacique capturé par la méthode de Chen *et al.* [26]. Seules trois composantes principales sont utilisées pour la reconstruction.

baptisé pour l'occasion *champ lumineux surfacique (surface light field)*, est illustré par la figure 5.9-gauche. (s, t) représente cette fois les coordonnées paramétriques d'un point sur la surface et (u, v) la direction d'observation, exprimée en coordonnées sphériques dans le repère local à ce point. Les données sont alors stockées dans une grille 4D, de la même manière (et en utilisant les mêmes techniques de compression) que pour les champs lumineux standards. Un mécanisme de chargement à la demande est finalement proposé, permettant une visualisation interactive en générant, à la volée, la texture d'un objet en fonction des directions locales d'observation en chaque point.

Chen *et al.* [26] proposent une représentation des champs lumineux surfaciques plus adaptée au rendu temps réel, et plus compacte que la version de Miller *et al.*. Pour ce faire, la fonction $f(s, t, u, v)$ définissant le champ lumineux surfacique est approximée par une somme de produits de deux fonctions de plus petite dimensionalité :

$$f(u, v, \theta, \phi) = \sum_{k=1}^K g_k(s, t) h_k(u, v) \quad (5.10)$$

où les ensembles de fonctions $\{g_k\}_{1 \leq k \leq K}$ et $\{h_k\}_{1 \leq k \leq K}$ encodent respectivement la variation spatiale et la variation liée au point de vue du champ lumineux surfacique. Disposant de $N \times M$ échantillons de la fonction f , pour N valeurs spatiales $\{(s_1, t_1), \dots, (s_N, t_N)\}$ et M directions d'observation $\{(u_1, v_1), \dots, (u_M, v_M)\}$, une matrice F peut être construite :

$$F = \begin{bmatrix} f(s_1, t_1, u_1, v_1) & \cdots & f(s_1, t_1, u_M, v_M) \\ \vdots & \ddots & \vdots \\ f(s_N, t_N, u_1, v_1) & \cdots & f(s_N, t_N, u_M, v_M) \end{bmatrix} \quad (5.11)$$

puis factorisée à l'aide d'une décomposition en valeurs singulières :

$$F = G H \quad (5.12)$$

Chaque ligne de la matrice G correspond à l'une des fonctions g_k , et chaque colonne de la matrice H à l'une des fonctions h_k . Chacune de ces fonctions est alors stockée sous la

forme d'une image dans une texture 2D, et l'équation 5.10 est évaluée à la volée par le matériel graphique.

Le point fort de la méthode est que le fait de ne garder que les K premières fonctions g_k et h_k de cette factorisation revient à effectuer une analyse en composantes principales du champ lumineux surfacique. Les auteurs montrent alors qu'une reconstruction d'ordre $K = 3$ ou 4 suffit pour produire des résultats visuellement très convaincants (figure 5.9-droite), n'impliquant du coup qu'un nombre très restreint de textures.

5 Reconstruction de textures diffuses

Une texture diffuse consiste en une simple image contenant l'information chromatique pour toute la surface de l'objet. Contrairement à toutes les techniques présentées précédemment, une telle texture ne tient compte ni de l'influence de l'éclairage, ni des conditions d'observation. Il s'agit donc, dans ce cas, de capturer la teinte pure de l'objet, dénuée de tout effet d'illumination.

■ 5.1 Combinaison d'images par moyennage

Même pour ce type de texture, l'acquisition nécessite plusieurs prises de vue, non pas pour capturer les variations dues au changement de point de vue mais simplement pour garantir que la texture pourra être reconstruite pour l'ensemble de la surface de l'objet. La texture finale doit alors être générée en combinant les différentes photographies acquises, tout en évitant les inconsistances telles que celles que l'on peut voir sur la figure 5.10-gauche.

Pour ce faire, Neugebauer *et al.* [129] effectuent une combinaison linéaire de différentes photographies dans les zones de la surface où plusieurs d'entre elles se chevauchent. Les poids utilisés sont alors choisis de manière à garantir des transitions douces d'une image à l'autre.

Rocchini *et al.* [150] adoptent une approche similaire, mais ils tentent d'abord de minimiser l'aire de la surface qui nécessitera une telle fusion. En effet, une moyenne pondérée entre différentes images tend à dégrader l'information initiale, notamment en introduisant du flou du fait des alignements imparfaits et des différences d'échantillonnage entre les photographies. Ainsi, les auteurs commencent par associer à chaque sommet du maillage l'image dans laquelle on le voit le plus de face (image engendrant la moindre distorsion). Utilisant cette information, les triangles du maillage sont ensuite classifiés en deux catégories :

- les *triangles internes*, qui correspondent aux triangles dont les trois sommets sont associés à la même image,
- les *triangles bordure*, qui correspondent à ceux dont les trois sommets sont chacun asso-



Figure 5.10 – Gauche : texture diffuse obtenue par un assemblage naïf de photographies. Milieu et droite : textures de normales et de coefficients de réflexion diffuse extraites par la méthode de Rushmeier *et al.* [154].

ciés à une image différente.

La reconstruction de la texture pour les triangles internes est immédiate : la photographie associée aux trois sommets est directement utilisée. La fusion par combinaison linéaire n'est finalement employée que pour les triangles bordure, en utilisant, pour ce faire, les poids barycentriques de chaque sommet.

La figure 5.11 illustre le résultat obtenu par cette méthode. Dans cet exemple, l'objet original a un matériau très brillant, ce qui permet immédiatement d'illustrer les défauts de ce genre de méthodes : on peut clairement voir des points lumineux (mis en valeur dans la figure par les flèches) apparaître dans la texture diffuse reconstruite. Ces points correspondent aux différents reflets spéculaires captés depuis chaque point de vue. Les photographies étant directement combinées sans chercher à supprimer de tels effets, ceux-ci demeurent forcément apparents dans la reconstruction finale.

L'approche de Callieri *et al.* [20] est similaire à celle de Rocchini, à ceci près qu'elle applique une légère correction de couleur pour éviter les dissimilarités chromatiques trop importantes entre les zones affectées à des images différentes. Cependant, le problème des reflets spéculaires n'est pas non plus pris en compte. Dans ce cas, il faut soit se restreindre à l'acquisition d'objets mats, soit travailler dans un environnement lumineux contrôlé, comme



Figure 5.11 — Reconstruction d'une texture diffuse par moyennage, sans contrôle de l'éclairage. Des artefacts visuels correspondant aux reflets spéculaires visibles depuis chacun des points de vue apparaissent (image tirée de [150]).

dans le cas de la technique présentée ci-après.

■ 5.2 Extraction de la composante diffuse par illumination inverse

Dans le but d'éliminer ces effets d'illumination pour produire une véritable texture diffuse, plusieurs travaux [151, 154, 155] se sont basés sur l'approche d'illumination inverse de Rushmeier *et al.* [156].

L'idée est de considérer, pour chaque prise de vue, plusieurs photographies (cinq ou six dans les travaux concernés), chacune correspondant à une direction d'illumination différente, mais parfaitement connue. Pour ce faire, la caméra et les sources lumineuses sont généralement montées sur une armature rigide et un calibrage est préalablement effectué pour déterminer la position relative de chaque source par rapport à la caméra. Pour chaque pixel de l'image, cinq à six valeurs de luminance sont ainsi recueillies.

Un premier filtrage de ces valeurs permet de supprimer les ombres portées et les reflets spéculaires : les valeurs sont triées par luminance croissante, et celles qui sont soit trop sombres soit trop saturées sont supprimées. Les valeurs $\{L_{r,1}, \dots, L_{r,K}\}$ restantes peuvent alors être considérées comme représentatives de la composante diffuse du matériau, uniquement.

En posant les hypothèses suivantes :

- la composante diffuse du matériau est parfaitement Lambertienne,
- les sources sont suffisamment ponctuelles pour que l'angle solide d'incidence soit ignoré,

– la luminance incidente est la même pour chaque source, l'équation 5.2 définissant la relation entre réflectance, éclairage et luminance lumineuse réfléchie permet d'aboutir à la relation suivante :

$$L_r = \rho_d L_i - \vec{n} \cdot \vec{l} \quad (5.13)$$

où :

- ρ_d représente la constante de réflexion Lambertienne,
- L_i est la luminance incidente de la source,
- \vec{l} est la direction d'incidence lumineuse,
- \vec{n} est la normale à la surface.

Dans cette équation, les seules données connues sont L_r et \vec{l} pour chaque échantillon. Les autres quantités peuvent être déterminées en résolvant le système linéaire suivant :

$$\begin{bmatrix} l_{1,x} & l_{1,y} & l_{1,z} \\ \vdots & \vdots & \vdots \\ l_{K,x} & l_{K,y} & l_{K,z} \end{bmatrix} \begin{bmatrix} -\rho_d L_i n_x \\ -\rho_d L_i n_y \\ -\rho_d L_i n_z \end{bmatrix} = \begin{bmatrix} L_{r,1} \\ \vdots \\ L_{r,K} \end{bmatrix} \quad (5.14)$$

On obtient alors deux informations à la fois :

- un coefficient relatif de réflectance Lambertienne (à un facteur près, dépendant de la luminance de la source), donné par la norme du vecteur résultat,
- un vecteur qui, une fois normalisé, correspond à la normale à la surface pour le pixel considéré.

Cette méthode permet donc de générer une véritable texture diffuse, mais également une texture de normales (figures 5.10 milieu et droite), ce qui est particulièrement intéressant pour simuler le relief de la surface lorsque l'information géométrique est représentée à un niveau d'échelle plus grossier que l'information de texture.

6 Bilan

Les travaux qui ont été présentés ici couvrent un large spectre du domaine de l'acquisition et du rendu de l'apparence d'objets réels, c'est pourquoi les différentes catégories de méthodes n'ont pas toutes été explorées de manière aussi exhaustive. Il faut également noter que l'ordre de présentation adopté ici respecte, à peu de choses près, une chronologie inverse, les travaux sur les textures bidirectionnelles les plus complexes étant, bien entendu, plus récents. Notre approche inverse a été de partir de la physique, et d'en faire découler les simplifications successives introduites pour rendre le problème plus abordable.

Nous avons volontairement vu large, dans cet état de l'art, afin que le lecteur puisse mieux situer nos travaux par rapport à l'ensemble de ce qui peut exister, parce que de nombreux éléments dans ces différents travaux nous seront utiles par la suite, mais également pour déterminer ce qu'il était réalisable d'entreprendre pour atteindre nos objectifs.

Contrairement à de nombreux projets de numérisation [12, 19, 98, 122], nous ne sommes pas intéressés par l'acquisition d'une texture diffuse. Notre objectif, en effet, est la restitution réaliste des copies numériques. Or, une représentation aussi simpliste ne permet pas de rendre compte de la véritable nature des matériaux composant l'objet.

Une texture bidirectionnelle permettrait, bien évidemment, de capturer l'information désirée, mais les raisons suivantes rendent le processus d'acquisition beaucoup trop complexe :

- il faut être capable de localiser une source lumineuse par rapport à la scène, mais les techniques présentées pour cela sont plutôt délicates à mettre en œuvre dans le cas concret, surtout si l'acquisition doit se dérouler dans un musée,
- pour capturer à la fois les variations lumineuses et les variations de points de vue, une grande quantité d'images doit être capturée, ce qui multiplie considérablement le temps d'acquisition.

L'acquisition d'une telle information ne semble finalement réalisable, dans l'état actuel des technologies, qu'en laboratoire. C'est d'ailleurs ainsi que nous l'avons mise en œuvre pour la numérisation de tableaux qui sera présentée au chapitre 9.

Les techniques de champs lumineux surfaciques, en revanche, constituent un bon compromis entre la simplicité d'acquisition et le réalisme visuel de la copie numérique. Même si l'éclairage est fixé une fois pour toute, une telle représentation peut permettre de restituer l'objet dans son environnement lumineux d'origine, comme par exemple celui du musée d'où il est issu, ce qui n'est pas totalement dénué de sens. De plus, il s'agit de la seule représentation ne nécessitant aucun contrôle de l'éclairage (bien qu'il faille s'assurer que le dispositif de numérisation ne perturbe pas l'environnement lumineux, ou très peu), et donc la seule ne demandant pas de résoudre le problème ardu de la localisation des sources. Ce type de représentations semble donc plus approprié pour une mesure *in situ*. C'est finalement ce vers quoi nous nous sommes dirigés pour le projet AMI3D, comme nous allons le voir dans le chapitre suivant.

Deuxième partie

— Travaux réalisés —

De nos quelques contributions

- Capture de champs lumineux -

Acquisition conjuguée de la forme et de l'apparence d'objets réels

Dans ce chapitre, nous présentons le protocole que nous avons développé dans le but d'acquérir la forme et l'apparence d'œuvres d'art, tout en satisfaisant les contraintes initialement fixées par le projet AMI3D, que nous rappelons ici :

- **Simplicité de la mesure** : la mesure doit pouvoir être effectuée par un opérateur non expert, c'est-à-dire que l'acte de prise de données ne doit nécessiter aucune connaissance technique particulière.
- **Mobilité du matériel** : étant donné qu'il s'agit de réaliser des mesures *in situ*, le matériel doit être particulièrement léger, c'est-à-dire peu encombrant d'une part, et ne nécessitant que peu de calibrage, voire pas du tout, d'autre part.
- **Automatisation** : la réalisation manuelle de certaines tâches, comme nous l'avons vu, peut rapidement devenir longue et fastidieuse. Nous souhaitons donc, autant que faire se peut, éviter à l'opérateur d'avoir à effectuer lui-même certains traitements.
- **Interactivité** : pouvoir contrôler le bon déroulement de la mesure est un point essentiel de la phase d'acquisition. Pour que ces vérifications puissent être effectuées en temps réel, et ainsi éviter les mauvaises surprises *a posteriori*, il est nécessaire que les traitements se fassent de manière suffisamment rapide.
- **Respect du patrimoine** : la numérisation d'œuvres d'art nécessite de prendre toutes les précautions possibles lors de la mesure, autant en ce qui concerne l'œuvre elle-même que son environnement. Cela impose des contraintes d'ordre technique, comme, par exemple, l'impossibilité de placer des mires physiques sur les objets à mesurer.

Compte tenu de toutes ces contraintes, nous nous sommes rapidement tournés vers l'acquisition de *champs lumineux surfaciques*, c'est-à-dire vers l'acquisition de la forme et de l'apparence d'objets plongés dans des environnements lumineux fixés. La raison principale de ce choix est, bien entendu, d'éviter d'avoir à se soucier, au moins dans un premier temps, du problème de localisation de sources lumineuses. Ce problème constitue, en effet, une difficulté certaine concernant au moins trois des contraintes que nous venons d'évoquer, à savoir la simplicité de la mesure, la mobilité du matériel et la possibilité d'un contrôle interactif. De plus, travailler en environnement lumineux fixe reste pertinent dans la mesure où il s'agit alors de capturer des objets dans le but de les restituer ultérieurement comme s'ils étaient placés dans leurs milieux d'origine.

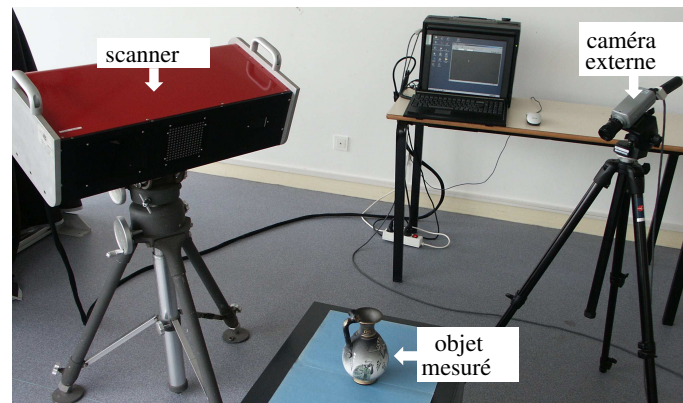


Figure 6.1 — Notre banc de numérisation. Seuls un scanner 3D à lumière structurée et une caméra numérique nous sont nécessaires à la capture de champs lumineux surfaciques.

La raison pour laquelle une représentation surfacique a été choisie plutôt qu'un modèle purement basé images vient simplement du fait que le point de départ du projet AMI3D est le dispositif de mesure de formes en lui-même. L'objectif est donc de fournir des modèles 3D dotés d'une information photométrique décrivant, d'une manière ou d'une autre, leur apparence.

Comme nous l'avons rappelé au cours des chapitres précédents, la problématique à laquelle nous avons à faire face est donc double :

- nous devons traiter le problème du recalage géométrique, de manière à aligner les différents morceaux de surface qui seront capturés au fur et à mesure du déroulement de la phase d'acquisition,
- nous devons localiser/recaler un ensemble de points de vue par rapport à la géométrie acquise, afin de permettre la mise en correspondance avec le modèle 3D de l'information de texture contenue dans un jeu de photographies.

Nous souhaitons donc effectuer ces deux tâches le plus rapidement possible, de manière à permettre le contrôle de la mesure à la volée, et ce à l'aide d'un protocole et d'un matériel simple.

À la fin de l'étape de numérisation, toutes les données nécessaires à un outil de visualisation de champs lumineux surfaciques doivent être obtenues, c'est-à-dire un modèle entièrement reconstruit et un ensemble d'images, échantillonnant la luminance, dont les points de vue sont connus.

Nous allons donc commencer, dans la section 1, par présenter le principe général de notre protocole d'acquisition. Nous décrivons ensuite, en section 2, comment nous réglons le problème de la détermination des correspondances nécessaires à la résolution des différents problèmes de recalages impliqués. Dans la section 3, nous voyons comment utiliser cette mise en correspondance pour élaborer notre protocole de numérisation de champs lumineux surfaciques, avant de parler de certains traitements indispensables à l'amélioration de la précision dans la section 4. Enfin, nous présentons quelques résultats (section 5), avant de conclure ce chapitre (section 6).

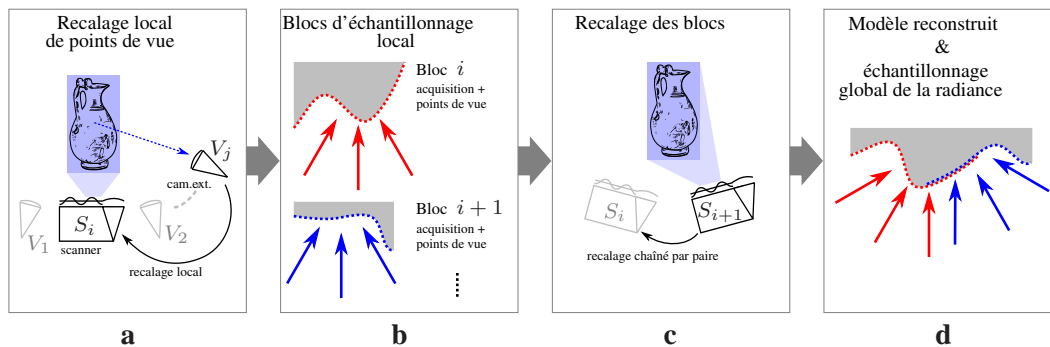


Figure 6.2 — Principe du protocole proposé pour l'acquisition de champs lumineux surfaciques. **a)** La luminance est échantillonnée localement par rapport à une unique image télémétrique. **b)** Cette opération est répétée de manière à obtenir différents *blocs d'échantillonnage local*. **c)** Ces blocs sont alors recalés entre eux par chaînage. **d)** Tous les morceaux de surface acquis, ainsi que tous les points de vue échantillonnant la luminance, sont finalement replacés dans un même repère global.

1 Principe

Nous proposons un protocole qui effectue le recalage géométrique et la détermination des points de vue de manière entièrement automatique et sans aucun contact ni déplacement de l'objet mesuré. De plus, par soucis de mobilité, nous n'utilisons pour ce faire qu'un matériel léger : seuls un scanner à lumière structurée et une caméra numérique sont nécessaires, comme on peut le voir sur la figure 6.1.

Notre protocole, résumé par la figure 6.2, fonctionne de la manière suivante :

1. **Échantillonnage local de points de vue** : nous proposons d'abord une procédure permettant de recalculer automatiquement un ensemble $\{V_j\}_{1 \leq j \leq N}$ de photographies prises par une caméra numérique sur la géométrie d'une image télémétrique venant d'être acquise depuis la position courante S_i du scanner 3D (voir figure 6.2-a). Cette procédure aboutit à ce que nous appellerons un *bloc d'échantillonnage local*, c'est-à-dire un bloc constitué de l'image télémétrique en question et de l'ensemble des photographies qui lui sont associées (qui sont localement recalées par rapport à cette image télémétrique).
2. **Acquisition chaînée** : la procédure précédente est alors répétée autant que nécessaire pour couvrir toute la surface de l'objet, générant ainsi plusieurs blocs isolés constitués d'une unique acquisition 3D et de l'ensemble des photographies qui lui sont associées (figure 6.2-b). Chaque bloc nouvellement acquis est alors recalé de manière chaînée par rapport au précédent (figure 6.2-c). Ce recalage est, une fois de plus, automatique, et la transformation rigide obtenue est appliquée non seulement à l'image télémétrique, mais également à son ensemble de photographies, permettant ainsi de replacer aussi bien les morceaux de surface que les points de vue nécessaires à l'échantillonnage de la luminance dans un même repère global (figure 6.2-d).

3. **Post-traitements** : nous proposons deux types de traitements pour parfaire le résultat de notre protocole d'acquisition. Le premier est un raffinement des transformations de recalage géométrique calculées lors de l'acquisition chaînée. Ce raffinement est effectué à l'aide d'un algorithme de type ICP, et se déroule durant la mesure elle-même, afin de permettre un bon contrôle par l'opérateur. Le second traitement est une passe de recalage globale. Celle-ci étant plus coûteuse en termes de temps de calcul, et nécessitant de connaître toutes les acquisitions à la fois, elle est effectuée *a posteriori*, une fois que la mesure est totalement achevée.

Les deux types de recalages impliqués (recalage géométrique et recalage de points de vue) consistent à calculer une transformation entre deux jeux de données à partir d'une liste de correspondances qu'il faut déterminer. La contribution de notre protocole d'acquisition est donc de résoudre ce problème de mise en correspondance, comme nous allons le voir dans la section suivante.

2 Détermination des correspondances

Dans cette section, nous présentons notre solution au problème de la détermination des correspondances entre plusieurs jeux de données. Pour cela, nous utilisons le modèle de lumière structurée par franges que nous avons présenté dans le chapitre 2. Ce modèle, initialement exploité par notre scanner pour effectuer la mesure de formes, a été détourné pour l'occasion afin de permettre de projeter une paramétrisation 2D sur la surface à mesurer. Cette paramétrisation permet alors d'identifier de manière unique chaque point qu'elle couvre à l'aide d'un couple de coordonnées. La recherche de correspondances se résume alors à trouver, dans les différents jeux de données, les points dont les coordonnées de paramétrisation sont identiques. Toute mesure étant, par définition, sujette à l'incertitude, certaines de ces correspondances peuvent être erronées. Par soucis de robustesse algorithmique, il est capital d'estimer la précision des données acquises afin d'éviter aux traitements ultérieurs de souffrir des divergences occasionnées. Dans ce but, nous introduisons une métrique d'erreur basée sur le modèle de paramétrisation proposé afin de détecter et d'éliminer de manière simple et efficace les faux-positifs éventuels, c'est-à-dire les appariements de points qui peuvent ne pas représenter des correspondances valides.

■ 2.1 Paramétrisation de surfaces réelles par la phase

Rappelons brièvement le principe du modèle de lumière structurée utilisé par notre scanner : un ensemble de franges verticales dont la variation d'intensité est décrite par une courbe sinusoïdale est projeté sur la surface à mesurer à l'aide d'un vidéoprojecteur, comme on peut le voir sur la figure 6.3. Chaque frange correspond à une colonne de pixels du vidéoprojecteur. Cette sinusoïde est projetée à plusieurs reprises, en introduisant à chaque fois un décalage de phase connu. Chacune de ces sinusoïdes projetées est capturée à l'aide d'une



Figure 6.3 — Franges lumineuses projetées par le scanner pour la mesure de formes. La variation d'intensité des franges est décrite par une courbe sinusoïdale en niveaux de gris, permettant d'identifier chaque frange de manière unique grâce à un calcul de la phase de cette sinusoïde en chaque pixel de la caméra d'acquisition.

caméra numérique, et les différentes images obtenues permettent alors de calculer une phase de sinusoïde en chaque pixel de cette caméra. La valeur de phase identifie chaque frange de manière unique, et permet donc de retrouver à quelle colonne de pixels du vidéoprojecteur se réfère l'un des points observés par la caméra. En connaissant précisément les positions relatives de la caméra numérique et du vidéoprojecteur, l'information de profondeur peut alors être déterminée par triangulation optique, en cherchant l'intersection du rayon traversant le pixel considéré de la caméra avec le plan défini par la frange observée (*cf.* chapitre 2, section 2).

Dans ce processus de mesure de formes, ce qui nous intéresse particulièrement pour résoudre notre problème de mise en correspondance est l'information de phase. Celle-ci induit une paramétrisation 1D de la surface selon une orientation orthogonale à celle des franges, du fait que la phase soit définie strictement croissante sur le domaine de projection. Ainsi, l'ensemble de points correspondant à l'intersection d'une frange donnée avec la surface se verra attribuer la même valeur de phase.

Cette paramétrisation 1D peut donc, bien entendu, être aisément étendue à une paramétrisation 2D. En projetant les franges lumineuses selon deux orientations différentes, comme cela est montré par la figure 6.4, un couple $\Phi(p) = (\phi_u(p), \phi_v(p))$ de valeurs de phase peut être défini en chaque point p de la surface. Comme les deux fonctions de phase $\phi_u(p)$ et $\phi_v(p)$ sont strictement croissantes sur leurs domaines respectifs, le couple $\Phi(p)$ représente un identifiant unique pour le point p . Dans la suite de ce document, nous appellerons cette paramétrisation 2D la *paramétrisation* Φ .

Comme nous l'avons rappelé un peu plus haut, le calcul de phase pour un point donné de la surface est effectué à partir des différentes intensités lumineuses perçues pour chaque déphasage introduit dans la fonction sinusoïde. Or, ces intensités sont, bien évidemment, dépendantes de la position de la caméra d'acquisition, dans la mesure où la quantité d'énergie lumineuse de la frange incidente qui est réfléchiée vers cette caméra dépend de la fonction de réflectance de la surface au point mesuré. Cependant, puisque ni le projecteur ni la ca-

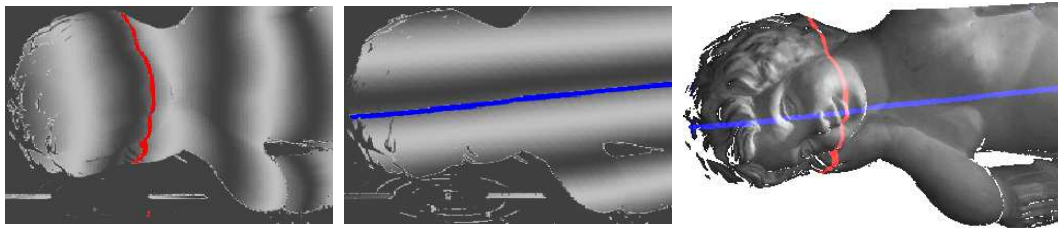


Figure 6.4 — Paramétrisation 2D par la phase. **Gauche** : image de phase calculée à partir de la projection de franges verticales. **Milieu** : image de phase calculée à partir de la projection de franges horizontales. **Droite** : chaque point de la surface est identifié de manière unique grâce aux valeurs de phases des deux franges qui s'y intersectent.

méra ne bougent par rapport à l'objet durant toute l'acquisition, les directions d'incidence lumineuse et d'observation restent fixes, et la valeur de la fonction de réflectance en chaque point est donc constante pour l'ensemble des mesures associées aux différents déphasages. Cela signifie que les variations d'intensité perçues en un même point de la surface depuis différents points de vue sont identiques à un facteur près. Ce constat implique que la sinusoïde observée pour n'importe quel point de vue ne diffère de la sinusoïde projetée que par son amplitude. **Le calcul de phase est donc indépendant du point de vue d'acquisition.**

Ainsi, tant que le projecteur reste fixe par rapport à l'objet d'intérêt, les valeurs de phases calculées en un même point de la surface pour différents points de vue sont identiques. Nous dirons donc, par la suite, que deux paramétrisations Φ sont identiques si elles ont été capturées selon la même position relative du projecteur par rapport à l'objet.

En considérant les propriétés qui viennent d'être énoncées, tant que le scanner et l'objet mesuré restent fixes l'un par rapport à l'autre, la paramétrisation Φ demeure la même et est totalement indépendante du point de vue choisi pour la capturer. En conséquence, tant que la paramétrisation ne change pas, deux pixels p et q pris dans deux points de vue différents font nécessairement référence au même point physique si $\Phi(p) = \Phi(q)$.

■ 2.2 Sélection des correspondances

Nous souhaitons exploiter la paramétrisation décrite précédemment dans le but de trouver les correspondances entre différents jeux de données. Cependant, l'égalité entre deux couples de phases est rarement possible dans le cas concret : étant donné qu'il s'agit d'une mesure, les sources d'erreur sont nombreuses. La plus incontournable de toute est certainement la discrétisation CCD des dispositifs numériques impliqués. L'incapacité de ces dispositifs à capturer des domaines continus pose de gros problèmes. En effet, chaque cellule CCD d'une caméra numérique n'observe pas un point unique mais un petit élément de surface correspondant à l'aire de la cellule projetée sur l'objet depuis le centre optique. Le signal renvoyé par la cellule est donc le résultat d'une intégration de l'information sur l'ensemble de cet élément de surface.

Cette intégration se manifeste par des effets néfastes de moyennage, plus particulièrement

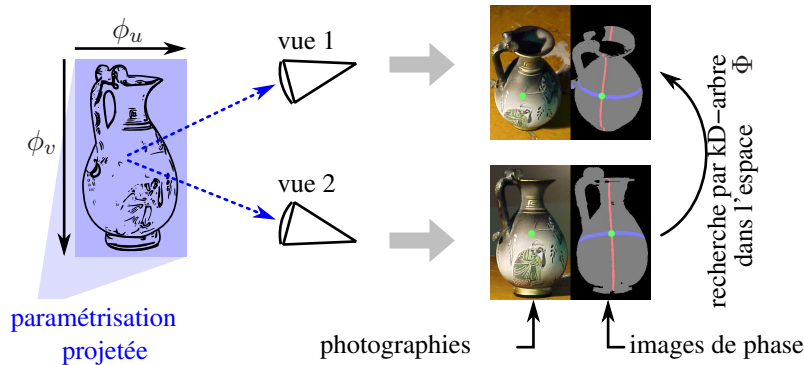


Figure 6.5 — Recherche de correspondances entre deux points de vue acquis à l'aide de la même paramétrisation Φ . Chaque pixel du premier point de vue est associé au pixel du second point de vue dont les coordonnées de phases sont les plus proches.

apparents au niveau des fortes discontinuités (saut de l'information de profondeur par rapport au point de vue), de la silhouette de l'objet, et dans les zones où la surface devient très rasante par rapport au point de vue. Dans le cas des discontinuités, l'intégration CCD a pour effet de lisser les angulosités. Pour ce qui est de la silhouette, l'aire projetée de la cellule CCD n'atteint pas uniquement la surface mesurée, mais également ce qui se trouve au-delà, qui peut être soit une autre partie de l'objet, soit un élément hors champ du décor. L'effet de moyennage tend alors à attirer la surface vers cet élément d'arrière-plan, et réciproquement. Pour ce qui est des surfaces rasantes, l'aire projetée de la cellule CCD sur la surface devient très importante et, par conséquent, l'effet de moyennage également. Si les artefacts engendrés sont souvent bien moins flagrants, au niveau visuel, que dans les cas précédents, ils n'en demeurent pas moins sources d'une grande imprécision.

Toutes ces raisons impliquent que l'égalité entre les couples de phases de deux pixels étant censés représenter le même point physique ne peut être assurée. La recherche de correspondances n'est donc plus un test d'égalité mais une recherche de plus proches voisins dans l'espace de la paramétrisation Φ .

Considérons donc deux pixels p et q quelconques, pris dans différents points de vue mais capturés avec la même paramétrisation Φ . La dissimilarité entre p et q , que nous notons $\delta_\Phi(p, q)$, est alors définie par la distance Euclidienne au carré entre leurs coordonnées de phases respectives $\Phi(p) = (\phi_u(p), \phi_v(p))$ et $\Phi(q) = (\phi_u(q), \phi_v(q))$:

$$\delta_\Phi(p, q) = \|\Phi(p) - \Phi(q)\|_2^2 = (\phi_u(p) - \phi_u(q))^2 + (\phi_v(p) - \phi_v(q))^2 \quad (6.1)$$

Ainsi, si des correspondances doivent être trouvées entre deux jeux de données identifiés par la même paramétrisation Φ , la recherche consiste à trouver, dans le second jeu, les éléments qui sont les plus proches de ceux du premier jeu de données par rapport à la distance δ_Φ de l'équation 6.1.

Pour effectuer cette recherche, nous avons employé des k D-arbres. Comme il s'agit d'explorer l'espace de la paramétrisation Φ , des arbres de dimension 2 suffisent. Pour la construction des arbres, nous avons adopté l'approche de Friedman *et al.* [50] qui consiste à sub-

diviser le sous-ensemble de données contenu en chaque nœud selon sa dimension de plus grande dispersion.

Il faut noter que si les bornes du domaine $[\phi_{u,min}, \phi_{u,max}] \times [\phi_{v,min}, \phi_{v,max}]$ correspondant à la plage de valeurs couverte par la phase de la sinusoïde projetée sont connues, de même que la résolution du vidéoprojecteur utilisé, une méthode de recherche plus immédiate peut être utilisée : en effet, chaque couple de coordonnées peut être assigné à l'un des pixels du projecteur. Ainsi, les éléments dont les coordonnées de phases font référence au même pixel projecteur peuvent être appareillés, ce qui aboutit à une recherche en temps constant. Cependant, ce mécanisme gère assez mal le cas où la dissimilarité entre les éléments devant être appareillés devient importante. Il faut dans ce cas considérer un voisinage de pixels, ou employer une méthode de type Elias [31] dans le plan image du projecteur. Ne disposant pas des informations matérielles nécessaires à la mise en œuvre d'une telle approche, nous sommes finalement restés sur la solution des kD -arbres.

■ 2.3 Classification des correspondances

À l'issue de l'étape de recherche de plus proches voisins, toutes les paires d'éléments résultantes peuvent ne pas représenter des correspondances valides. En effet, même si la recherche par kD -arbre fournit un résultat, l'élément le plus proche qui a été trouvé n'est pas nécessairement une bonne correspondance. On peut prendre l'exemple de deux points de vue différents et d'un point sur la surface de l'objet n'étant visible que depuis le premier d'entre eux. Le pixel du premier point de vue associé à ce point aura nécessairement un plus proche voisin dans le second point de vue par rapport à ses coordonnées de phases. Cependant, le pixel trouvé fera nécessairement référence à un autre point de la surface, puisque le vrai point physique n'est pas visible de ce second point de vue.

Ainsi, pour déterminer la validité de chaque appariement, nous utilisons la mesure de dissimilarité δ_{Φ} : si la distance δ_{Φ} entre les deux éléments appareillés dépasse un certain seuil S_{δ} , ceux-ci sont considérés comme trop éloignés l'un de l'autre dans le domaine de la paramétrisation pour définir une correspondance valide. La paire d'éléments incriminée est alors éliminée. Ce seuillage, en plus de supprimer les faux-positifs, est un bon moyen de ne retenir que les correspondances les plus précises en choisissant un seuil suffisamment bas. Nous discuterons de la valeur de ce seuil dans la section résultat.

3 Description du protocole d'acquisition

Afin d'exploiter le principe d'identification précédemment énoncé, le scanner 3D que nous utilisons a dû être modifié par ses concepteurs sur notre demande de manière à répondre à trois besoins spécifiques :

- tout d’abord, il nous fallait pouvoir récupérer une carte de phase avant que celle-ci ne soit convertie en carte géométrique (image télémétrique).
- ensuite, nous voulions pouvoir projeter les franges dans une direction arbitraire, et non pas uniquement celle initialement prévue pour la mesure de forme. Ainsi, pour capturer la paramétrisation Φ , nous projetons deux séries de franges : l’une verticale et l’autre horizontale. En théorie, il suffit juste que les orientations des deux séries de franges soient différentes pour générer la paramétrisation Φ . Dans la pratique, il est préférable que ces orientations soient les plus opposées possible afin de limiter les imprécisions numériques lors des traitements ultérieurs (eg. recherche par kD -arbres).
- enfin, nous souhaitions pouvoir acquérir la phase de la sinusoïde projetée également à partir d’une caméra autre que celle embarquée dans le scanner. Comme nous allons le voir, cette dernière modification nous est indispensable pour la capture de l’apparence. Nous parlerons de *caméra embarquée* pour désigner la caméra incluse dans le scanner, et de *caméra externe* lorsque nous ferons référence à une seconde caméra numérique, indépendante, cette fois-ci, de l’outil de mesure de formes.

Notre protocole se décompose alors en deux parties. Tout d’abord, la luminance réfléchie par l’objet d’intérêt est échantillonnée localement par rapport au dernier morceau de surface acquis, à l’aide d’un ensemble de clichés photographiques capturés depuis des points de vue proches de la position actuelle du scanner. Ainsi, chaque acquisition forme, avec l’ensemble des points de vue qui lui sont associés, un *bloc d’échantillonnage local*. Ces blocs sont alors recalés entre eux par une procédure chaînée afin de replacer toutes les informations, géométriques et photométriques, dans un même repère.

■ 3.1 Échantillonnage local de l’apparence

Plaçons nous dans le cas où une image télémétrique R_i vient d’être acquise. Nous demandons alors au scanner de projeter les deux séries de franges, verticales et horizontales, de manière à pouvoir capturer la paramétrisation Φ à l’aide de la caméra embarquée. Si le scanner n’a pas bougé entre l’acquisition géométrique et l’acquisition de phase, il existe une correspondance pixel à pixel entre l’image télémétrique et l’image de phase, du fait qu’elles aient toutes deux été acquises depuis le même point de vue. Un identifiant de phase $\Phi(p)$ peut donc être assigné à chaque point p de l’image télémétrique.

La luminance relative à cette image télémétrique peut alors être acquise très simplement, comme l’illustre la figure 6.6. Préalablement, un kD -arbre est construit, permettant de trouver le point 3D de l’image télémétrique ayant les coordonnées de phases les plus proches d’un couple de coordonnées requête. Cette arbre va alors servir à établir les correspondances 2D/3D entre les photographies et la géométrie.

La luminance est alors échantillonnée à l’aide d’un certain nombre de clichés photographiques $\{V_1, \dots, V_N\}$ pris par une caméra externe. Pour chaque point de vue, en plus de la photographie V_j est également capturée la paramétrisation Φ projetée par le scanner, qui n’a toujours pas été déplacé par rapport à l’objet (la paramétrisation Φ est donc restée la même). Là encore, il existe une correspondance pixel à pixel entre la photographie et

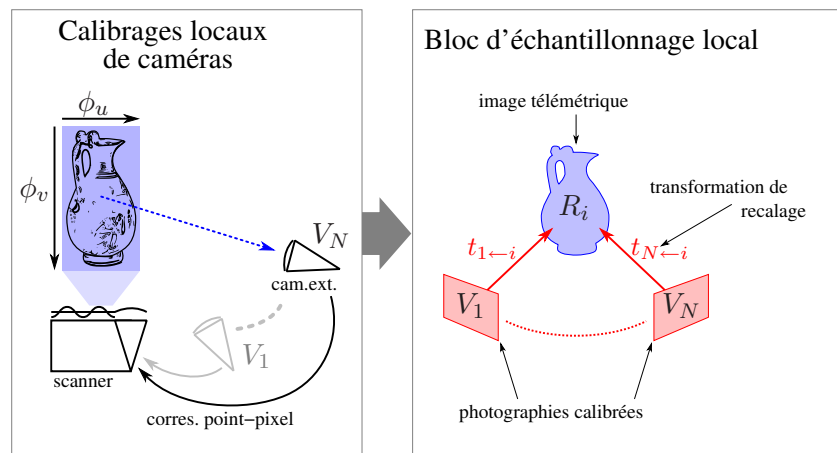


Figure 6.6 — Échantillonnage local de la luminance réfléchie par l'objet. Les points de vue utilisés pour l'échantillonnage sont recalés par rapport à la dernière image télémétrique acquise.

l'image de phase, ce qui permet d'identifier chaque pixel de la photographie à l'aide d'un couple de coordonnées de phases. Des correspondances peuvent alors être établies entre les pixels de V_j et les points 3D de R_i en utilisant le k D-arbre précédemment construit comme cela a été présenté dans la section 2.2.

Recaler la photographie V_j sur l'image télémétrique R_i revient alors à déterminer la transformation caméra $t_{j \leftarrow i}$ permettant de projeter n'importe quel point de R_i sur le plan image de V_j . Cette transformation peut finalement être calculée à partir des correspondances précédentes par un algorithme de calibrage caméra (figure 6.6-droite). Nous avons utilisé, pour cela, l'algorithme de Tsai [187]. L'image télémétrique R_i et l'ensemble des photographies $\{V_j\}_{1 \leq j \leq N}$ qui ont été calibrées par rapport à celle-ci forment alors un bloc d'échantillonnage local. Nous noterons dorénavant V_j^i la j -ième photographie associée à la i -ième image télémétrique. De plus, nous emploierons sans distinction les expressions *calibrage de points de vue*, *recalage de points de vue* ou *localisation de points de vue* pour décrire le processus qui vient d'être présenté.

Dans la pratique, tous les points 3D ou tous les pixels d'une photographie ne reçoivent pas forcément un identifiant de phase. En effet, il se peut que l'une ou l'autre des coordonnées de phases (voire les deux) n'ait pu être calculée pour diverses raisons. Les points ou les pixels concernés sont alors simplement ignorés lors des traitements.

- On peut noter, à propos du calibrage caméra, que deux cas de figure peuvent se présenter :
- **la mesure est effectuée avec une optique fixe**, c'est-à-dire que seule la position de la caméra est modifiée (déplacement), mais pas son objectif. Dans ce cas, seuls les paramètres extrinsèques changent entre chaque point de vue. Ainsi, les paramètres intrinsèques n'ont besoin d'être calibrés qu'une seule fois, et peuvent ensuite être réutilisés tout au long de la mesure.
 - **la mesure est effectuée avec une optique variable** : ce cas de figure peut se produire, par exemple, si l'on souhaite acquérir des points de vue à différentes distances par rap-

port à l'objet. Dans ce cas, la mise au point doit être effectuée, modifiant en conséquence les paramètres intrinsèques pour chaque nouvelle prise de vue. Un calibrage complet est donc nécessaire pour chaque image.

Un calibrage complet est bien entendu plus coûteux qu'un simple calibrage des paramètres extrinsèques. Pour des raisons de performance, nous avons choisi de travailler en optique fixe, ce qui impose à la caméra externe d'être toujours placée approximativement à la même distance de l'objet. Dans la pratique, cette obligation n'est en fait pas si contraignante. Ainsi, un seul calibrage complet est nécessaire. Celui-ci peut être effectué soit à partir des données relatives au tout premier point de vue acquis, soit par une étape de précalibrage, ce qui est généralement préférable par soucis de précision.

Il va de soi qu'il reste tout à fait possible de travailler en optique variable : il suffit alors de demander un calibrage complet pour chaque acquisition de point de vue au lieu d'un simple calibrage des paramètres extrinsèques, au prix de temps de traitements plus longs.

■ 3.2 Recalage géométrique chaîné par bloc

Chaque bloc d'échantillonnage local ne représente qu'une sous-partie du champ lumineux surfacique final. Tous ces blocs doivent donc être fusionnés par un processus de recalage géométrique. Comme nous l'avons déjà dit, ce processus ne se contente pas simplement d'aligner les différents morceaux de surface entre eux : les transformations rigides, calculées à partir des données géométriques, sont appliquées à l'ensemble des blocs d'échantillonnage local, c'est-à-dire autant aux images télémétriques qu'aux jeux de photographies calibrées qui les accompagnent.

À chaque fois qu'un nouveau bloc doit être acquis, le scanner est déplacé vers une nouvelle position. Le problème posé par le recalage géométrique est alors d'établir un référentiel commun entre cette nouvelle position et l'ancienne. Pour résoudre ce problème, nous proposons d'utiliser la caméra externe comme repère fixe entre deux acquisitions successives.

Comme le montre la figure 6.7, cette caméra va capturer les deux paramétrisations respectivement projetées depuis l'ancienne position S_i , puis la nouvelle position S_{i+1} du scanner. Sur le même principe que le recalage de points de vue présenté à la section précédente, ces deux paramétrisations Φ permettent d'établir des correspondances 3D/2D entre les pixels de la caméra externe et les points 3D de chacune des deux images télémétriques R_i et R_{i+1} . Les pixels de la caméra externe qui sont liés aux deux images télémétriques à la fois nous fournissent finalement les correspondances géométriques nécessaires au recalage des blocs. Nous utilisons alors la méthode du quaternion de Horn [78] pour calculer la transformation rigide $T_{i \leftarrow i+1}$, plaçant l'image télémétrique R_{i+1} dans le repère de R_i , à partir de ces correspondances. Une fois cette transformation connue, nous l'appliquons à l'ensemble du bloc. Ainsi, chaque bloc est recalé par rapport à son prédécesseur par une procédure chaînée, et les transformations sont alors toutes définies localement, de proche en proche.

Bien entendu, les transformations permettant de passer directement des différents repères

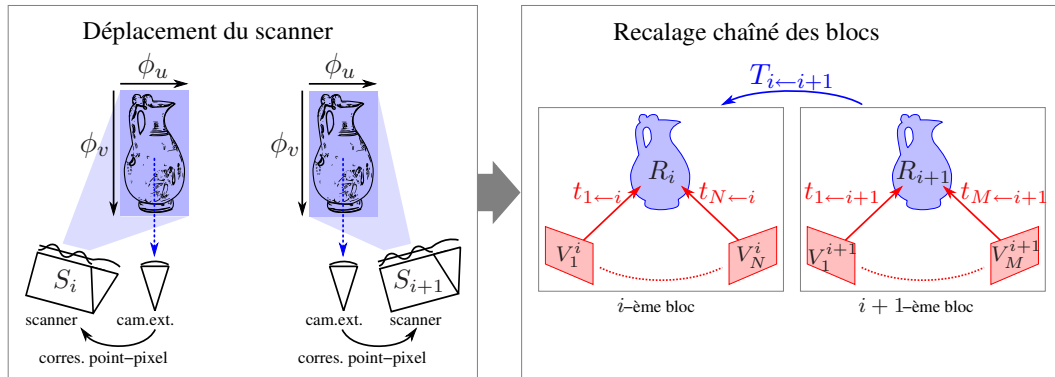


Figure 6.7 – Recalage chaîné des blocs d'échantillonnage local. La caméra externe sert de point fixe pour établir des correspondances entre deux positions successives du scanner à l'aide de la paramétrisation Φ .

locaux au repère global peuvent être retrouvées de manière immédiate. Ainsi, la transformation T_{global} permettant de replacer n'importe quelle image télémétrique R_i dans le repère global est donnée par :

$$T_{global}(R_i) = T_{1 \leftarrow 2} \circ \dots \circ T_{i-1 \leftarrow i} \quad (6.2)$$

et la transformation t_{global} permettant de passer du repère global à l'espace image de n'importe quelle photographie V_j^i est donnée par :

$$t_{global}(V_j^i) = t_{j \leftarrow i} \circ T_{global}(R_i)^{-1} \quad (6.3)$$

La méthode de recalage chaîné que nous venons de présenter a bien entendu une solution plus immédiate lorsque deux scanners à projection de franges sont disponibles : les scanners sont déplacés à tour de rôle, et celui qui vient d'être bougé capture alors la paramétrisation Φ projetée par celui resté immobile (figure 6.8). Des correspondances directes entre les points 3D des deux images télémétriques peuvent alors être déterminées, rendant obsolète l'usage de la caméra intermédiaire comme référentiel commun.

À ce propos, il est clair que le fait d'introduire un dispositif intermédiaire plutôt que d'utiliser deux scanners risque d'engendrer une perte de précision. Ce point sera abordé dans les résultats de ce chapitre, en section 5.2. Néanmoins, pour limiter la complexité et le coût matériel, nous avons préféré opter pour la solution n'utilisant qu'un scanner et la caméra externe, celle-ci s'étant révélée suffisamment précise pour être exploitée. Tous les exemples fournis dans ce document ont d'ailleurs été produits par cette méthode.

■ 3.3 Amélioration de la précision

Lors de l'établissement des correspondances entre la caméra externe et une image télémétrique, il est important de remarquer que le seuillage sur la distance δ_Φ que nous avons

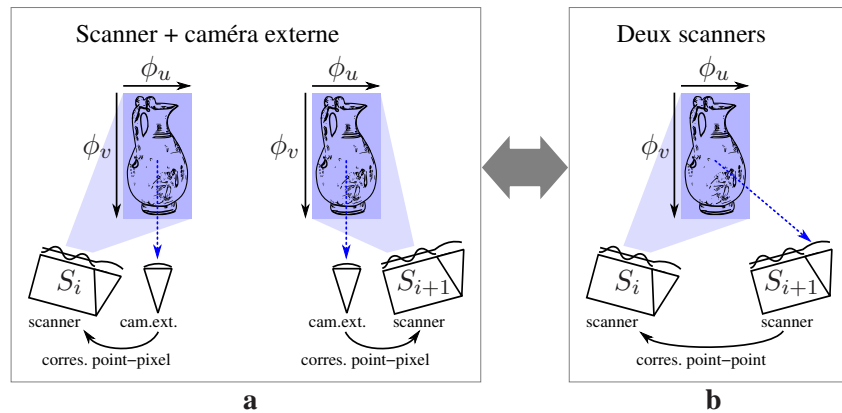


Figure 6.8 – Les deux approches matérielles possibles pour le recalage des blocs. a) Un seul scanner est utilisé, et la caméra externe sert de référentiel commun entre deux acquisitions successives. b) Deux scanners sont utilisés, et sont déplacés à tour de rôle. La paramétrisation projetée par le premier est capturée par le second, et les correspondances géométriques entre les deux positions sont immédiatement trouvées.

introduit précédemment ne suffit pas à éliminer tous les mauvais appariements. Considérons l'exemple de la figure 6.9, représentant l'étape d'identification d'une surface à l'aide de la paramétrisation Φ . On peut voir que les points p_1 , p_2 et p_3 reçoivent des identifiants de phases correspondant à trois franges successives en espace projecteur (trois pixels adjacents dans le plan image du projecteur). Les points p_1 et p_3 ont donc des coordonnées de phases très proches de celles de p_2 .

Admettons maintenant que la caméra externe acquière cette même paramétrisation Φ depuis un point de vue donné, et effectue une recherche de correspondances parmi les points de la géométrie acquise, de la façon que nous avons proposée. Les imprécisions numériques lors du calcul de phase peuvent très bien faire qu'au lieu de trouver le point p_2 comme correspondance pour un couple de coordonnées de phases donné nous trouvions soit le point p_1 , soit le point p_3 . Si c'est le point p_1 qui est trouvé à la place du point p_2 , l'erreur n'est pas franchement préjudiciable dans la mesure où ces deux points sont relativement proches, géométriquement parlant. En revanche, si c'est p_3 qui est trouvé à la place de p_2 , cela devient beaucoup plus problématique puisque la proximité dans la paramétrisation Φ n'est pas du tout corrélée, dans ce cas, à la proximité géométrique. Les différents calculs de transformations qui s'ensuivent risquent alors d'être faussés.

Ce problème va en fait se poser à chaque fois qu'un saut de profondeur sera observable sur la surface du point de vue du projecteur. Heureusement pour nous, ces sauts de profondeur peuvent facilement être détectés dans l'image télémétrique : si deux points 3D situés dans des pixels adjacents de cette image se trouvent à une distance très importante l'un de l'autre, il y a de grandes chances pour qu'il ne s'agisse en fait pas de points voisins sur la surface, auquel cas ces deux points seront marqués comme étant à proximité d'un saut de profondeur. Ainsi, nous pouvons résoudre notre problème d'imprécision en éliminant, dans la liste des correspondances trouvées, celles qui contiennent des points 3D ainsi marqués.

Une autre source d'imprécision peut survenir lors de l'étape de calibrage caméra : un point

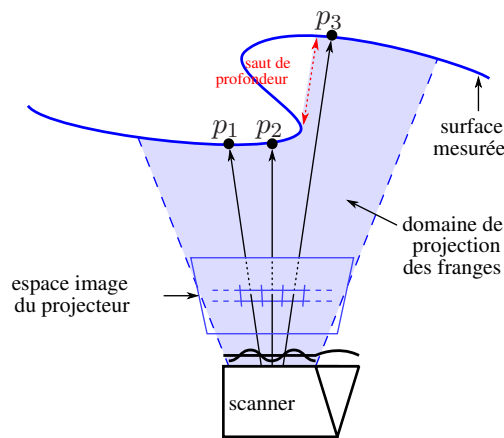


Figure 6.9 – Les points situés de part et d'autre d'un saut de profondeur par rapport au point de vue du projecteur auront des coordonnées de phases très proches (car issues de franges adjacentes dans l'espace image du projecteur) bien qu'ils soient très éloignés l'un de l'autre sur la surface. Ces points sont donc sources d'imprécision lors de la détermination de correspondances par la phase.

essentiel est de s'assurer, pour cette opération, une bonne répartition des correspondances 3D/2D dans l'espace image de la photographie à calibrer. Le calibrage est, en effet, un processus de minimisation, qui estime les paramètres caméra de manière à tenir compte au mieux des correspondances fournies. Il est donc forcément sensible aux données en entrée : si les correspondances sont toutes groupées dans une petite partie de l'image, le calibrage, considéré à un niveau global, ne sera pas aussi précis que si ces correspondances étaient réparties de manière plus homogène.

Nous avons développé une méthode permettant de palier à ce second problème. Tout d'abord, nous fixons pour la distance δ_Φ un seuil suffisamment élevé pour que le nombre de correspondances récupérées soit important. Celles qui sont associées à des points situés près des sauts de profondeur sont éliminées, comme nous venons de l'expliquer. Pour celles qui restent, la boîte englobante qui les contient dans l'espace image est calculée, puis placée dans une liste.

L'algorithme de sélection des correspondances est alors le suivant :

- la boîte dont l'aire est la plus grande, en nombre de pixels, est sortie de la liste puis découpée, au milieu de sa dimension la plus grande, en deux nouvelles boîtes.
- chacune des ces deux boîtes est réajustée de manière à coller aux données qu'elle contient, puis insérée à son tour dans la liste.
- cette opération de découpage est répétée jusqu'à ce qu'un nombre donné de boîtes ait été atteint.

L'ensemble final de correspondances est alors obtenu en prenant, dans chaque boîte, la correspondance la plus précise, c'est-à-dire celle pour laquelle la distance δ_Φ entre ses deux éléments est la plus petite.

Cet algorithme présente deux avantages : il aboutit à une meilleure répartition des cor-

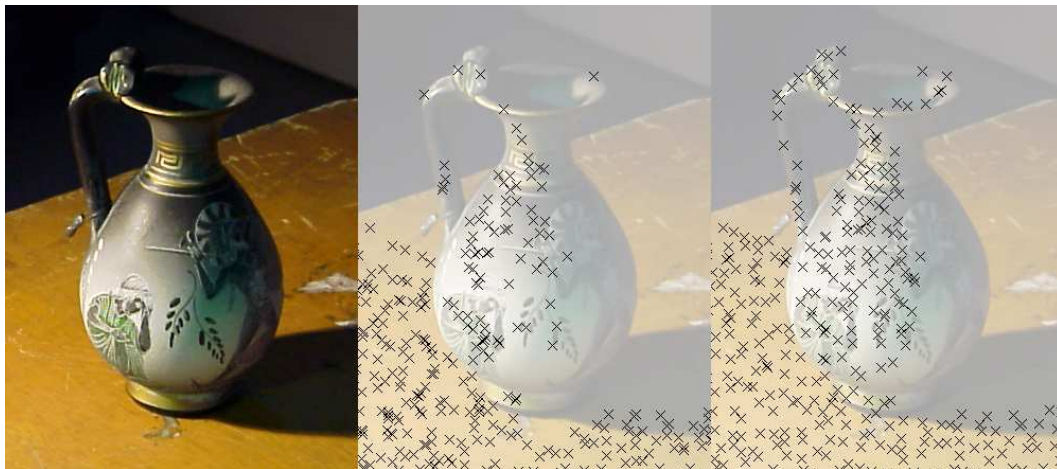


Figure 6.10 – Sélection améliorée des correspondances pour le calibrage caméra. La répartition en espace image est plus homogène (**droite**) par rapport à la simple sélection par phase (**milieu**).

respondances dans le plan image, comme on peut le voir sur la figure 6.10, tout en tenant compte du critère de précision δ_Φ , et il permet également de contrôler exactement le nombre de correspondances qui vont être fournies à l’algorithme de calibrage caméra. Ce dernier point a, bien entendu, un impact positif sur les performances, dans la mesure où le calibrage s’effectue généralement plus vite lorsque le nombre de correspondances à traiter est plus faible.

4 Post-traitements

Le recalage par phase que nous venons de décrire fonctionne donc par chaînage, en recalant chaque bloc par rapport au précédent, dans l’ordre de leur acquisition. À ce titre, notre méthode est sujette à une accumulation de l’erreur au cours de la mesure, comme cela a été expliqué dans la section 4 du chapitre 3. Nous allons donc maintenant proposer deux solutions, complémentaires l’une de l’autre, pour palier à ce problème.

La première de ces solutions, traitée dans la section 4.1, consiste à effectuer un raffinement des recalages par paires *à la volée*, c’est-à-dire durant la mesure elle-même. L’idée est, en fait, de profiter des temps morts qui surviennent lorsque l’opérateur est occupé à déplacer le scanner 3D ou la caméra externe. Ainsi, le recalage par phase fournit un premier retour visuel à l’opérateur, qui peut ainsi juger immédiatement du bon déroulement des opérations. Alors qu’il retourne à ses instruments pour continuer la mesure, le logiciel affine ce premier résultat à l’aide d’un algorithme de minimisation tournant en parallèle. Bien entendu, cette approche n’élimine pas définitivement le problème de l’accumulation de l’erreur : il permet simplement d’en limiter l’impact, au moins pour le temps de la mesure, où il est important

de pouvoir correctement estimer la validité des résultats intermédiaires.

La seconde solution est l'implantation d'un algorithme de recalage global. Étant donné que ce genre d'algorithme est assez coûteux en temps de calcul et qu'il nécessite de connaître la totalité des données géométriques à la fois, cette étape globale est effectuée *a posteriori*, une fois la mesure totalement achevée. Ce point sera détaillé dans la section 4.2.

■ 4.1 Raffinement à la volée du recalage chaîné

Lorsqu'une nouvelle image télémétrique vient d'être acquise, le recalage par phase est d'abord exécuté de manière à obtenir une première transformation. Cette transformation, déjà très précise, permet de vérifier immédiatement que le morceau de surface nouvellement acquis couvre bien la zone désirée. Si c'est le cas, les photographies constituant, avec cette acquisition, le bloc d'échantillonnage local courant peuvent être acquises. Sinon, la mesure peut être refaite dans la foulée. Quoi qu'il en soit, l'opérateur a de toute façon du travail devant lui, ce qui va générer des temps d'inactivité pour notre logiciel de traitements.

Nous proposons donc de mettre à profit ces temps morts pour raffiner les transformations issues du recalage par phase. Pour ce faire, nous utilisons un algorithme de type ICP¹ en prenant, comme conditions initiales, les transformations rigides déjà obtenues. Étant donné que ce raffinement doit se dérouler pendant la mesure, il est préférable que sa vitesse d'exécution soit la plus rapide possible, de manière à s'achever avant que l'opérateur ne revienne contrôler l'acquisition suivante. Dans le but d'obtenir une convergence rapide et précise, nous avons conçu notre algorithme de minimisation en nous basant sur l'étude de Rusinkiewicz *et al.* [158]. Dans cette étude, plusieurs pistes sont proposées pour améliorer la convergence de tels algorithmes, autant en termes de vitesse que de précision :

- A— l'étape de sélection des correspondances consiste à rechercher, dans le jeu de données destination, le plus proche voisin d'un point du jeu de données source. Les auteurs recommandent un échantillonnage bilatéral, c'est-à-dire que chacun des deux jeux de données est à la fois source et destination. Ainsi, pour certains points de la première surface, les plus proches voisins sont recherchés dans la seconde, et *vice versa*.
- B— lorsqu'une information sur la fiabilité de chaque correspondance est disponible, celle-ci peut être utilisée comme pondération lors du calcul de la transformation rigide afin de rendre ce calcul plus précis. Ainsi, à chaque couple (p, q) de points appareillés, nous assignons le poids suivant :

$$w(p, q) = \max(0, \vec{n}_p \cdot \vec{n}_q) \quad (6.4)$$

où \vec{n}_p et \vec{n}_q sont les vecteurs normaux unitaires aux points p et q , respectivement. La définition de ce poids part du principe que si les deux surfaces sont déjà à peu près alignées (ce qui est notre cas), deux points censés être appareillés ont forcément des normales dont les orientations sont approximativement les mêmes. Si ce n'est pas le cas, cela peut vouloir dire que le plus proche voisin trouvé pour le point concerné

¹Le principe de cet algorithme a été rappelé à la section 3 du chapitre 3.

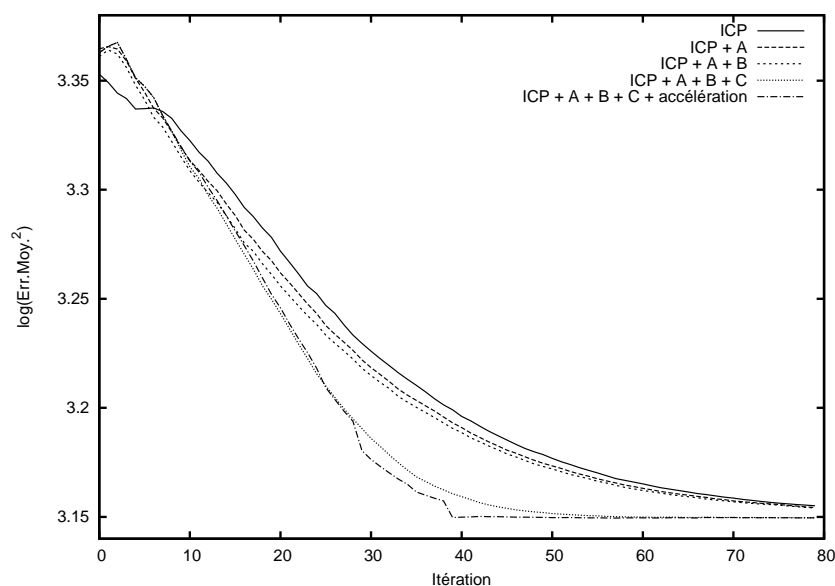


Figure 6.11 — Vérification expérimentale des constats établis par Rusinkiewicz *et al.* [158] sur l'efficacité de différentes optimisations de l'algorithme ICP.

ne constitue pas un choix d'appariement pertinent, et le poids prend donc une valeur faible, voire nulle.

- C— il est conseillé de travailler à partir d'un sous-échantillonnage des données initiales construit en choisissant une distribution uniforme sur la sphère de Gauss. Nous avons déjà parlé de cette technique dans le chapitre 3 : l'objectif de cette stratégie d'échantillonnage est de donner, lors du calcul de la transformation rigide, autant de poids à la forme générale de l'objet qu'aux fins détails de sa géométrie, sans quoi ces détails risquent de ne pas être correctement alignés. Nous utilisons donc cette stratégie, en prenant, pour ce faire, 1 à 2% des données initiales.

Il est également communément admis que si une correspondance contient au moins un point du bord de l'une des deux surfaces, elle est éliminée. Cela permet d'éviter d'établir des correspondances entre des points qui n'appartiennent pas à la zone de recouvrement.

Avant de les intégrer à notre implantation, nous avons testé chacune de ces optimisations dans le but de vérifier leurs contributions individuelles. Les résultats de ces tests sont reportés sur le graphique de la figure 6.11, qui donne l'erreur de recalage (distance moyenne au carré entre les deux surfaces recalées) en fonction du nombre d'itérations. On peut voir, par rapport à la version originale d'ICP, que chacune de ces variantes améliore effectivement la vitesse de convergence. La variante C, en plus d'un accroissement flagrant de la vitesse, provoque également un abaissement significatif du seuil de convergence. Ces tests confirment donc bien les conclusions de Rusinkiewicz *et al.* [158].

La dernière courbe de la figure 6.11, sur laquelle on peut voir des diminutions brusques de l'erreur de recalage entre les itérations 30 et 40, a été obtenue en ajoutant à notre implantation la technique d'accélération décrite par Besl *et al.* dans le papier original de l'algorithme ICP [13]. Cette accélération consiste à regarder l'évolution de la transformation de recalage

sur quelques itérations. Si cette évolution est régulière, elle est extrapolée de manière à faire avancer plus rapidement la transformation de recalage vers la solution. Cela explique les diminutions brusques de l'erreur.

Concernant la métrique employée, nous avons utilisé la distance Euclidienne point à point, et non la distance point à plan de Chen *et al.* [27], pour la raison très simple que la première est plus rapide à déterminer. La recherche des plus proches voisins est alors effectuée à l'aide de k D-arbres en dimension 3.

Enfin, nous avons choisi de faire s'exécuter notre algorithme ICP dans un processus à part, afin que l'opérateur puisse conserver le contrôle de l'application principale, au cas où le raffinement ne serait pas achevé à son retour. Il est donc tout à fait possible que plusieurs raffinements tournent en parallèle à un instant donné.

■ 4.2 Recalage global

Une fois la numérisation achevée, nous proposons un autre traitement visant à éliminer, cette fois de manière effective, l'effet d'accumulation de l'erreur : il s'agit d'une étape de recalage global. Nous avons vu qu'il existait deux approches différentes pour ce genre d'algorithmes : les méthodes par arbres de recouvrement minimal et les ICP globaux.

Les méthodes par arbres de recouvrement partent du principe qu'aucune information *a priori* n'est disponible. C'est un avantage, en ce sens que cela rend le processus de recalage totalement automatique, mais c'est également le principal défaut de cette catégorie d'algorithmes, dans la mesure où cela les rend très gourmands en temps de calcul (voire impraticables lorsque le nombre d'acquisitions à prendre en compte devient important). Les ICP globaux, quant à eux, nécessitent la connaissance d'un premier alignement initial. Ils fournissent cependant des résultats bien plus précis, du fait qu'il s'agisse de véritables algorithmes de minimisation, et en des temps d'exécution bien moindres. Pour ces raisons de précision, notre choix s'est porté sur le développement d'un ICP global, d'autant plus que l'alignement initial requis nous est fourni par les traitements précédents. Ce type d'algorithmes reste cependant encore trop coûteux pour être exécuté à la volée, c'est pourquoi il s'agit, dans notre cas, d'un traitement effectué *a posteriori*.

L'ICP global, nous le rappelons, est un algorithme de minimisation itérative visant à chercher, pour chaque point d'une surface donnée, ses plus proches voisins dans toutes les autres surfaces, et d'établir autant de correspondances avec ce point qu'il y a de plus proches voisins trouvés. La transformation de recalage est alors calculée à partir de toutes ces correspondances. Étant donné qu'il s'agit d'un ICP, certaines des optimisations dont nous avons discuté dans la section précédente restent toujours valables. Par exemple, le sous-échantillonnage sur la sphère de Gauss peut aussi être employée ici, de même que l'échantillonnage bilatéral, la pondération des correspondances par un indice de confiance et l'élimination des correspondances impliquant des points du bord. Nous avons cependant ajouté quelques suppléments à cette version globale.

En premier lieu, il faut remarquer que si le nombre d'acquisitions est très important, il n'est pas possible de toutes les garder en mémoire à la fois. Nous avons donc développé notre algorithme de recalage global à l'aide d'un mécanisme *hors mémoire*, proposant un chargement à la demande des différentes acquisitions. De même, pour éviter d'avoir à maintenir autant de kD -arbres qu'il y a d'acquisitions, nous avons choisi d'adopter une recherche de plus proches voisins utilisant les informations de calibrage du dispositif de mesure de formes pour reprojeter les points d'une acquisition dans l'espace de l'image télémétrique d'une seconde, à l'instar de Blais *et al.* [14].

En second lieu, on peut constater que si l'alignement initial est déjà très bon, les relations de recouvrement entre les différentes acquisitions ne changeront pas beaucoup au fur et à mesure des itérations. Or, connaître ces relations *a priori* permet d'éviter d'effectuer des recherches de plus proches voisins entre deux surfaces pour lesquelles on sait qu'aucun recouvrement n'existe. Nous proposons alors de détecter ces recouvrements en deux étapes :

- avant le lancement de l'algorithme, des tests d'intersection entre les boîtes englobantes des différentes acquisitions sont effectués : si deux surfaces ont leurs boîtes englobantes qui s'intersectent, elles sont considérées comme ayant une zone de recouvrement en commun. Ce test permet d'avoir une première idée des relations de recouvrement qui existent dans les données.
- un second test est effectué à chaque itération de l'ICP global, mettant à jour les relations de recouvrement si celles-ci évoluent. Si, à l'issue de la recherche de correspondances entre deux surfaces, aucun appariement valide de points n'a été trouvé, la relation de recouvrement qui lie ces deux surfaces est rompue, et ne sera donc plus testée lors des itérations suivantes.

Ce second test est en fait surtout important lors de la première itération, car il permet de raffiner le test des boîtes englobantes, qui a tendance à largement surestimer les relations de recouvrement. Pour la détection de ces relations, on peut également noter l'approche de Pingi *et al.* [139], qui proposent d'utiliser une grille 3D très grossière stockant, en chaque voxel, les identifiants des surfaces qui s'y trouvent.

5 Résultats expérimentaux

Dans cette section, nous évaluons indépendamment les différents composants que nous avons intégrés à notre protocole d'acquisition, à savoir le recalage local de points de vue, le recalage géométrique chaîné par bloc, le raffinement par ICP et le recalage global.

Afin de révéler l'intérêt du raffinement et du recalage global, des comparaisons sur la précision des différents composants sont présentées. De plus, étant donné que notre protocole se veut interactif afin de permettre un contrôle durant la mesure elle-même, nous allons également présenter des mesures de performances. À ce propos, toutes les mesures de performances qui sont présentées dans cette section ont été obtenues sur un PC de bureau doté d'un processeur AMD Athlon 3800+.

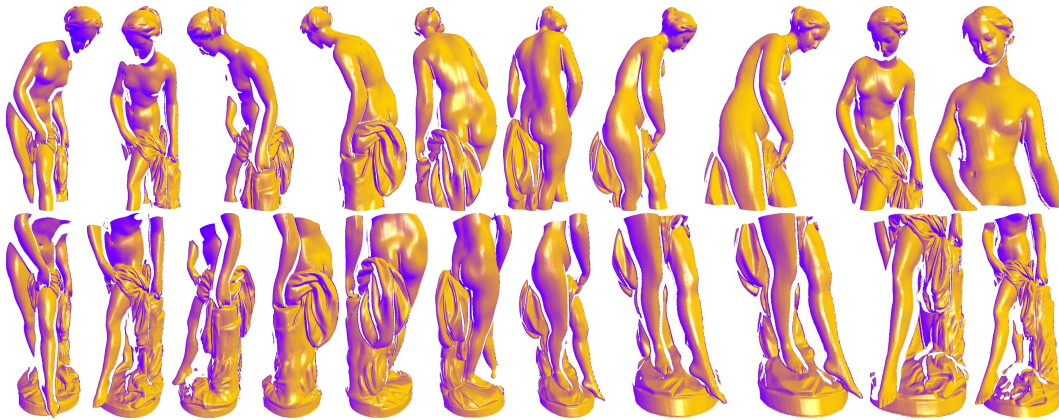


Figure 6.12 – Les 21 morceaux de surfaces composant la chaîne d’acquisitions de la Venus au Bain.

■ 5.1 Étude du recalage géométrique par phase

Concernant le recalage par phase, la figure 6.13 montre un exemple d’images produites par notre logiciel d’acquisition. Plus particulièrement, l’image de droite montre le résultat du recalage chaîné des 21 acquisitions illustrées sur la figure 6.12. Ce résultat est le fruit de l’unique recalage par phase, sans raffinement ni passe globale. On voit clairement que ce recalage est déjà suffisamment précis pour permettre à l’opérateur de contrôler visuellement la mesure, sans nécessairement rajouter des traitements supplémentaires.

Nous allons maintenant étudier l’influence du seuil S_δ sur la précision du recalage obtenu. Comme nous l’avons évoqué lorsque nous avons introduit ce seuil dans la section 2.3, plus sa valeur est basse, plus les correspondances retenues sont censées être fiables. En revanche, elles sont également moins nombreuses, ce qui signifie que l’algorithme de calcul de transformations rigides devra travailler sur un échantillonnage plus restreint, et donc moins représentatif. Un abaissement du seuil ne garantit donc pas forcément une amélioration du recalage.

Pour vérifier cela, nous avons effectué le recalage par phase de la même paire de surfaces en prenant à chaque fois un seuil S_δ différent. Voici les valeurs que nous avons testées, ainsi que le nombre de correspondances auquel chacune aboutit :

- $S_\delta = 0.125$: ~ 8200 correspondances,
- $S_\delta = 0.100$: ~ 3800 correspondances,
- $S_\delta = 0.050$: ~ 250 correspondances,
- $S_\delta = 0.025$: 14 correspondances.

Les résultats sont donnés par la figure 6.14. Il s’agit de l’histogramme des points présents dans la zone de recouvrement commune aux deux surfaces en fonction de la distance à leurs plus proches voisins respectifs dans la surface opposée. Ce graphique s’interprète alors de la façon suivante : étant donné qu’un bon recalage amène les points plus près de leurs homologues dans l’autre surface qu’un recalage moins bon, plus l’histogramme est tassé vers la gauche du graphique, meilleur est le recalage qu’il représente.

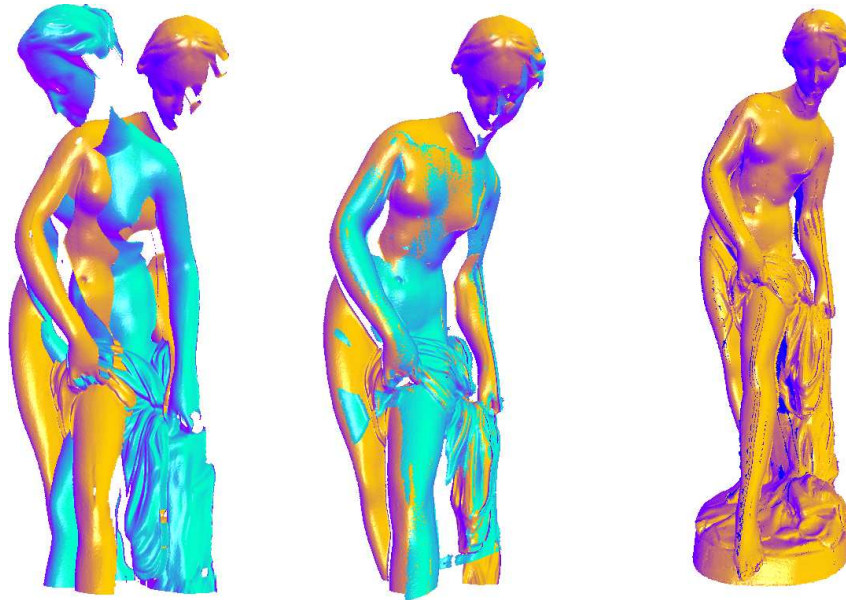


Figure 6.13 – Recalage par phase. **Gauche et milieu** : deux surfaces, avant et après recalage par la phase. **Droite** : objet obtenu après recalage des 21 acquisitions.

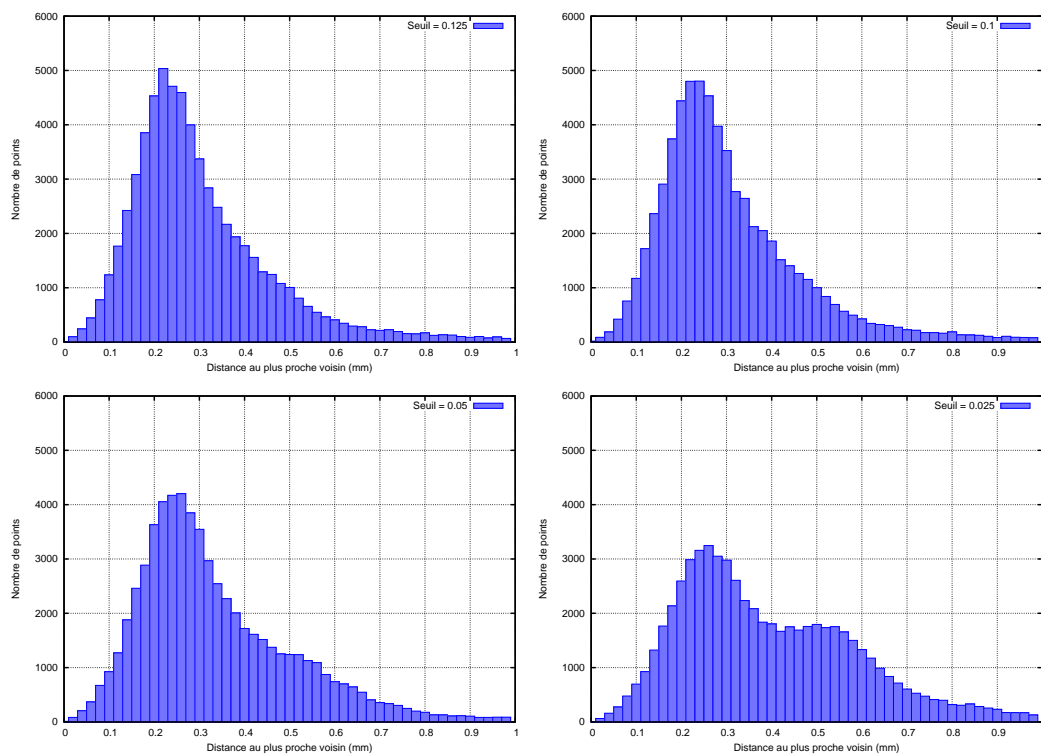


Figure 6.14 – Qualité du recalage par phase pour différentes valeurs du seuil S_δ .

Taille surface 1 (# points)	Taille surface 2 (# points)	# corres. trouvées	Temps de recalage (ms)		
			construction <i>k</i> D-arbres	recherche corres.	calcul transfo.
326K	330K	15K	472,94		
			152,26	320,03	0,65
323K	350K	35K	648,54		
			169,69	477,34	1,51
212K	214K	9500	382,69		
			97,84	284,45	0,40
132K	135K	4000	365,40		
			61,28	303,93	0,19

Tableau 6.1 – Temps mesurés pour le recalage par phase de différentes paires d’acquisitions.

Au vu de ces quatre graphiques, on se rend alors compte que notre assertion initiale affirmant qu’une valeur plus faible du seuil aboutit à un meilleur recalage ne se vérifie pas. Il s’agit en fait même du contraire. Le calcul de la transformation rigide est donc plus sensible au nombre et à la distribution des correspondances qu’à leur précision. Il faut cependant noter que pour des valeurs de seuil trop hautes, le recalage peut échouer du fait des correspondances invalides qui peuvent alors survenir.

Le choix de la valeur de S_δ n’est donc pas un problème trivial, ce qui laisse à penser qu’il est un facteur limitant à l’automatisme de notre protocole. Il n’en est en fait rien : sa valeur est directement corrélée à la résolution de la phase en espace projecteur, ce qui signifie qu’une fois que la bonne valeur a été trouvée, elle peut être fixée une fois pour toute pour un dispositif de mesure de forme donné, indépendamment de l’objet mesuré. La détermination de ce seuil peut donc être intégrée à l’étape de calibrage du dispositif. Ne disposant pas nous même de l’information de résolution précitée, nous avons fixé ce seuil de manière empirique ($S_\delta = 0, 1$) et avons effectivement pu constater que la valeur ainsi trouvée a donné de bons résultats pour toutes les mesures que nous avons pu faire avec ce même dispositif.

En termes de performance, les temps nécessaires au recalage par la phase de différentes paires d’acquisitions sont donnés dans le tableau 6.1. La partie la plus lourde de ce recalage est la recherche dans l’espace de la paramétrisation Φ . Étant donné que cette recherche est effectuée à l’aide d’un *k*D-arbre de dimension 2, elle demeure très rapide, même pour des jeux de données de taille importante. On voit clairement que des temps aussi courts permettent aisément un contrôle interactif de la mesure en effectuant le recalage d’une surface immédiatement après son acquisition, comme nous le souhaitons. Cet objectif est donc atteint.

Il est important de noter que le choix du seuil S_δ dont nous avons discuté précédemment n’a aucune influence sur les temps de recalage : en effet, la comparaison à ce seuil ne peut être effectuée qu’une fois que la correspondance d’un point par rapport à la phase a été trouvée. La recherche de plus proche voisin dans l’espace de la paramétrisation Φ doit donc de toute manière être effectuée pour la totalité des données. Seul le calcul de la transformation

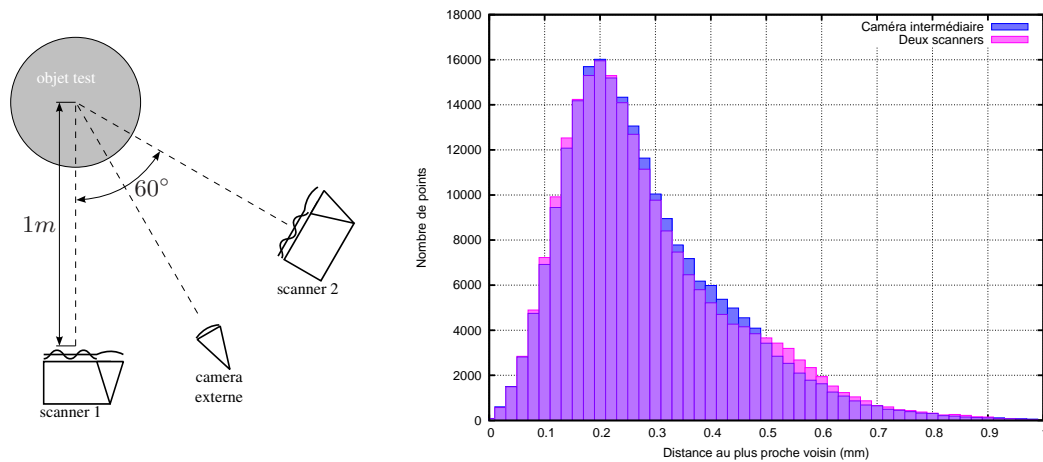


Figure 6.15 — Mesure de la différence de précision pour le recalage géométrique entre la configuration matérielle impliquant deux scanners et celle impliquant une caméra externe. **Gauche** : positionnement du matériel pour la réalisation de cette mesure. **Droite** : comparaison de la précision des deux recalages obtenus.

rigide est en fait dépendant du nombre de correspondances trouvées (et donc du seuil) mais, comme on peut le voir sur le tableau 6.1, ce calcul est quasi instantané, quelque soit le cas.

■ 5.2 Comparaison des différentes configurations matérielles

Nous avons fait allusion, dans la section 3.2, à deux configurations matérielles possibles pour effectuer le recalage géométrique par phase :

- la configuration n'utilisant qu'un seul scanner et une caméra externe,
- la configuration impliquant deux scanners.

Nous avons donc voulu tester la différence de précision existant entre ces deux versions.

Pour ce faire, nous nous sommes procurés un second scanner 3D, du même type que celui dont nous disposions déjà. Les deux scanners ont été placés approximativement à la même distance d'un objet test, et de manière à ce que l'angle qui les sépare par rapport à cet objet soit d'environ 60° . La caméra externe a alors été placée entre les deux scanners, toujours à la même distance de l'objet. La configuration de ce banc de tests est illustrée sur la figure 6.15-gauche. Les deux surfaces à recalcer correspondent aux images télémétriques acquises par chacun des deux scanners. Toutes les données nécessaires pour effectuer les deux versions du recalage géométrique ont alors été acquises sans qu'aucun de ces dispositifs ne soit déplacé durant toute la mesure. La figure 6.15-droite compare les résultats obtenus pour ces deux versions.

On peut ainsi constater que les deux histogrammes obtenus sont quasiment confondus. La différence de précision entre les deux versions est donc négligeable, alors que l'on aurait pu s'attendre à une perte due à l'introduction de la caméra externe comme dispositif intermédiaire. Ce résultat démontre le fait que le recalage géométrique n'utilisant qu'un scanner est tout à fait exploitable dans le cas concret.

Il aurait été intéressant de comparer deux chaînes d'acquisition complètes, chacune ayant été capturée avec l'une des deux variantes. Malheureusement, des modifications relativement peu évidentes ont déjà dû être mises en œuvre pour pouvoir effectuer la comparaison précédente. Aussi, cela n'a pas pu être fait et reste donc en suspens pour le moment.

■ 5.3 Étude du raffinement à la volée

Nous allons maintenant voir quel est l'apport de l'algorithme ICP exécuté à la volée dans le but de raffiner les transformations trouvées lors du recalage par phase. Nous avons, pour cela, considéré différentes paires d'acquisitions, chacune correspondant à un maillon de l'une des chaînes de recalage que nous avons capturées. Nous avons alors comparé les histogrammes de recalage pour chacune de ces paires, avant et après raffinement par ICP. Les résultats sont reportés sur la figure 6.16.

Comme on peut le voir sur les trois exemples illustrés, les cas de figure peuvent être très variés :

- graphique 6.16-a : le recalage par phase est à ce point précis que le raffinement n'est d'aucune utilité.
- graphique 6.16-b : le raffinement apporte une amélioration légère, qui n'est pas forcément perceptible au niveau visuel, mais qui est cependant notable en termes de gain de précision. Il s'agit du cas de figure le plus répandu (environ 80% d'après nos tests).
- graphique 6.16-c : le raffinement corrige une grosse faiblesse du recalage par phase, survenue pour diverses raisons, et qui provoque un défaut d'alignement nettement perceptible entre les deux surfaces. La figure 6.16-d illustre le genre d'erreurs d'alignement très visibles que l'on peut observer dans ce cas, ainsi que la correction issue du raffinement. Ce troisième cas justifie à lui seul l'intérêt d'un raffinement.

On voit donc que le raffinement aboutit toujours à une amélioration du résultat. On peut notamment voir que les histogrammes représentant les résultats du raffinement pour chacun des trois exemples ont à peu près la même forme, ce qui signifie que la précision obtenue grâce à cette étape est à peu près constante, quelque soit le cas.

Concernant les performances, le tableau 6.2 donne les temps d'exécution du raffinement par ICP pour différentes paires d'acquisitions. Ces mesures représentent le temps nécessaire à ICP pour s'initialiser, ainsi que le temps moyen nécessaire au déroulement d'une itération (lorsque deux surfaces sont déjà bien recalées, comme dans notre cas, ce temps est à peu près constant). Même pour une vingtaine d'itérations, ce qui est généralement plus que suffisant étant donné la qualité du recalage par phase, ces temps montrent bien que l'exécution du raffinement en parallèle de la mesure est tout à fait acceptable.

L'intégration des optimisations dont nous avons discutées dans la section 4.1 permet, bien entendu, d'accélérer la vitesse de convergence, mais ce qui rend le calcul rapide est surtout le fait que nous ne travaillons que sur un sous-échantillonnage des données. Ce sous-échantillonnage ne contient que 2% des données complètes. Bien entendu, en gardant davantage de points, la précision du raffinement serait accrue, mais ses performances chuteraient drastiquement. Or, il est important pour nous que le raffinement reste rapide afin que

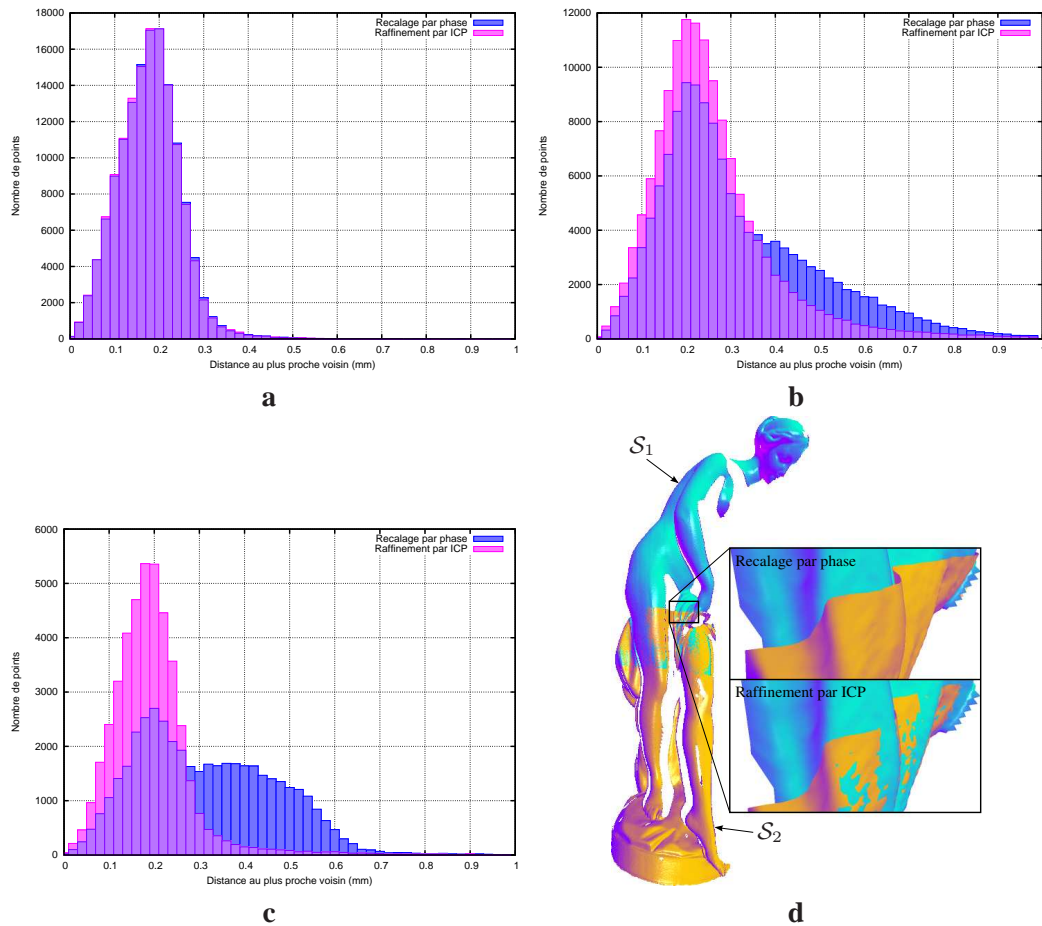


Figure 6.16 – Apport du raffinement par ICP sur les transformations calculées par le recalage par phase.

Taille surface 1 (# points)	Taille surface 2 (# points)	Temps de raffinement (ms)	
		initialisation	ICP : 1 itération
124K	124K	610	513
181K	226K	848	171
328K	315K	1266	1737
323K	350K	1308	3834

Tableau 6.2 – Temps mesurés pour le raffinement par ICP de différentes paires d'acquisitions.

l'opérateur puisse bénéficier de ses effets au cours de la mesure.

■ 5.4 Étude de l'accumulation de l'erreur et du recalage global

Pour étudier le phénomène d'accumulation de l'erreur introduit par le processus de recalage chaîné, nous avons regardé l'histogramme des distances aux plus proches voisins généré à partir de la zone de recouvrement existant entre la première et la dernière image télémétrique de notre chaîne de 21 acquisitions, c'est-à-dire entre les deux surfaces de cette chaîne pour lesquelles l'erreur de recalage est potentiellement la plus grande.

Nous avons alors comparé deux autres histogrammes à celui obtenu par le recalage par phase :

- le premier nous permet de vérifier l'impact du raffinement par ICP sur l'accumulation de l'erreur. En effet, cette phase a été initialement développée dans le but de réduire, durant la mesure elle-même, cet effet d'accumulation. Nous avons vu précédemment que ce raffinement permettait d'améliorer les recalages locaux, mais il est important de vérifier si notre objectif de correction de l'erreur à un niveau plus global est également atteint.
- Le second histogramme, quant à lui, nous permet d'illustrer l'intérêt que présente une passe de recalage globale par rapport à la seule application d'un raffinement des transformations locales.

Les résultats de ces comparaisons sont respectivement illustrés par les figures 6.17-gauche et 6.17-droite.

Le graphique de gauche nous montre que le raffinement par ICP a bel et bien un effet sur l'accumulation de l'erreur : comparé au recalage par phase, la précision semble quelque peu améliorée. Si cette amélioration semble faible, on peut regarder la figure 6.18 pour se rendre compte de son effet réel : l'image de gauche, qui correspond au recalage par phase, présente de gros défauts d'alignement au niveau des plis du linge. Creux et bosses ne sont absolument pas en phase entre les deux surfaces du fait que la surface bleue (vingt et unième de la chaîne) est légèrement trop haute par rapport à la surface orange (première de la chaîne). Cet effet est le résultat de l'accumulation de l'erreur que nous cherchons à éliminer. La figure du milieu montre le résultat fourni par le raffinement : on voit clairement que la surface bleue a été descendue de manière à mieux coller à la surface orange. L'amélioration est donc, cette fois, manifeste. On peut alors dire que notre objectif visant à réduire des défauts de précision au niveau global à l'aide d'un raffinement exécuté à un niveau local est atteint.

Concernant le recalage global, les histogrammes de la figure 6.17 nous montrent une nette amélioration du recalage entre la première et la dernière acquisition de la chaîne. Ce résultat n'est bien entendu pas une surprise, dans la mesure où ce genre d'algorithmes est clairement conçu dans ce but. Au niveau visuel, la figure 6.18-droite nous montre encore une amélioration de l'alignement entre la surface orange et la surface bleue par rapport à ce qu'a déjà produit le raffinement par ICP, révélée par une meilleure interpénétration entre ces deux surfaces. L'intérêt d'une passe de recalage global est donc manifeste.

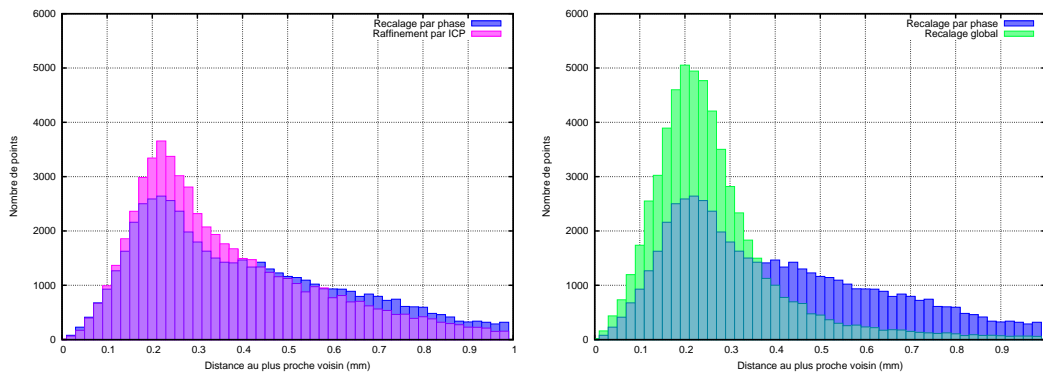


Figure 6.17 – Histogramme des distances aux plus proches voisins construit à partir de la zone de recouvrement entre la première et la dernière image télémétrique d'une chaîne de 21 acquisitions. **Gauche** : comparaison entre le recalage par phase et le raffinement par ICP. **Droite** : comparaison entre le recalage par phase et le recalage global.

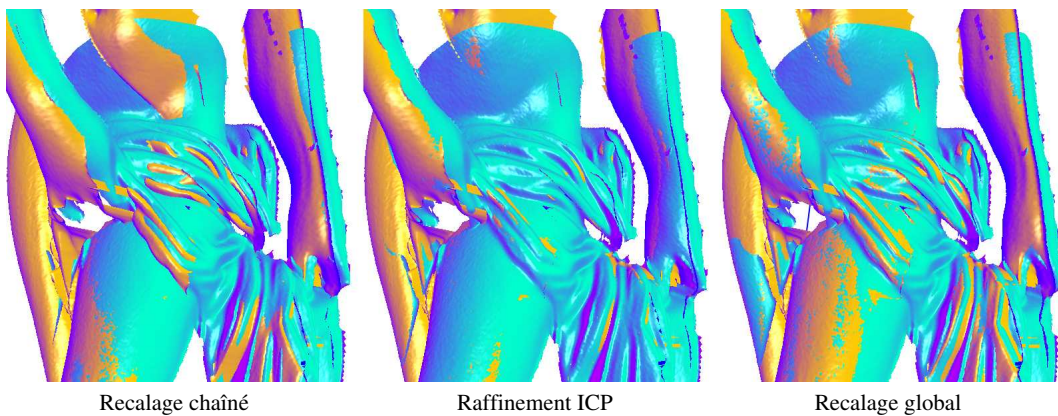


Figure 6.18 – Première et dernière surfaces d'une chaîne de 21 acquisitions. **Gauche** : recalage par phase, uniquement. Un mauvais alignement dû à l'accumulation de l'erreur est nettement visible. **Milieu** : amélioration induite par le raffinement des transformations locales. **Droite** : résultat du recalage global.

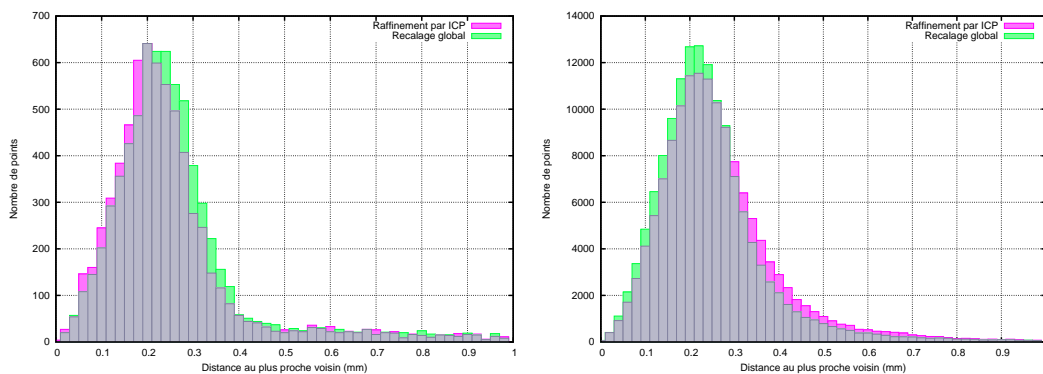


Figure 6.19 – Influence du recalage global sur les transformations locales déterminées en chaque maillon de la chaîne par le processus de raffinement.

Un autre point auquel on peut s'attendre avec ce genre d'algorithmes est une réduction de la précision des recalages locaux (c'est-à-dire, dans notre cas, ceux calculés entre deux maillons de la chaîne de recalage), du fait que, dans le cas global, chaque surface est déplacée de manière à tenir compte à la fois de toutes les surfaces avec lesquelles elle présente un recouvrement. Le fait que ces recalages locaux soient altérés par la passe globale peut éventuellement signifier un désalignement des surfaces impliquées.

Pour vérifier ces suppositions, nous avons comparé, pour chaque maillon de la chaîne, les transformations de recalage obtenues soit par raffinement uniquement, soit par raffinement suivi d'un recalage global. Les histogrammes correspondant à deux de ces comparaisons sont donnés par la figure 6.19. Deux cas de figure sont illustrés : dans le cas du graphique de gauche, le recalage global engendre effectivement une petite perte de précision, alors que dans le cas du graphique de droite, le recalage global aboutit, au contraire, à une amélioration. Nous avons présenté ce second cas pour montrer que l'altération des recalages locaux par le recalage global n'est pas systématique, mais il est important de remarquer que ce second cas est en fait très rare. Quoiqu'il en soit, au vu de cette figure, il est surtout évident que les différences de précision entre le recalage global et les transformations calculées localement par le processus chaîné, raffinement inclus, sont insignifiantes.

Pour ce qui est des performances, le recalage global dépend du nombre de surfaces à recalage, mais également du nombre de relations de recouvrement entre ces surfaces. Pour l'exemple des 21 acquisitions de la Venus au Bain, ce recalage demande, en moyenne, 11 secondes par itération. Il faut cependant rappeler que ce recalage global est un post-traitement, ne s'exécutant qu'une fois la mesure achevée : sa vitesse d'exécution ne constitue donc, en aucun cas, un facteur déterminant.

■ 5.5 Étude du recalage local de points de vue

Concernant la localisation de points de vue, nous n'avons pour le moment pas trouvé de moyen d'évaluer la qualité des résultats autrement que par un contrôle visuel. Nous allons donc montrer quelques exemples représentant la silhouette du modèle 3D telle qu'elle est perçue depuis le point de vue considéré, en surimpression de la photographie.

La première série d'images a été obtenue lors de la numérisation d'un vase de style grec antique. Pour la capture du champ lumineux, nous avons pris une série de 27 photographies. Quatre de ces photographies sont présentées sur la figure 6.20. Le contour vert représente la silhouette de l'objet 3D correspondant, projeté en espace image, et les flèches que l'on peut voir sur certaines de ces images sont là pour mettre en valeur les zones où des défauts d'alignement sont visibles. La direction des flèches indique la direction de divergence du contour objet par rapport à la silhouette dans la photographie. Les points de vue correspondant à ces 27 photographies sont représentés sur la figure 6.21.

Sur les 27 photographies, une seule présente des défauts d'alignement nettement visibles entre le modèle 3D projeté et la photographie. Il s'agit de la photographie de droite de la figure 6.20. Les problèmes rencontrés pour le calibrage de cette photographie peuvent



Figure 6.20 – Estimation visuelle de la qualité du calibrage caméra, par comparaison de la silhouette projetée de l'objet 3D dans l'espace image par rapport à son contour dans la photographie. Données obtenues par la numérisation du vase grec.

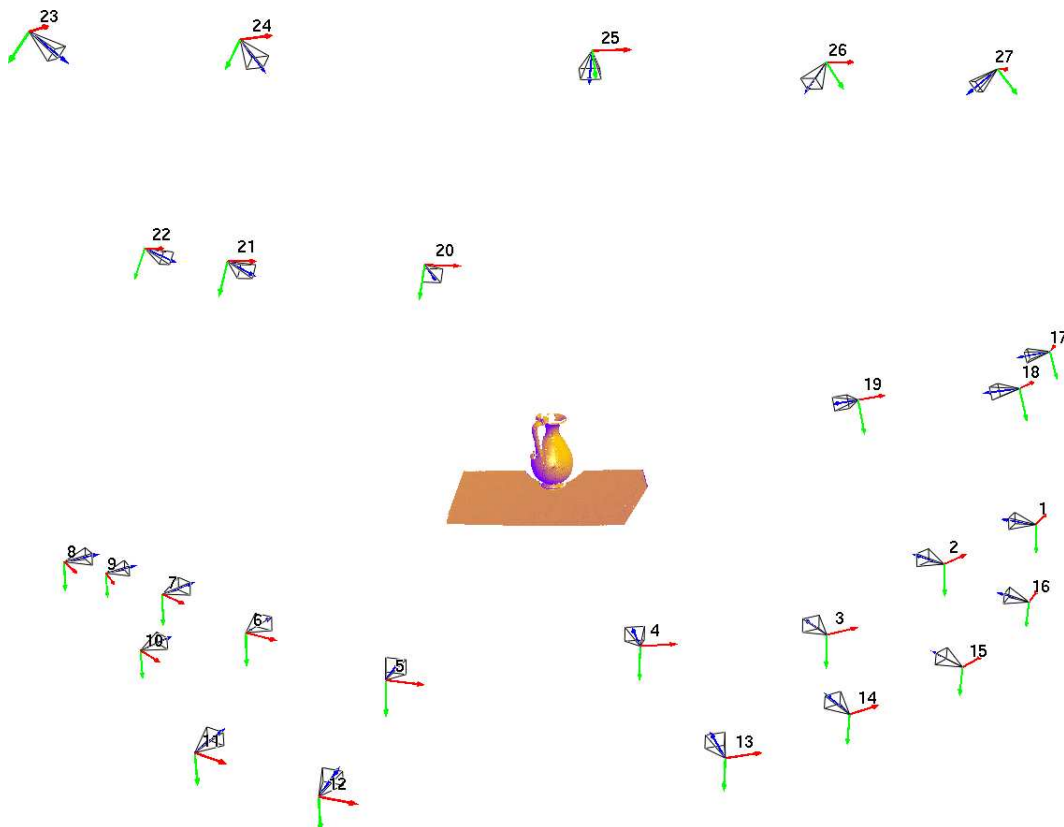


Figure 6.21 – Les points de vue associés aux 27 photographies du vase grec, et localisés grâce à la paramétrisation par phase.

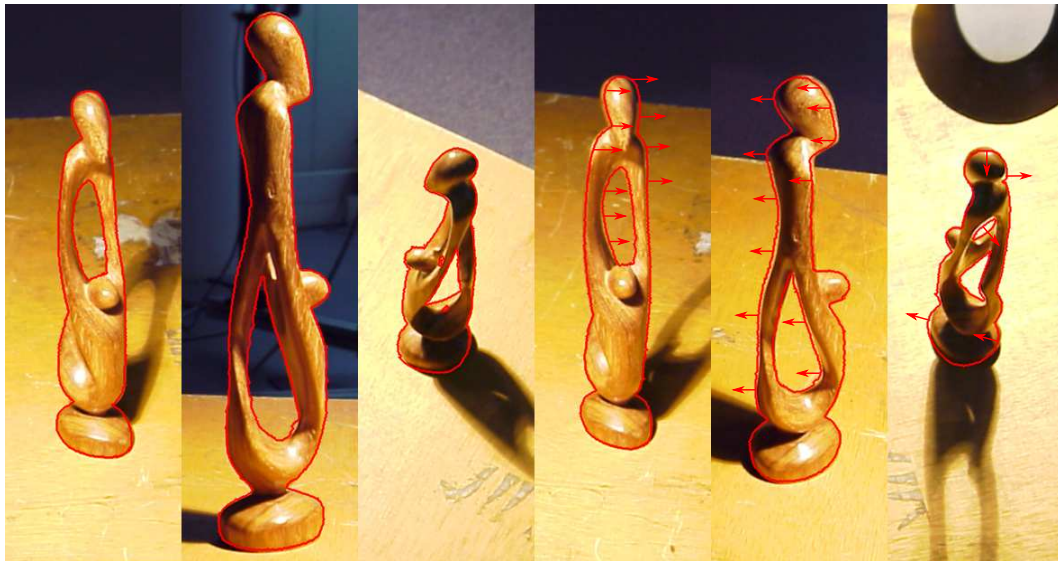


Figure 6.22 — Estimation visuelle de la qualité du calibrage caméra, par comparaison de la silhouette projetée de l'objet 3D dans l'espace image par rapport à son contour dans la photographie. Données obtenues par la numérisation de la statue africaine.

s'expliquer par le fait que son acquisition s'est déroulée dans des conditions extrêmes, en ce sens que l'angle entre la caméra externe et le scanner 3D par rapport à l'objet était très important lors de la mesure. Cela implique que seule une petite fraction de la région sur laquelle a été projetée la paramétrisation Φ était visible de cette caméra. Ainsi, peu de correspondances ont pu être extraites, et leur répartition en espace image n'était pas forcément suffisamment homogène. Le recalage n'a donc pas pu se dérouler correctement.

La seconde série d'images que nous allons présenter ici a été obtenue à partir de la numérisation d'une statue en bois vernis, de style africain. Cette série comporte, en tout, 42 photographies. Certaines d'entre elles sont illustrées sur la figure 6.22. Pour plus de visibilité, le contour est maintenant tracé en rouge.

Six de ces photographies présentent des erreurs d'alignement aisément perceptibles. Des exemples sont donnés sur les trois images de droite de la figure 6.22. Une fois de plus, ces erreurs proviennent du fait que, pour ces points de vue, le nombre de correspondances n'est pas suffisant, de même qu'elles ne sont pas suffisamment bien réparties dans l'espace image.

Il faut également remarquer que cette statue est en elle-même assez problématique : sa surface vernie fait que le bruit de numérisation dans les données 3D acquises est plus important que pour le modèle précédent du vase grec, qui présentait un matériau plutôt mat. Les correspondances trouvées sont donc forcément moins précises. De plus, le scanner 3D que nous utilisons n'est pas prévu pour des objets d'une si petite taille. Cela implique que la zone occupée par cet objet dans l'espace image des différents dispositifs (scanner ou caméra externe) ne permet pas une bonne distribution des correspondances pour le recalage de points de vue. Nous n'avons malheureusement pas testé ce que cela aurait donné si la

caméra d'acquisition avait été plus proche de l'objet à mesurer, de manière à ce que celui-ci couvre une zone plus importante de l'image.

Il est également important de noter que nous avons réalisé ces mesures avec une optique fixe pour la caméra externe, en partant du principe que les paramètres intrinsèques avaient été déterminés au préalable. Or, cela a été accompli à partir des données issues de l'acquisition du tout premier point de vue, ce qui introduit forcément une source d'erreur supplémentaire. Une véritable phase de précalibrage permettrait sans doute d'améliorer ces résultats.

Concernant le seuil δ_{Φ} utilisé pour sélectionner les correspondances récupérées grâce à la phase, son influence sur la précision du calibrage est quasi nulle, contrairement à ce que nous avons observé pour le recalage géométrique. Les points de vue pour lesquels l'alignement des silhouettes dans l'espace image est mauvais avec une valeur de seuil donnée ne se voient pas améliorer avec une valeur plus faible ou plus grande. Il en va de même pour ceux dont l'alignement est bon : changer la valeur du seuil ne détériore pas, ni n'améliore, les résultats (à l'exception, bien entendu, d'un seuil choisi trop bas, auquel cas le nombre de correspondances devient insuffisant).

Concernant les deux modifications que nous avons apportées à la méthode dans le but de la rendre plus précise, l'élimination des points situés près des sauts de profondeur est la plus efficace : le calibrage caméra est, en effet, très sensible aux faux-positifs dans les correspondances. Sans cette modification, les points de vue pour lesquels le processus échoue totalement sont plus nombreux (c'est-à-dire ceux pour lesquels un alignement des silhouettes en espace image, même de mauvaise qualité, ne parvient pas à être trouvé). Elle se révèle donc indispensable. En revanche, la seconde modification, destinée à garantir une bonne répartition des correspondances en espace image, s'est révélée presque sans effet sur la précision : les points de vue pour lesquels l'alignement n'était pas tout à fait correct n'ont pas été améliorés. Nous avons tout de même pu constater, à de rares occasions, que cette seconde modification pouvait empêcher que le calibrage de certains points de vue n'échoue totalement. Même si les résultats, en termes de précision, sont un peu décevants, cette modification présente tout de même l'avantage de permettre le contrôle du nombre de correspondances à fournir à l'algorithme de calibrage, et ainsi en accélérer l'exécution. Nous l'avons donc conservée, et nous en avons d'ailleurs tenu compte dans les mesures de performances qui vont suivre.

De l'ensemble des exemples présentés, on peut voir que les erreurs dans la localisation ne sont jamais très importantes : les résultats obtenus pourraient donc servir de point de départ à un algorithme de raffinement, comme cela a été proposé dans le cas du recalage géométrique. Bien entendu, dans le cas où le calibrage échoue totalement, les points de vue concernés doivent immédiatement être éliminés, et la mesure refaite. Nous allons voir de suite que les temps requis pour effectuer le calibrage le permettent. Il faut cependant remarquer que, même si le calibrage semble moins robuste que le recalage géométrique, il aboutit tout de même à un résultat exploitable dans la plupart des cas, comme en attesteront les résultats présentés dans le chapitre 8, lorsque nous nous intéresserons à la visualisation des copies numériques.

Concernant les performances, le tableau 6.3 donne les temps mesurés lors du calibrage de

Taille surface (# points)	# corres. trouvées	Temps de calibrage (ms)		
		recherche corres.	répartition uniforme	calibrage
212K	53K	563		
		293	37	233
232K	4450	714		
		173	8	533
326K	44K	717		
		390	32	295
326K	10K	627		
		359	9	259

Tableau 6.3 – Temps mesurés pour le calibrage local de différentes photographies.

quelques unes des photographies acquises. Puisque nous nous sommes placés dans le cas d'une caméra à optique fixe, ces temps ne correspondent qu'au calibrage des paramètres extrinsèques. Les temps de construction des kD -arbres ne sont cette fois pas mentionnés, puisqu'elle est censée avoir déjà été réalisée pour permettre le recalage géométrique. Ces constructions n'ont de toute façon besoin d'être effectuées qu'une seule fois par bloc d'échantillonnage local, immédiatement après que la géométrie ait été identifiée à l'aide de la paramétrisation Φ . Il est donc inutile de s'en préoccuper ici. Au vu des chiffres présentés, on peut constater que nos objectifs d'interactivité pour le contrôle à la volée sont à nouveau atteints.

6 Bilan

Dans ce chapitre, nous avons proposé un protocole conçu pour l'acquisition de champs lumineux surfaciques. Ce protocole permet d'effectuer le recalage géométrique et la localisation de points de vue, qui sont les deux tâches essentielles à l'acquisition d'une telle information, de manière entièrement automatique. De plus, le matériel impliqué est très léger, que ce soit au niveau du coût, des besoins en calibrage et de l'encombrement, ce qui le rend particulièrement mobile et permet donc un déploiement sur site pour effectuer la mesure d'objets d'art fragiles, qu'il est généralement préférable de ne pas déplacer.

Les deux tâches de recalage et de localisation de points de vue, en plus d'être automatisées, sont également exécutées à la volée de manière à permettre à l'opérateur de contrôler le bon déroulement de la mesure. Pour arriver à cela, nous avons utilisé un modèle de lumière structurée pour paramétrer chaque point de la surface mesurée à l'aide d'un identifiant dont le calcul est indépendant du point de vue. Cet identifiant nous permet alors d'établir, de façon quasi immédiate, des correspondances entre les différents jeux de données, et donc de déterminer en temps interactifs les transformations de recalage et de calibrage caméra.

Le recalage géométrique étant effectué de manière chaînée, nous avons introduit deux processus de correction pour pallier au problème de l'accumulation de l'erreur :

- le premier est un raffinement effectué par un algorithme de type ICP, tournant en parallèle pendant que l'opérateur continue la mesure,
- le second est un algorithme de recalage global exécuté, quant à lui, *a posteriori*, c'est-à-dire une fois la mesure complètement achevée.

Les études que nous avons menées au sujet de ce protocole ont montré que les objectifs que nous nous étions fixés ont été atteints. Il reste cependant un certain nombre d'éléments qui peuvent encore être discutés.

En premier lieu, il faut constater que le temps gagné par l'automatisation des tâches de recalage et de calibrage est partiellement perdu en temps de mesure : en effet, pour chaque point de vue échantillonnant la luminance, une acquisition de phase est nécessaire, ce qui prend beaucoup plus de temps que la capture d'une simple photographie. Il en va de même, d'ailleurs, pour le recalage géométrique. Concernant ce dernier point, des méthodes de recherche automatique de correspondances pourraient éventuellement être utilisées, bien qu'il ne soit pas certain que les calculs coûteux qu'elles nécessitent ne prennent moins de temps que l'acquisition d'une paramétrisation par la phase. Pour le recalage de points de vue, en revanche, nous ne connaissons pas de techniques, à l'heure actuelle, qui soient capables de fournir un retour interactif dans le cas des conditions arbitraires d'un environnement non contrôlé (mesure sur site, hors laboratoire).

En second lieu, au même titre que nous effectuons actuellement un raffinement du recalage géométrique, il pourrait être intéressant de proposer un raffinement de la localisation des points de vue. Dans ce cas, il faudrait travailler avec l'information disponible dans chaque photographie pour établir des correspondances avec la géométrie. Mais cela pose, une fois encore, des difficultés lorsqu'il s'agit d'effectuer une mesure sur site, sans contrôle de l'environnement. Une méthode qui pourrait être mise en œuvre serait d'associer à chaque point 3D acquis une information de couleur en prenant simplement un cliché avec la caméra embarquée dans le scanner juste après avoir capturé une image télémétrique. Cette information chromatique pourrait alors être utilisée pour trouver des correspondances 3D/2D en projetant cette «géométrie colorée» dans l'espace image du point de vue, grâce aux paramètres caméra précédemment trouvés, et en faisant coïncider l'image ainsi générée avec la photographie acquise à l'aide d'algorithmes de corrélation d'images, un peu comme ce que font Bernardini *et al.* [10].

Toujours à propos de ce calibrage de points de vue, nous aimerions également trouver un moyen, autre que la simple estimation visuelle, d'en déterminer la précision. Généralement, cela se fait en testant la distance, en nombre de pixels, entre le projeté d'un point 3D en espace image et le pixel qui est censé lui correspondre. Mais comme nous l'avons dit, à part les correspondances fournies par la phase, il est difficile de connaître cette relation point/pixel autrement que par une sélection manuelle. Une fois de plus, si lors de l'acquisition, une information chromatique peut être associée à la géométrie, des algorithmes de corrélation d'images pourraient être utilisés dans ce but.

Enfin, concernant le recalage géométrique, un élément dont nous n'avons pas discuté jus-

qu'à présent est le critère d'arrêt des deux algorithmes itératifs de minimisation que nous utilisons : le raffinement et le recalage global. Actuellement, ces deux algorithmes s'exécutent suivant un nombre fixe d'itérations, que nous avons déterminé empiriquement à l'issue de nombreux tests. Un critère basé sur la précision devra cependant être intégré au dispositif si l'on souhaite pouvoir le déployer lors de campagnes de numérisation de grande envergure.

- Traitements géométriques -

Alignement non rigide des acquisitions, automatisation du filtrage et intégration

Nous allons discuter dans ce chapitre des traitements à appliquer aux données géométriques acquises en vue de reconstruire un modèle 3D unifié. En effet, si l'étape de recalage a permis de replacer tous les morceaux de surface fournis par le scanner dans un même repère global, ces morceaux n'en demeurent pas moins déconnectés les uns des autres.

Nous allons donc présenter les premières approches que nous avons envisagées pour l'intégration, mais également le filtrage, de ces différents morceaux de surface. Par «premières approches», nous entendons que ces travaux sont à replacer chronologiquement comme étant les derniers réalisés au cours de cette thèse, et qu'ils ne constituent, à l'heure actuelle, qu'une étude préliminaire de ces problématiques.

Nous allons tout d'abord présenter une technique permettant de raffiner l'alignement trouvé entre les différents morceaux de surface lors de l'étape de recalage. Nous l'avons déjà dit : le recalage calcule une transformation rigide qui tient compte au mieux des données qui lui sont fournies. Cependant, ces données peuvent contenir du bruit de différentes natures :

- un bruit aléatoire, perturbant les coordonnées de chaque point par rapport à ses coordonnées théoriques, et qui constitue le bruit le plus apparent,
- un bruit plus global, lié à des imprécisions de calibrage, et qui tend à déformer légèrement chaque morceau acquis par rapport à la réalité (aplatissement ou bombement).

Ainsi, même lorsque le recalage rigide est excellent, les surfaces ne sont jamais parfaitement alignées. Il suffit d'ailleurs de regarder les résultats obtenus au chapitre précédent par notre recalage global (sans pour autant avoir la prétention d'affirmer que ce recalage est «excellent») pour s'en convaincre.

Nous allons donc commencer par effectuer une correction non rigide de l'alignement, dont le principe est décrit dans la section 1, dans le but d'amener les différentes surfaces à coïncider parfaitement. Cette correction ne constitue pas une approche originale en soi, en ce sens que le principe mis en œuvre ici a déjà été présenté par Neugebauer *et al.* [127, 128] pour effectuer le calcul d'une surface consensuelle dans le cadre d'un algorithme d'intégration.

En revanche, l'originalité de notre approche est d'utiliser le résultat de cette correction pour déterminer automatiquement les paramètres de l'algorithme de filtrage. Nous avons effectivement vu, dans le chapitre 4, que ces algorithmes sont, pour la plupart d'entre eux, des procédés itératifs qui peuvent requérir un certain nombre de paramètres en entrée. La valeur de ces paramètres, ainsi que le nombre d'itérations permettant d'aboutir à un filtrage plus ou moins correct, doivent généralement être fournis manuellement et ne peuvent être correctement estimés que de manière empirique, à l'issue de nombreux essais. L'étape de correction non rigide, quant à elle, tend naturellement à lisser la surface de l'objet par une sorte de moyennage des différents morceaux de surface. Ainsi, là où l'information est redondante, le filtrage est fourni *de facto* par cette correction. Par contre, en dehors des zones de recouvrement, la correction est sans effet. C'est pourquoi nous pensons que le couplage correction/filtrage que nous proposons est pertinent, d'autant plus que nous ne connaissons pas, actuellement, de méthodes s'étant déjà intéressées à la détermination automatique des paramètres de filtres.

L'idée est donc d'effectuer d'abord la correction non rigide, ce qui a pour effet de filtrer le maillage dans les zones de recouvrement. Un algorithme de filtrage est ensuite utilisé pour également débruiter les portions non redondantes de la surface. Le résultat obtenu par la correction précédente pour les zones de recouvrement est alors exploité pour déterminer les bons paramètres du filtre employé.

Nous proposons pour cela une extension d'un filtre existant dans la littérature : le filtre bilatéral. L'objectif de cette extension est de permettre au filtrage de donner un résultat satisfaisant en une unique itération (contrairement au filtre bilatéral original), mais également de réduire le nombre de paramètres qu'il requière à une seule valeur, qui pourra être déduite de l'étape de correction non rigide précitée. La section 2 présente donc un rappel du filtre bilatéral, ainsi que le descriptif de l'extension que nous proposons pour le rendre automatique.

Le dernier point abordé dans ce chapitre est l'intégration, c'est-à-dire la fusion effective des différentes acquisitions géométriques. Cette étape est en fait indissociable du couplage précédent, puisque la correction et le filtrage permettent d'éviter que des artefacts de reconstruction dus à des mauvais alignements entre les surfaces ou au bruit de numérisation n'apparaissent. Nous avons utilisé, pour effectuer cette fusion, un algorithme existant dans la littérature. Nous en parlons dans la section 3.

Nous présentons, dans la section 4, les quelques résultats préliminaires que nous avons obtenus concernant les différents points évoqués, puis nous concluons ce chapitre par la section 5, dans laquelle nous discutons des éléments manquant encore avant que les approches proposées ne deviennent pleinement exploitables.

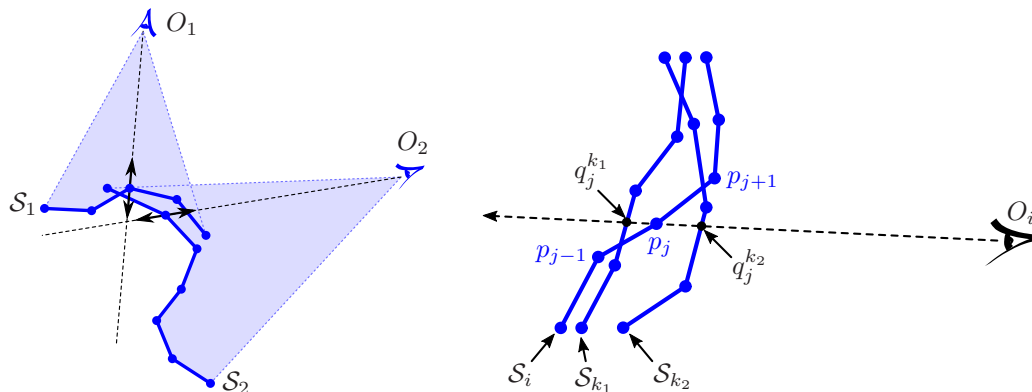


Figure 7.1 – Correction non rigide. **Gauche** : chaque surface est corrigée par rapport à son point de vue d'acquisition, en déplaçant les points le long des droites d'observation. **Droite** : les nouvelles coordonnées des points sont obtenues en cherchant les intersections de ces droites avec toutes les autres surfaces.

1 Correction non rigide des acquisitions géométriques

L'algorithme que nous avons mis en place pour améliorer l'alignement entre les différents morceaux de surface utilise le même principe que celui proposé par Neugebauer *et al.* [127, 128] pour le calcul d'une surface consensuelle, dans le cadre de leur méthode d'intégration. Nous allons détailler ici notre propre implantation de cet algorithme.

■ 1.1 Principe de l'algorithme

Pour les outils de vision effectuant de la mesure de formes, la distribution de l'incertitude n'est généralement pas homogène pour toutes les directions. La mesure consiste, en effet, à déterminer quels sont les rayons, parmi ceux partant du centre optique et traversant chaque pixel, qui intersectent l'objet d'intérêt, ainsi que la distance à laquelle ils l'intersectent. Si le calibrage caméra est bon, la direction de ces rayons d'observation peut être considérée comme fiable. En revanche, la distance à laquelle la surface est estimée le long de ces rayons l'est généralement beaucoup moins, du fait des nombreux paramètres qui peuvent entrer en jeu lors de la mesure. L'incertitude pour de tels dispositifs présente donc une forte anisotropie, orientée majoritairement le long des droites d'observation du capteur.

Notre algorithme tient compte de ce constat en appliquant, à chaque point de mesure nécessitant une correction, un déplacement le long de la droite d'observation qui le traverse. Le principe, illustré par la figure 7.1, est alors de déterminer l'intersection de cette droite avec les autres acquisitions et de déplacer le point vers la moyenne de ces intersections.

Soit $\{S_i\}_{1 \leq i \leq N}$ l'ensemble des N surfaces acquises et recalées. Soit O_i le point de vue (position du centre optique) depuis lequel la surface S_i a été capturée. Pour chaque point p_j

de chaque acquisition \mathcal{S}_i , la correction s'effectue alors de la manière suivante :

1. construire le rayon partant de O_i et orienté vers p_j .
2. chercher les intersections $\{q_j^k\}_{1 \leq k \leq N, k \neq i}$ de ce rayon avec toutes les autres surfaces $\mathcal{S}_k, \forall k \neq i$.
3. déterminer les nouvelles coordonnées p'_j du point p_j par combinaison linéaire des intersections trouvées :

$$p'_j = \frac{w(p_j)p_j + \sum_{k \in \{1, \dots, N\} \setminus \{i\}} w(q_j^k)q_j^k}{w(p_j) + \sum_{k \in \{1, \dots, N\} \setminus \{i\}} w(q_j^k)} \quad (7.1)$$

La fonction de pondération w utilisée ici est introduite dans le but de tenir compte de potentielles sources d'erreurs. En effet, dans les chapitres précédents, nous avons évoqué le fait que les parties de surface acquises à angle rasant ou étant situées à proximité de sauts de profondeur pouvaient être sujettes à une plus grande incertitude. Ainsi, le poids w est défini en chaque point p par le produit de deux fonctions :

$$w(p) = r(p) \cdot s(p) \quad (7.2)$$

avec $r, s : \mathbb{R}^3 \rightarrow [0, 1]$

où $r(p)$ correspond au produit scalaire entre la normale en p et le vecteur en ce même point orienté vers l'observateur, et $s(p)$ tend vers 0 en fonction de la proximité du point p à un saut de profondeur. Le poids w permet donc de donner moins d'importance lors du calcul de la nouvelle position p'_j aux intersections avec les autres surfaces dont la fiabilité est moindre.

Du fait que chaque point utilise les autres pour calculer sa nouvelle position, cette procédure n'est pas séquentielle. Les points ne sont donc mis à jour qu'une fois que tous se sont vus affecter leurs nouvelles coordonnées. Cette procédure de correction est alors répétée autant que nécessaire jusqu'à ce que les différentes surfaces aient convergé l'une vers l'autre. La détection de cette convergence signifie qu'une condition d'arrêt doit être déterminée. Pour l'instant, nous nous sommes contentés de fixer un nombre d'itérations. Un critère basé sur la distance entre p et p' serait certainement plus judicieux, mais ce critère reste encore à déterminer.

■ 1.2 Détails d'implantation

Tel que nous venons de le décrire, cet algorithme considère que chaque rayon intersecte systématiquement toutes les surfaces \mathcal{S}_i . Bien entendu, cela n'est pas forcément le cas dans la pratique. Il se peut même que l'intersection trouvée (si intersection il y a) ne soit pas «compatible» avec le point p_j courant : le but de cette correction étant de rapprocher les surfaces entre elles au niveau des zones de recouvrement, si l'une des intersections est très éloignée du point p_j , il y a de grandes chances pour qu'elle n'appartienne pas à un recouvrement en commun avec le point p_j considéré.

Pour détecter les intersections, nous projetons chaque triangle pour lequel un test d'intersections doit être effectué dans l'espace 2D de l'image télémétrique associée à la surface \mathcal{S}_i

courante. L'ensemble des pixels atteints par cette projection correspond alors aux points 3D pour lesquels le rayon d'observation issu du scanner intersecte le triangle. Seuls ces rayons sont donc considérés, ce qui nous évite une recherche exhaustive.

Étant donné que cette projection dans l'image télémétrique utilise les paramètres caméra du scanner, des imprécisions de calibrage lors de l'estimation de ces paramètres peuvent nous faire manquer certaines intersections, du fait que le triangle ne se projette pas exactement là où il devrait. Nous étendons alors la zone de recherche à une bordure de quelques pixels de large autour de la projection afin d'éviter d'éventuels problèmes liés à ces imprécisions.

Nous détectons alors les intersections incompatibles à l'aide de deux tests :

- l'intersection doit être orientée dans le même sens que p_j (produit scalaire de leurs normales ≥ 0),
- la distance qui sépare l'intersection du point p_j doit être inférieure à un seuil ϵ_c .

Dans notre application, ce seuil est automatiquement calculé à partir des données, et correspond à la distance moyenne d'un point à son 8-voisinage dans l'image télémétrique associée à la surface \mathcal{S}_i courante. Les intersections inexistantes ou étant jugées incompatibles sont alors éliminées des calculs précédents (ce qui revient à leur assigner une pondération $w = 0$).

Le résultat de cette correction non rigide est alors d'attirer progressivement les différentes surfaces l'une vers l'autre dans les zones de recouvrement. La façon dont ce résultat est obtenu revient à effectuer un moyennage de plusieurs mesures d'une même surface. Or, le bruit étant de nature aléatoire, ce moyennage tend à le faire disparaître du signal mesuré. L'effet observé à l'issue de cette correction est alors un filtrage partiel, où les parties de la surface se trouvant dans les zones de recouvrement sont automatiquement débruitées grâce à la redondance des données. Ce résultat sera illustré ultérieurement, dans la section 4.

Étant donné que le résultat de la correction nous est utile pour la suite, nous notons désormais \mathcal{S}_c la surface obtenue à partir de la surface \mathcal{S}_i à la suite de cette correction.

2 Automatisation du processus de filtrage

Cette section détaille les choix que nous avons faits pour l'automatisation du processus de filtrage en lui-même. Nous commençons par rappeler le principe du filtre bilatéral en section 2.1, avant de décrire, dans la section 2.2, la nouvelle extension de ce filtre que nous avons proposée, visant à réduire le nombre de paramètres requis à une unique valeur. La section 2.3 présente finalement comment nous exploitons le résultat donné par la correction non rigide décrite dans la section précédente pour estimer judicieusement la valeur de ce seul paramètre.

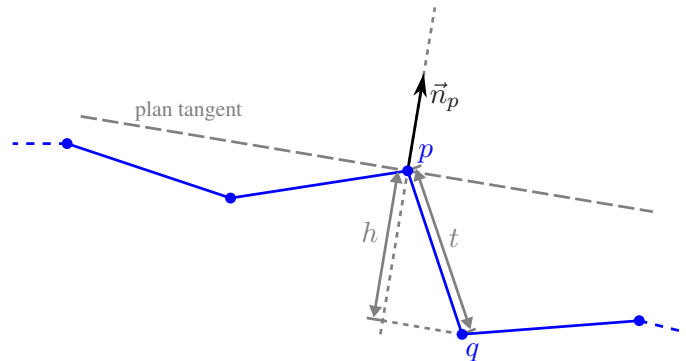


Figure 7.2 – Le filtre bilatéral correspond à la composition de deux filtres Gaussiens, respectivement appliqués à la distance Euclidienne t d'un point p à son voisinage et à la distance h de ce voisinage au plan tangent en p .

■ 2.1 Rappel sur le filtre bilatéral

Le filtre bilatéral [47] est un filtre adaptatif, c'est-à-dire qu'il a été conçu pour lisser la surface tout en préservant les caractéristiques géométriques qu'elle peut présenter. Il s'agit d'un filtre à puissance variable¹, consistant en la combinaison de deux filtres Gaussiens :

- le premier tient compte de la distance Euclidienne d'un point p à chacun de ses voisins directs, ce qui permet de donner moins de poids à ceux de ces voisins qui sont les plus éloignés, à l'instar d'un filtrage Gaussien standard.
- le second est appliqué à la distance de ce même voisinage au plan tangent du point p , ce qui permet d'atténuer l'impact sur le filtrage des points qui sont trop éloignés de l'approximation locale de la surface que constitue ce plan. C'est ce second filtre Gaussien qui permet de préserver les caractéristiques de la géométrie et qui fait du filtre bilatéral un filtre adaptatif.

Ce filtre travaille donc directement sur les sommets du maillage, et considère, pour calculer la distance au plan tangent, que chacun d'eux est muni d'un vecteur normal. Soient un sommet quelconque p à filtrer, \mathcal{N}_p l'ensemble des points voisins de p (reliés directement à p par une arête), et \vec{n}_p la normale en p . Les nouvelles coordonnées p' du point p sont alors obtenues par un déplacement le long de sa normale, d'une distance donnée par la combinaison des deux filtres Gaussiens, comme le définit l'équation suivante :

$$p' = p + \frac{\sum_{q \in \mathcal{N}_p} G_{\sigma_t}(t) G_{\sigma_h}(|h|) h}{\sum_{q \in \mathcal{N}_p} G_{\sigma_t}(t) G_{\sigma_h}(|h|)} \vec{n}_p \quad (7.3)$$

où G_σ est la fonction Gaussienne de variance σ , $t = \|\vec{pq}\|$ est la distance Euclidienne entre p et q , et $h = \vec{pq} \cdot \vec{n}_p$ est la distance de q au plan tangent en p , comme on peut le voir sur la figure 7.2. σ_t et σ_h sont les paramètres du filtre, représentant les variances respectives des deux filtres Gaussiens.

Du fait que seul le voisinage immédiat soit considéré, plusieurs applications successives de ce filtre sont généralement nécessaires pour obtenir un niveau de filtrage suffisant (dispari-

¹cf. chapitre 4, section 2.

tion totale du bruit). Ce filtre compte donc, en tout, trois paramètres :

- la variance σ_t pour la fonction Gaussienne appliquée à la distance Euclidienne,
- la variance σ_h pour la fonction Gaussienne appliquée à la distance au plan tangent,
- le nombre d'itérations nécessaires à l'obtention d'un filtrage suffisant.

Les deux premiers paramètres sont généralement obtenus par tâtonnement, alors que le dernier est estimé visuellement, en stoppant le processus itératif lorsque l'utilisateur juge que la surface est suffisamment lissée à son goût.

Notre objectif est donc de proposer une extension du filtre bilatéral qui permette d'automatiser la détermination de ces paramètres.

■ 2.2 Filtre bilatéral non itératif à un seul paramètre

Le principe de notre nouvel algorithme de filtrage est d'utiliser un masque pour le filtre bilatéral qui ne soit pas uniquement restreint au voisinage immédiat du point p à filtrer. Nous allons effectivement proposer une construction de ce masque permettant d'en étendre les dimensions tant que les points qui s'y trouvent respectent un certain critère de proximité par rapport à une estimation locale de la surface en p . Ainsi, plus le masque sera grand, plus l'effet du filtrage sera important, éliminant ainsi la nécessité d'appliquer le filtre en plusieurs itérations.

Le critère de proximité que nous proposons est le suivant : si un point quelconque de la surface se situe à une distance au plan tangent de p inférieure à un seuil δ_h donné, il est considéré comme *admissible* pour le filtrage. Dans le cas contraire, il est rejeté. Compte tenu de cette définition, notre masque est construit en deux étapes :

1. **Construction du masque préliminaire** : les points voisins de p sont progressivement sélectionnés par croissance de région, en ne considérant pour cela que les points admissibles (figure 7.3-a).
2. **Construction du masque final** : parmi les points définissant la bordure de la région obtenue à l'étape précédente, celui situé le plus près de p est utilisé pour définir le rayon δ_t d'une sphère centré en p . Le masque final \mathcal{M}_p associé au point p est alors obtenu en ne gardant que les points contenus dans cette sphère (figure 7.3-b).

La sphère centrée en p et de rayon δ_t définit la plus petite région sphérique à l'intérieur de laquelle tous les points sont admissibles, c'est-à-dire considérés comme suffisamment proches de l'estimation locale de la surface pour pouvoir être incorporés au processus de filtrage. Ainsi, pour les régions planes, le masque pourra s'étendre de manière conséquente avant qu'un point ne viole le critère de proximité, rendant l'effet du filtrage d'autant plus prononcé. À l'inverse, pour les zones de fortes courbures, le masque sera très petit, évitant ainsi d'appliquer le filtre sur des zones pouvant représenter des caractéristiques géométriques. Cette adaptativité du masque est illustrée sur la figure 7.4.

Une fois le masque défini, le filtre bilatéral est appliqué en tenant compte de l'ensemble des points qu'il contient. Concernant la fonction Gaussienne G_σ , nous avons utilisé le noyau K_σ de type Gaussien à support fini proposé par Ohtake *et al.* [135] et défini de la manière

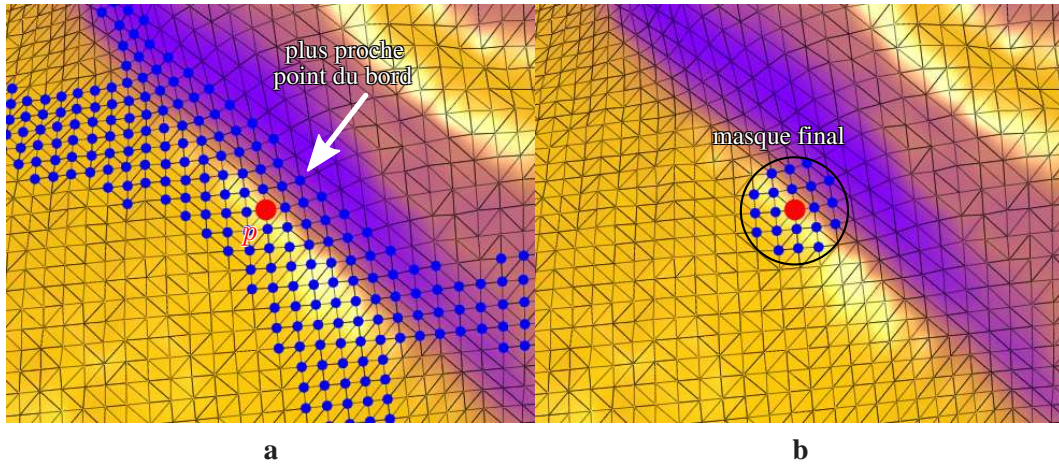


Figure 7.3 – Construction du masque pour notre filtre bilatéral. a) Construction d'un masque préliminaire par croissance de région sur les points situés à une distance inférieure à δ_h du plan tangent en p . b) Le masque final correspond alors à la sphère centrée en p et incidente au plus proche point du bord du masque préliminaire.

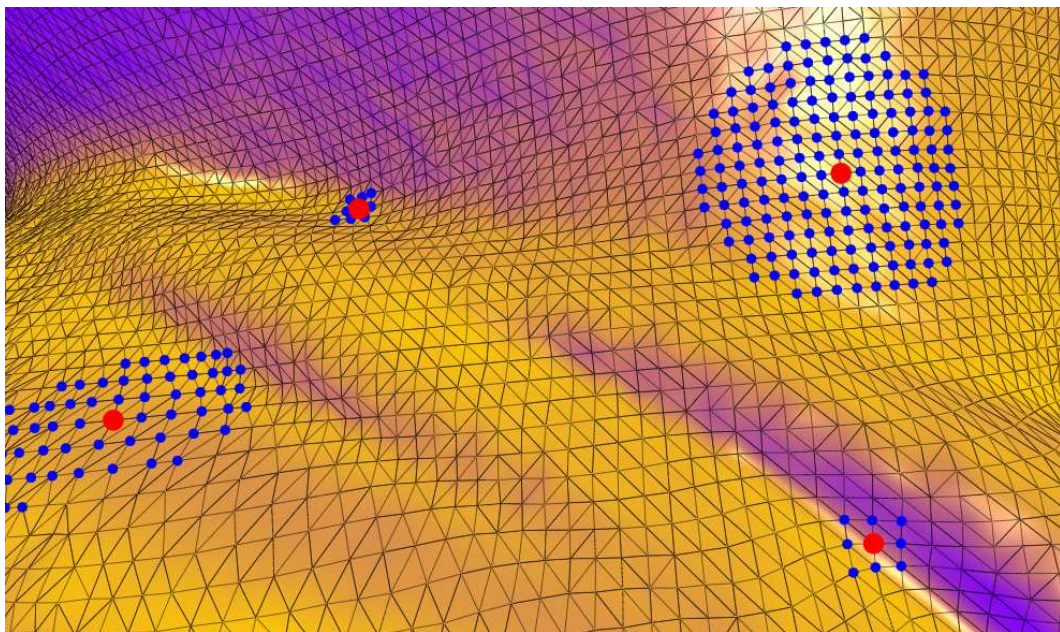


Figure 7.4 – Adaptativité du masque proposé : le rayon du masque devient plus petit dans les zones de fortes courbures, et plus grand dans les zones planes.

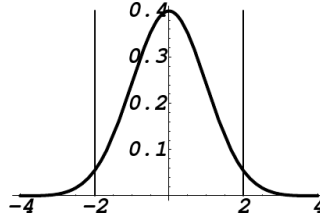


Figure 7.5 – Noyau K_σ de type gaussien à support fini. Courbe obtenue pour $\sigma = 1$. (Image tirée de [135]).

suivante :

$$K_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \begin{cases} e^{-\frac{x^2}{2\sigma^2}} & \text{si } |x| \leq 2\sigma \\ \frac{1}{16e^2} \left(4 - \frac{|x|}{\sigma}\right)^4 & \text{si } 2\sigma < |x| \leq 4\sigma \\ 0 & \text{si } |x| > 4\sigma \end{cases} \quad (7.4)$$

L'allure de ce noyau est illustrée sur la figure 7.5.

Ce noyau possède un rayon équivalent à quatre fois le paramètre de variance σ . Or, la façon dont nous construisons le masque \mathcal{M}_p nous assure que tous les points qu'il contient se trouvent à une distance au plan tangent en p inférieure à δ_h et à une distance Euclidienne au point p inférieure à δ_t . Nous pouvons donc paramétrer les deux filtres Gaussiens utilisés par le filtre bilatéral de manière à ce que leurs rayons respectifs tiennent compte des deux distances δ_t et δ_h :

$$\sigma_t = \frac{\delta_t}{4} \quad \text{et} \quad \sigma_h = \frac{\delta_h}{4} \quad (7.5)$$

Les nouvelles coordonnées p' obtenues par filtrage du point p sont alors déterminées grâce à l'équation suivante :

$$p' = p + \frac{\sum_{q \in \mathcal{M}_p} K_{\frac{\delta_t}{4}}(t) K_{\frac{\delta_h}{4}}(|h|) h}{\sum_{q \in \mathcal{M}_p} K_{\frac{\delta_t}{4}}(t) K_{\frac{\delta_h}{4}}(|h|)} \vec{n}_p \quad (7.6)$$

où t et h sont définis comme précédemment.

On peut donc voir que le seul paramètre demandé à l'utilisateur est la distance δ_h , la seconde distance δ_t étant estimée automatiquement lors de la construction du masque \mathcal{M}_p . De plus, le fait que les dimensions du masque s'adaptent aux variations géométriques de la surface permet d'obtenir un filtrage suffisant à l'issue d'une seule itération. Cependant, ce résultat dépend fortement de la valeur δ_h fournie en entrée, comme nous le montrerons plus tard (section 4).

Concernant les plans tangents, ils sont définis en chaque point à partir de sa position et de sa normale. Or, la normale est une donnée extrêmement sensible au bruit. Dans le but de rendre l'estimation des plans tangents plus robuste, nous commençons par appliquer un filtre moyenneur sur les normales, en considérant le voisinage immédiat de chaque point uniquement. Cela permet de les rendre moins sensibles au bruit. Il est à noter que seules les normales sont filtrées lors de cette étape préliminaire : les positions ne sont modifiées que par le filtrage bilatéral en lui-même.

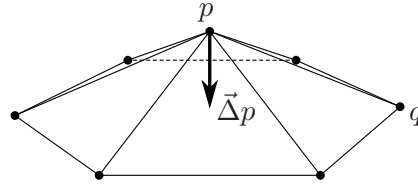


Figure 7.6 – Opérateur Laplacien en un sommet p .

■ 2.3 Estimation automatique de l'unique paramètre

Maintenant que nous avons reformulé le filtre bilatéral de manière à ce qu'il ne nécessite plus qu'un seul paramètre δ_h , nous souhaitons estimer la valeur de ce paramètre de manière automatique et uniquement à partir des données. Pour ce faire, nous allons utiliser la surface \mathcal{S}_c préalablement obtenue par la correction non rigide décrite dans la section 1. Comme nous l'avons dit précédemment, cette surface représente un filtrage partiel de la surface initiale \mathcal{S}_i dont elle est issue, c'est-à-dire que les régions de \mathcal{S}_c situées dans les zones de recouvrement ont été débruitées par cette correction. Ces régions peuvent donc être utilisées comme vérité terrain pour définir ce à quoi doit ressembler la surface \mathcal{S}_i une fois filtrée.

Notre souhaitons donc exprimer l'estimation du paramètre δ_h sous la forme d'un problème d'optimisation, en définissant une fonction représentant l'erreur commise par le filtrage par rapport à cette vérité terrain en fonction de la valeur du paramètre. Nous allons commencer par définir une fonction représentant cette erreur en chaque sommet du maillage.

Notons \mathcal{S}_f^x la surface obtenue à partir de \mathcal{S}_i après application de notre filtre en prenant comme paramètre $\delta_h = x$. Étant donné que \mathcal{S}_c et \mathcal{S}_f^x sont toutes deux construites à partir de \mathcal{S}_i sans supprimer aucun sommet ni modifier aucune relation d'adjacence, un point p donné de l'une de ces trois surfaces a nécessairement un homologue dans chacune des deux autres. De la même manière, les points définissant son voisinage restent les mêmes pour chacune de ces surfaces.

Soient $\vec{\Delta}_c p$ et $\vec{\Delta}_f^x p$ les opérateurs Laplaciens calculés au point p pour chacune des deux surfaces \mathcal{S}_c et \mathcal{S}_f^x , respectivement. À titre de rappel, cet opérateur est défini en chaque sommet p d'un maillage par un vecteur calculé à l'aide de l'équation suivante :

$$\vec{\Delta} p = \frac{1}{\text{card}(\mathcal{N}_p)} \sum_{q \in \mathcal{N}_p} q - p \quad (7.7)$$

où \mathcal{N}_p est l'ensemble des points partageant une arête avec p (ensemble des points de l'ombrelle de p), comme l'illustre la figure 7.6.

De manière générale, plus un point est filtré, plus son ombrelle est aplatie, et donc plus la norme de son Laplacien est petite. Si l'on considère la vérité terrain représentée par \mathcal{S}_c , l'erreur relative commise par l'application de notre filtre pour un paramètre de valeur $\delta_h = x$ peut être mesurée au point p par la fonction ϵ suivante :

$$\epsilon(p, x) = \left\| \Delta_c p - \Delta_f^x p \right\| \quad (7.8)$$

Cette fonction prend une valeur très faible lorsque les ombrelles de p dans \mathcal{S}_c et \mathcal{S}_f^x présentent des convexités similaires. En revanche, la valeur de la fonction augmente lorsque la convexité de l'une des deux ombrelles augmente par rapport à celle de l'autre. Ainsi, la fonction ϵ permet de détecter aussi bien le sous-filtrage que le sur-filtrage de p par rapport à la vérité terrain :

- si $\|\Delta_c p\| < \|\Delta_f^x p\|$, cela signifie que l'ombrelle représentant l'objectif à atteindre est plus aplatie que l'ombrelle obtenue grâce au filtre. Il s'agit alors d'un cas de sous-filtrage, ce qui signifie que la valeur x pour le paramètre δ_h est trop petite.
- à l'inverse, si $\|\Delta_c p\| > \|\Delta_f^x p\|$, nous sommes dans un cas de sur-filtrage, ce qui veut dire que la valeur de x est cette fois trop grande.

Bien entendu, notre objectif n'est pas d'optimiser le paramètre δ_h en chaque sommet du maillage. Cela n'aurait d'ailleurs aucun sens, dans la mesure où l'effet de la correction non rigide n'est pas homogène sur toute la surface. Nous voulons donc construire une fonction d'erreur globale \mathcal{E} permettant d'estimer le meilleur paramètre pour débruiter l'ensemble de la surface initiale \mathcal{S}_i . Nous définissons cette fonction de la manière suivante :

$$\mathcal{E}(x) = \frac{\sum_{p \in \mathcal{S}_i} w(p) \epsilon(p, x)}{\sum_{p \in \mathcal{S}_i} w(p)} \quad (7.9)$$

où les poids w sont donnés par $w(p) = \frac{1}{\|\Delta_c p\|}$, ce qui signifie que plus l'ombrelle d'un point est plane dans \mathcal{S}_c , plus le point considéré a de l'importance dans le calcul de la fonction \mathcal{E} . L'intérêt de définir les poids de cette manière est double :

- les ombrelles les plus planes de \mathcal{S}_c correspondent aux points les plus filtrés par la correction non rigide. Il est donc normal de donner à ces points plus de poids, dans la mesure où ce sont eux qui portent l'information la plus représentative.
- les ombrelles les plus bombées, quant à elles, ne représentent pas uniquement les points les moins filtrés par la correction. Elles peuvent également correspondre à des points situés sur des caractéristiques géométriques. Or, ces caractéristiques sont plus sujettes à l'incertitude que les zones planes (masquage, interreflexion des franges projetées par le scanner, etc.) et la correction peut aboutir à une modification drastique de la géométrie pour ces points. Leur donner moins de poids est donc légitime.

Le fonction d'erreur \mathcal{E} atteint donc son minimum lorsque le filtrage produit un résultat proche de celui de la correction non rigide pour les zones les plus significatives (*ie.* les plus débruitées par la correction et les moins sujettes à l'incertitude).

La valeur du paramètre δ_h est finalement déterminée en recherchant ce minimum. Dans notre implantation, nous avons utilisé une recherche dichotomique, en prenant comme intervalle de départ $[0, 2 \max_{p \in \mathcal{S}_i} \Delta p]$. Cette méthode de recherche simpliste présuppose que la fonction \mathcal{E} n'admet qu'un seul minimum dans cet intervalle. Bien que nous ayons développé cette fonction dans ce but, aucune preuve n'a pour le moment été donnée à ce sujet. Nous nous sommes uniquement basés sur l'expérimentation. Il est cependant évident que la méthode de recherche du minimum de \mathcal{E} doit être revue.

Le filtrage peut alors être appliqué aux surfaces avant de les intégrer. Il faut noter que le filtre peut être appliqué soit à \mathcal{S}_i , soit à \mathcal{S}_c . Le premier cas signifie que la correction non rigide n'est utilisée que pour estimer les paramètres du filtre puis ignorée dans la suite des traite-

ments, alors que le second cas conserve les modifications de la correction.

3 Intégration

Une fois le processus de filtrage effectué, nous utilisons l'algorithme VRIP (*Volumetric Range Image Processing*) pour effectuer la fusion des différentes images télémétriques. Il s'agit d'une méthode d'intégration volumique, qui effectue la fusion à l'aide de champs de distances signés définis dans des grilles de voxels. Pour plus de détails, nous invitons le lecteur à consulter l'article de Curless *et al.* [34] décrivant cet algorithme.

Pour optimiser la construction des champs de distances, ceux-ci ne sont définis que dans une bande de quelques voxels d'épaisseur autour des surfaces qu'ils représentent. De plus, pour éviter l'explosion mémoire liée à l'utilisation de grilles 3D, nous avons opté pour une implantation hors mémoire de cet algorithme.

Il est à noter que VRIP peut être étendu de manière à boucher automatiquement les trous provenant d'une numérisation géométrique incomplète. Nous n'avons pour le moment pas intégré cette extension.

4 Résultats préliminaires

■ 4.1 Correction non rigide

L'algorithme de correction non rigide que nous avons employé revient en fait à moyennner plusieurs mesures d'une même surface. Le bruit étant de nature aléatoire, ce moyennage a pour effet d'en réduire l'influence et de faire tendre la surface résultante vers la surface théorique idéale. En conséquence, l'effet du filtrage induit par la correction non rigide est forcément corrélé au nombre d'acquisitions présentes dans la zone de recouvrement traitée : plus le nombre de mesures augmente, plus le signal se distingue du bruit. Il suffit de regarder l'image 7.7 pour s'en convaincre. On constate en effet une réduction progressive du bruit à mesure que le nombre des acquisitions prises en compte augmente.

Ce constat est quelque peu problématique en ce qui concerne l'utilisation du résultat obtenu par cette correction comme donnée d'entrée pour l'estimation des paramètres du filtre géométrique : l'amplitude de bruit estimée dépendra forcément du nombre de surfaces combinées lors de la correction.

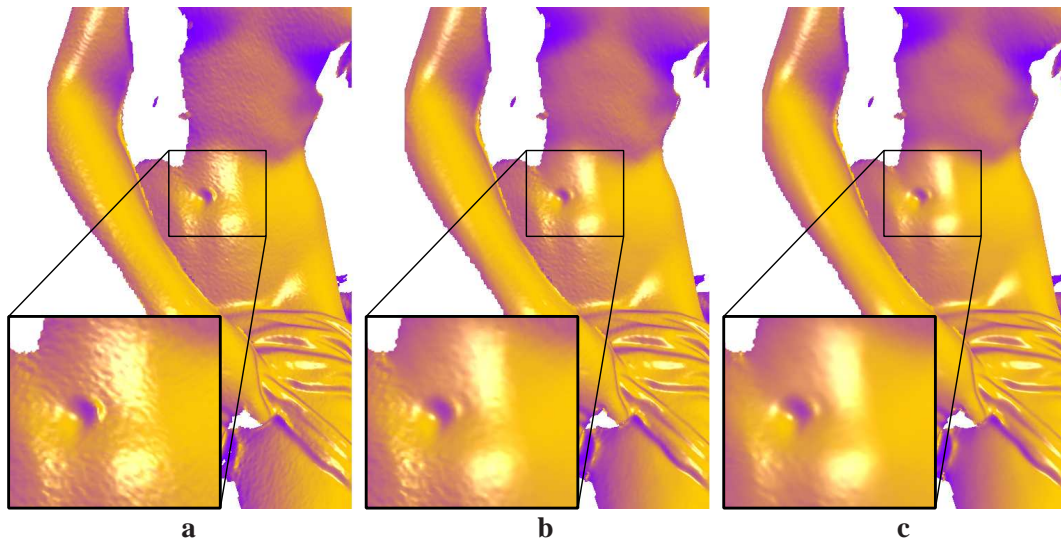


Figure 7.7 – Résultat de la correction non rigide en fonction du nombre d'acquisitions présentes dans la zone de recouvrement. **a)** Surface initiale, non corrigée. **b)** Deux acquisitions dans la zone de recouvrement. **c)** Quatre acquisitions. (5 itérations sont appliquées dans chacun des cas).

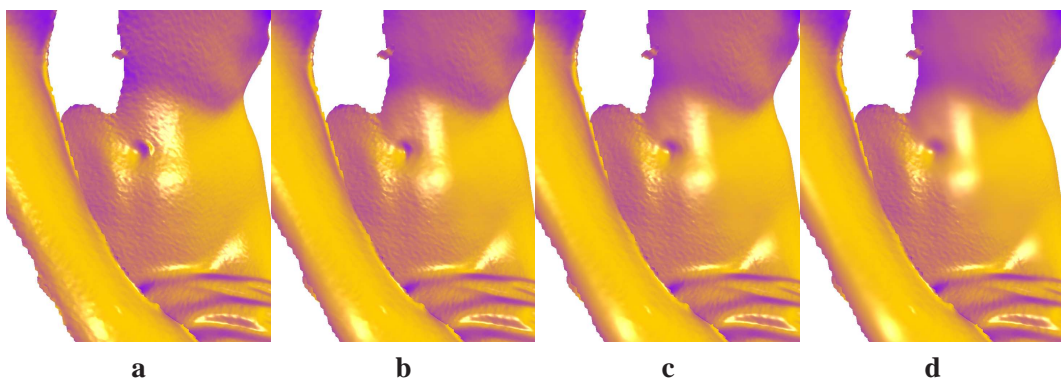


Figure 7.8 – Résultat de la correction non rigide en fonction du nombre d'itérations. **a)** Surface initiale, non corrigée. **b)** 1 itération. **c)** 3 itérations. **d)** 15 itérations. (recouvrement de trois surfaces dans chacun des cas).

Cette correction non rigide est un processus itératif. À ce titre, la précision dépend également du nombre n d'itérations. La figure 7.8 illustre les résultats obtenus pour différentes valeurs de n . On peut d'ailleurs clairement voir, sur cette image, que seules certaines parties de la surface sont corrigées : ces régions correspondent aux zones de recouvrement avec d'autres surfaces. Il est d'ailleurs à noter que le nombre de surfaces présentes dans la zone de recouvrement est seulement de trois dans cet exemple, et le lissage obtenu au bout de 15 itérations est déjà très bon. Ainsi, le constat que nous venons de faire sur la dépendance de l'amplitude de bruit estimée par rapport au nombre de surfaces considérées n'est, *in fine*, pas si problématique, à condition que le nombre d'itérations soit suffisant pour que la correction arrive à convergence. Le temps nécessaire pour effectuer ces 15 itérations sur trois surfaces est de 6 minutes 34 secondes dans l'état actuel de l'implantation.

En plus d'un lissage progressif, on constate également la disparition d'artefacts provenant d'erreurs de mesure. Par exemple, la crête apparaissant sur le rebord droit du nombril n'existe pas sur l'objet réel : elle provient du fait que, depuis ce point de vue, les interférences se produisant à l'intérieur même du nombril ont perturbé l'analyse des franges lumineuses lors du calcul de géométrie par le scanner. Pour d'autres points de vue, cette crête n'apparaît en effet pas au même endroit (voire n'apparaît pas du tout). Le fait de combiner plusieurs mesures de la même surface permet donc d'atténuer ces artefacts.

■ 4.2 Étude du filtre proposé et de son estimation automatique

La figure 7.9 illustre le résultat de notre filtre pour différentes valeurs du paramètre δ_h . Pour les deux valeurs les plus faibles (0,1mm et 0,2mm), on observe un net sous-filtrage dans la grande région plane du ventre ou sur la main. À l'inverse, pour les valeurs les plus grandes (0,6mm et 1mm), c'est un phénomène de sur-filtrage que l'on observe par rapport à la surface initiale. Pour la surface filtrée avec $\delta_h = 0,6\text{mm}$, ce phénomène est moins flagrant, mais il demeure tout de même notable sur certaines caractéristiques fines, comme celles mises en valeur par les flèches. Pour la valeur $\delta_h = 0,4\text{mm}$, en revanche, on constate que le filtrage est bon dans les zones planes, et que les caractéristiques sont plutôt bien préservées dans leur ensemble. Par ailleurs, ces images ont été obtenues en appliquant le filtre directement sur la surface initiale et non sur celle résultant de la correction non rigide.

Lors des estimations manuelles que nous avons faites pour produire ces images, nous avons constaté que les meilleurs résultats visuels pour cet exemple étaient obtenus pour une valeur de δ_h située entre 0,4mm et 0,5mm. Or, si l'on regarde le graphique de la figure 7.10, qui illustre l'évolution de la fonction d'erreur développée pour l'optimisation de δ_h , on peut voir que le minimum de cette fonction se situe justement dans cet intervalle de valeurs. La recherche automatique de ce minimum a d'ailleurs abouti à $\delta_h = 0.4017\text{mm}$. L'estimation manuelle vient donc corroborer les résultats obtenus par notre méthode.

Du fait de la construction du masque par croissance de région, le temps nécessaire à notre filtre pour débruiter la surface dépend du paramètre δ_h : plus ce paramètre est grand, plus le nombre de points à ajouter au masque et à traiter est important. Ainsi, dans l'exemple présenté, le filtrage s'effectue en 3,2 secondes pour $\delta_h = 0.25\text{mm}$, 6,6 secondes pour

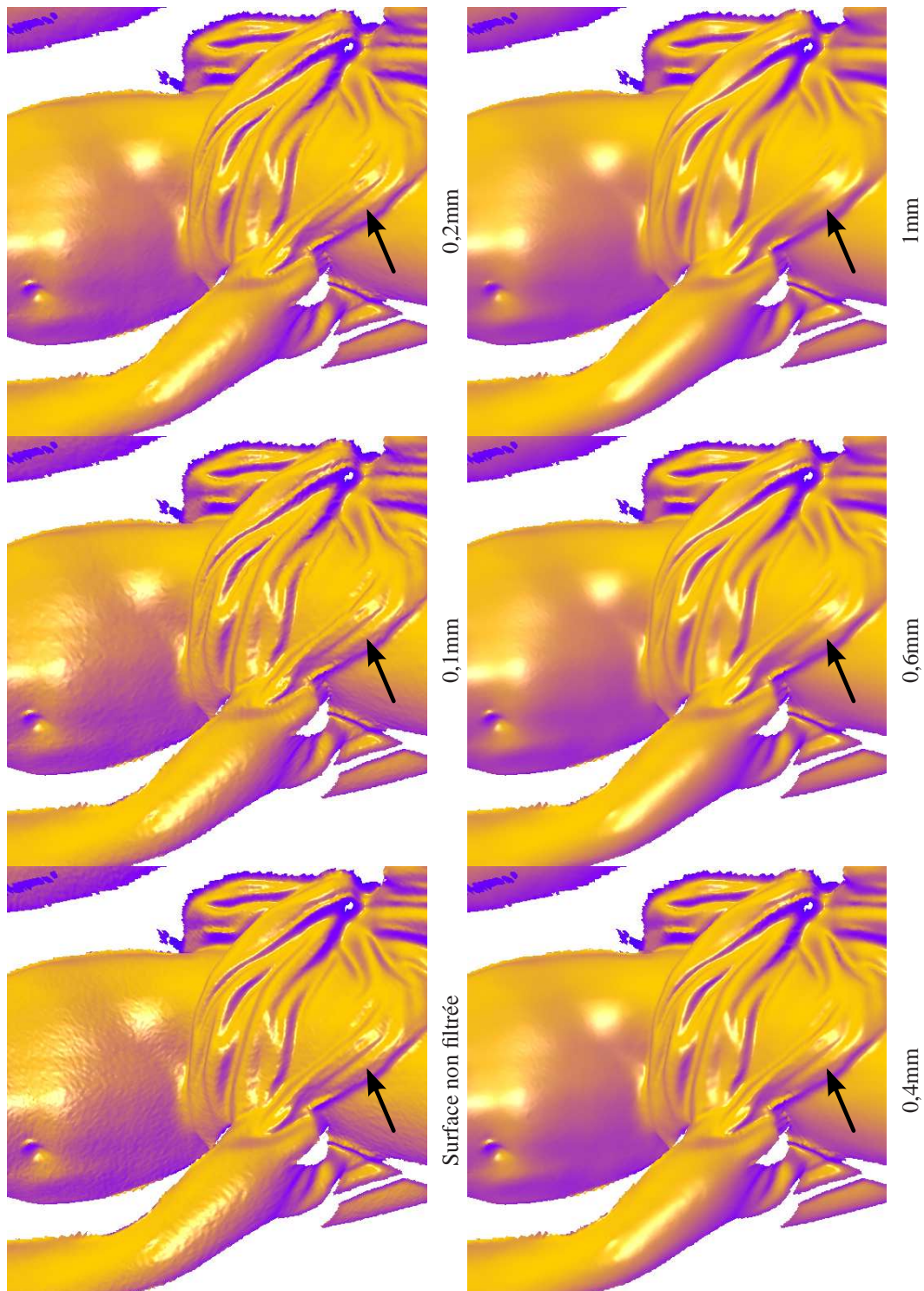


Figure 7.9 – Résultats de notre filtre pour différentes valeurs de δ_h (une seule itération).

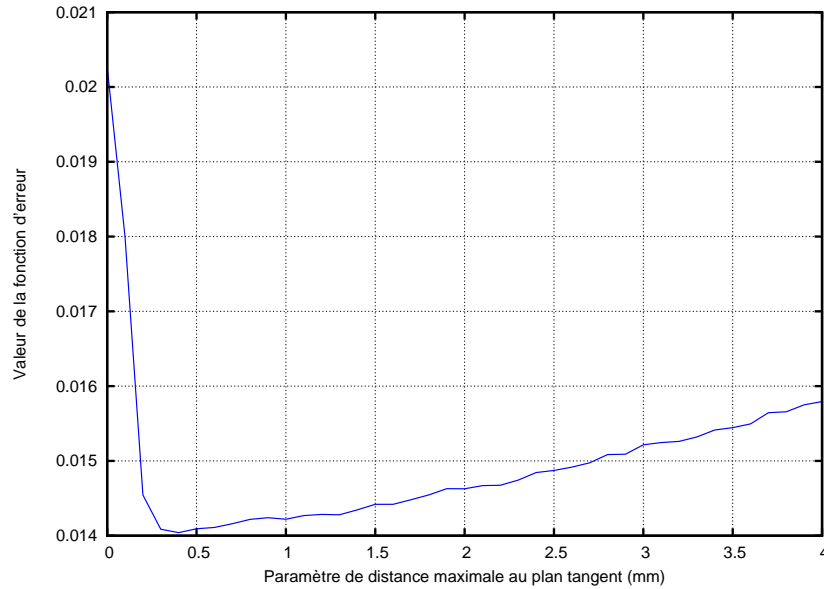


Figure 7.10 – Allure de la fonction d'erreur utilisée pour optimiser le paramètre δ_h .

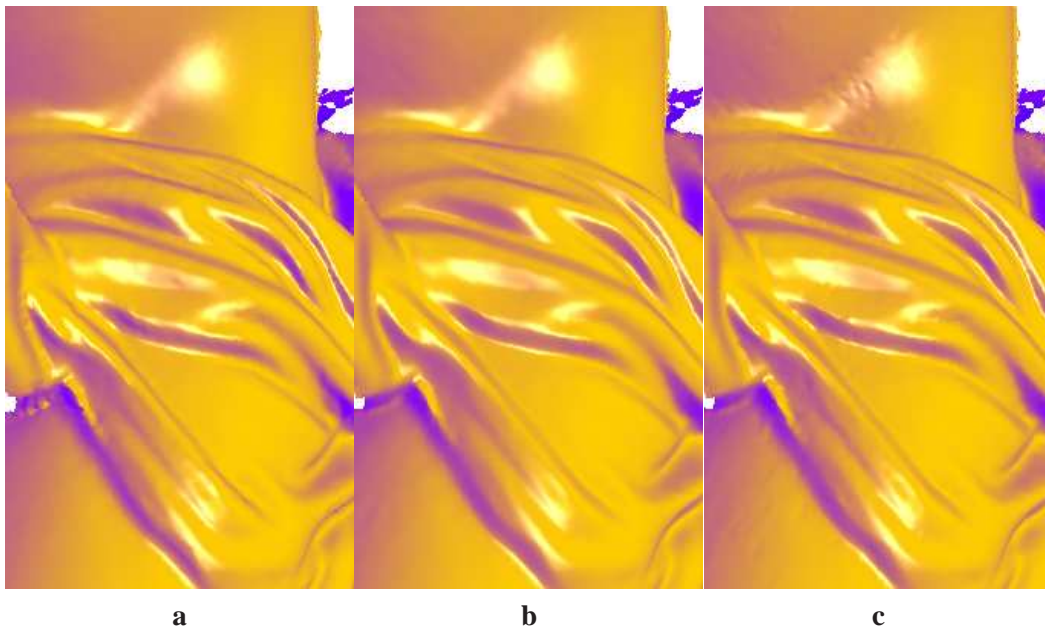


Figure 7.11 – Comparaison de notre filtre au filtre bilatéral. **a)** Résultat obtenu avec notre filtre pour le paramètre déterminé de manière automatique. **b)** Meilleur résultat obtenu avec le filtre bilatéral. Un léger sur-filtrage peut être constaté au niveau des caractéristiques géométriques par rapport à notre méthode. **c)** Tentative d'amointrissement du surfiltrage précédent. Des artefacts apparaissent, du fait que du bruit soit parfois lui aussi considéré comme des caractéristiques à conserver.

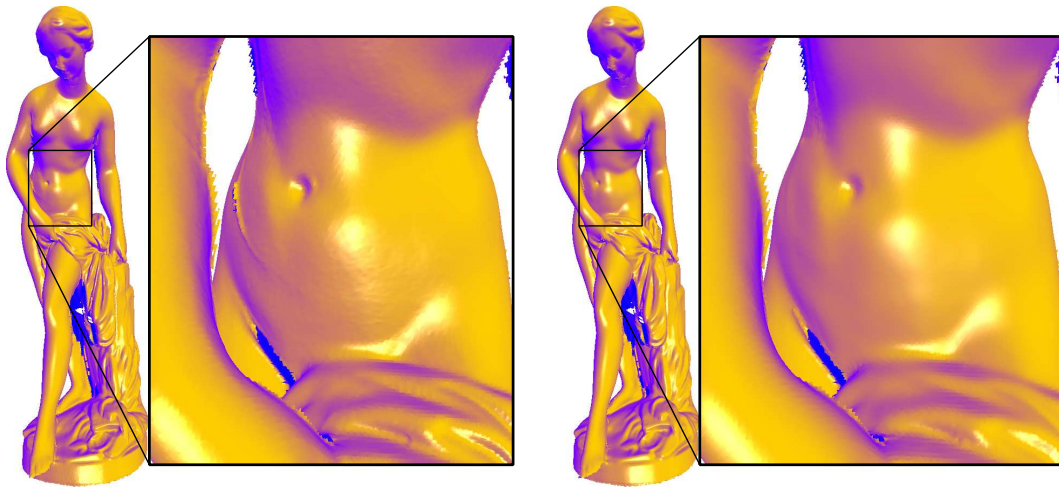


Figure 7.12 – Reconstruction d'un modèle 3D par intégration des différentes acquisitions. **Gauche** : intégration directement après recalage. **Droite** : intégration précédée d'une correction non rigide.

$\delta_h = 0.5\text{mm}$ et 12,8 secondes pour $\delta_h = 1\text{mm}$. À titre de comparaison, le filtre bilatéral standard met 0,15 seconde pour accomplir une itération.

Nous nous sommes également comparés au filtre bilatéral original au niveau visuel, mais nous ne sommes pas parvenus à déterminer des paramètres permettant d'obtenir une qualité comparable à ce que qui a été fourni par notre méthode. La figure 7.11 illustre ce constat : l'image de gauche représente le résultat de notre filtre et l'image du milieu le meilleur résultat que nous avons obtenu pour le filtre bilatéral original. On peut observer qu'un léger sur-filtrage apparaît au niveau des caractéristiques géométriques de la surface. Nous avons tenté de remédier à cela en diminuant la variance σ_h réglant la puissance du filtre Gaussien appliqué à la distance au plan tangent, mais cela a fait apparaître des artefacts, devenant de plus en plus prononcés à mesure que le nombre d'itérations augmentait (image de droite). Ces artefacts sont apparus du fait que la valeur trop faible de σ_h a entraîné une reconnaissance de caractéristiques géométriques par le filtre bilatéral là où il n'y avait en réalité que du bruit de numérisation.

■ 4.3 Résultats obtenus après intégration

La figure 7.12 montre deux maillages obtenus par intégration à l'aide de l'algorithme VRIP. Dans le premier cas (figure 7.12-gauche), l'intégration a été effectuée sans appliquer de correction non rigide au préalable. Dans le second cas (figure 7.12-droite), en revanche, cette correction a été effectuée.

Nous avons souhaité présenter ces deux images pour illustrer l'intérêt qu'apporte la correction non rigide à l'intégration. A priori, l'objectif de cette correction est de calculer une surface consensuelle dans les zones de recouvrement. Or, l'algorithme VRIP, comme tous

les algorithmes d'intégration basés sur des champs de distances, est censé déterminer lui-même cette surface consensuelle lors de la combinaison des champs de distances associés aux différentes acquisitions. On peut cependant voir sur la figure de gauche que cela ne suffit pas à éliminer tous les artefacts, contrairement au cas où la correction non rigide a été préalablement exécutée. On peut également constater que la surface est plus lisse dans ce second cas, du fait du débruitage occasionné par la correction. Le filtre n'a, quant à lui, pas été utilisé pour cet exemple.

5 Bilan

Comme nous l'avons souligné dans le préambule de ce chapitre, ces travaux n'en sont encore qu'à un stade embryonnaire, aussi de nombreux points restent encore à développer et à étudier.

Concernant la correction non rigide, nous avons pu constater qu'elle présentait un intérêt indéniable, et ce pas seulement pour l'estimation des paramètres de filtrage : l'intégration s'en trouve également améliorée. Ce résultat n'est pas surprenant, dans la mesure où cette correction a été initialement développée pour effectuer le calcul d'une surface consensuelle lors d'un algorithme d'intégration [127, 128]. Il s'agit cependant d'un processus très coûteux : pour chaque point de chaque acquisition, une recherche d'intersections rayon/surface doit être effectuée avec toutes les autres acquisitions, et ce pour un certain nombre d'itérations. Inutile de préciser que, dans de telles conditions, les temps de calcul deviennent très longs lorsque le nombre d'acquisitions augmente.

Concernant le filtrage, nous n'avons, pour le moment, pas eu l'occasion de comparer le filtre que nous avons développé à d'autres filtres existant dans la littérature. Cette étape est cependant indispensable pour en vérifier la validité par rapport à ce domaine de recherche. Notamment, il serait intéressant de nous comparer à d'autres filtres non itératifs [87, 135].

D'après ses auteurs, l'un des problèmes du filtre bilatéral est le phénomène de rétrécissement subi par les objets filtrés à mesure que le nombre d'itérations augmente. Étant donné que notre algorithme est basé sur ce dernier, il est certain qu'il en souffre également. Cependant, nous avons développé cet algorithme dans le but de ne pas avoir à effectuer plus d'une itération. Ce défaut de rétrécissement n'est donc pas problématique dans notre cas, dans la mesure où son impact n'est pas pénalisant à l'issue d'une unique itération.

Un autre problème vient du fait que, pour le moment, notre technique estime et applique le filtre de manière indépendante à chaque acquisition, risquant d'aboutir à un filtrage non homogène de la surface une fois ces acquisitions intégrées. Il serait donc préférable de trouver une solution pour estimer le bruit de manière globale, c'est-à-dire de façon à pouvoir l'appliquer après l'intégration, une fois l'objet reconstruit.

Pour ce qui est de l'estimation automatique des paramètres du filtre, on peut voir que nous sommes passés des trois paramètres du filtre bilatéral à un unique paramètre. Celui-ci est alors estimé à partir du résultat fourni par la correction non rigide. Cependant, cette correction est un processus itératif, nécessitant un paramètre de seuil pour en détecter la convergence. L'automatisme n'est donc pas total pour le moment. Ce paramètre de convergence est cependant beaucoup plus intuitif que les paramètres du filtre, dans la mesure où il correspond à une distance moyenne minimum entre plusieurs surfaces dans les zones de recouvrement.

Enfin, la fonction d'erreur que nous avons utilisée pour effectuer l'optimisation du paramètre δ_h doit être retravaillée. Nous pensons qu'elle peut faire échouer le processus dans un minimum local : le graphique 7.10 révèle en effet des instabilités dans l'évolution de la fonction, donnant lieu à de tels minima. Un risque de sur-filtrage ou sous-filtrage est alors à craindre le cas échéant, si le bon paramètre n'est pas trouvé. Nous pourrions notamment ajouter une pondération en chaque point dépendant du nombre de surfaces ayant servi à le corriger : plus ce nombre est grand, plus le filtrage implicitement généré par la correction non rigide est important, et donc plus le point est pertinent pour notre fonction d'erreur.

- Visualisation de champs surfaciques -

Modèles de champs surfaciques pour applications multiples en visualisation

De manière générale, un champ lumineux surfacique désigne une information perçue en chaque point de la surface d'un objet, et pour toutes les directions possibles d'observation. Nous employons ici volontairement le terme «information» au lieu de «couleur» ou «intensité lumineuse», dans la mesure où ces champs peuvent être employés pour la représentation d'autres types de données que celles-ci. Nous devrions d'ailleurs plutôt parler de *champs surfaciques directionnels*, mais nous garderons la dénomination précédente, qui fait référence à un modèle de paramétrisation 4D bien spécifique de cette information et désormais admis dans la communauté de la synthèse d'images (voir chapitre 5, section 4.3).

Nous allons voir, au cours de ce chapitre, différentes utilisations des champs lumineux surfaciques. La première, bien entendu, a pour objectif de permettre la visualisation des données qui ont été acquises à l'aide des méthodes présentées lors des chapitres précédents. Nous ne discuterons donc pas uniquement de visualisation, mais également des problématiques liées à la reconstruction de la texture à partir de ces données. Ces travaux sont présentés dans la section 1.

La seconde application des champs lumineux surfaciques qui est abordée ici porte sur la restitution d'objets transparents. Un phénomène optique complexe peut être observé pour de tels objets, la *réfringence*, qui induit une déviation des rayons lumineux les traversant. Ces déviations, liées au changement de vitesse de la lumière lorsqu'elle passe du milieu extérieur à l'objet à un milieu intérieur de densité différente, sont décrites par des lois physiques permettant de simuler très facilement le phénomène à l'aide d'algorithmes de lancer de rayons. Ces algorithmes sont malheureusement très coûteux, et le rendu temps réel d'un phénomène si complexe, même approximé, est alors particulièrement difficile à atteindre.

Nous proposons alors une représentation du phénomène de réfraction à l'aide d'un modèle de champs lumineux surfaciques qui sera présenté dans la section 2. Nous montrons notamment que cette approche tire parti des capacités des cartes graphiques actuelles pour aboutir à une restitution temps réel, qu'elle est visuellement plausible, et qu'elle permet de

prendre en compte les déviations induites par des géométries complexes, posant problème à certaines approches récemment présentées dans la littérature. Nous proposons d'ailleurs, à cette occasion, un bref état de l'art sur le sujet (section 2.1). Cette seconde application constitue en réalité la principale contribution parmi tous les travaux qui sont présentés dans ce chapitre.

Comme nous allons le voir, ces deux applications ne se basent absolument pas sur les mêmes modèles. Nous avons en fait adapté les moyens en fonction des besoins :

- nous avons choisi un modèle très simple pour la restitution des œuvres d'art numérisées, dans la mesure où celles-ci devraient être intégrées, à plus ou moins longs termes, à des galeries d'art virtuelles. Pour de telles galeries, plusieurs œuvres peuvent potentiellement être affichées en même temps, et le matériel informatique utilisé par les musées n'est pas forcément aussi performant que celui utilisé dans un laboratoire de recherche en informatique graphique.
- pour la restitution d'objets réfringents, à l'inverse, nous avons utilisé un modèle permettant une reconstruction beaucoup plus précise, vu la complexité du phénomène optique à simuler. Ce modèle est cependant beaucoup plus lourd, et ce à tous les niveaux, comme nous le verrons.

Des résultats concernant ces deux modèles sont présentés et étudiés séparément, dans les sections 1.4 et 2.7 respectivement, avant de clore ce chapitre par une conclusion générale en section 3.

1 Application 1 : visualisation des œuvres numérisées

Dans cette section, nous allons discuter de la reconstruction d'une texture dépendant du point de vue à partir de données réelles. Cette texture va nous permettre de restituer l'apparence des copies numériques que nous avons acquises. Les variations de cette apparence seront bien sûr contraintes à des conditions d'éclairage fixes, mais permettront à l'observateur de se déplacer librement autour de l'objet.

Nous considérons donc, pour commencer, que toutes les données issues des traitements décrits dans les chapitres précédents sont disponibles, c'est-à-dire :

- un modèle 3D résultant de la fusion des différentes acquisitions géométriques,
- un ensemble de photographies recalées sur ce modèle.

Nous allons donc commencer par décrire, dans la section 1.1, le modèle utilisé pour représenter la dépendance au point de vue en chaque point de la surface. Nous voyons ensuite, dans la section 1.2, comment nous estimons ce modèle à partir des données recueillies, de même que la manière dont nous le stockons pour en permettre la visualisation à l'aide du matériel graphique. Enfin, nous présentons, en section 1.4, les résultats obtenus pour différents objets réels que nous avons numérisés.

■ 1.1 Modèle de champs lumineux surfaciques

Le modèle de représentation que nous avons utilisé est le modèle de textures polynomiales présenté par Malzbender *et al.* [109]. Ce modèle a été initialement proposé dans le but de faire du rééclairage d'images, c'est-à-dire pour représenter la variation d'une surface observée depuis un point de vue fixe, mais éclairée selon une direction d'illumination arbitraire (voir chapitre 5, section 4.1).

À titre de rappel, ce modèle est défini de la manière suivante, à l'aide d'un polynôme quadratique dépendant de la direction locale d'illumination :

$$L_r(\vec{l}) = a_1 l_x^2 + a_2 l_y^2 + a_3 l_x l_y + a_4 l_x + a_5 l_y + a_6 \quad (8.1)$$

où L_r est la luminance réémise par le point considéré, \vec{l} est la direction d'incidence lumineuse exprimée localement dans le repère de ce point, et (l_x, l_y) correspond aux coordonnées issues de la projection de \vec{l} sur son plan tangent.

Dans notre cas, ce même modèle est utilisé mais en remplaçant la direction d'illumination \vec{l} par la direction d'observation \vec{v} afin de prendre en compte, cette fois-ci, les variations liées au déplacement de l'observateur :

$$L_r(\vec{v}) = a_1 v_x^2 + a_2 v_y^2 + a_3 v_x v_y + a_4 v_x + a_5 v_y + a_6 \quad (8.2)$$

Pour tenir compte du fait que l'information est en couleur, nous estimons un tel modèle pour chacun des trois canaux du modèle trichromatique RVB avec lequel nous travaillons.

L'argument avancé par les auteurs pour préférer encoder les variations liées au changement d'éclairage est que les variations liées au point de vue sont beaucoup plus chaotiques, du fait des phénomènes de masquage induits par la parallaxe. Cet argument est valable lorsque l'information géométrique n'est pas disponible, et que l'apparence de la surface, méso-structure incluse, doit être entièrement codée de manière photométrique par ce modèle. Cependant, si, comme dans notre cas, la géométrie est connue, y compris à l'échelle mésoscopique, ces effets de masquage peuvent être détectés et éliminés lors d'une phase d'analyse, aboutissant alors à des variations beaucoup plus lisses dans l'information à reconstruire.

De plus, il faut noter que le fait d'encoder les variations liées à l'éclairage introduit également des irrégularités dans l'information, du fait des ombrages qu'introduisent ces mêmes problèmes de masquage lorsque la direction d'illumination devient rasante. Nous pensons donc que, dès lors que la géométrie est connue, les textures polynomiales sont tout aussi adaptées à encoder un champ lumineux qu'à faire du rééclairage d'images.

■ 1.2 Reconstruction à partir de données réelles

Les maillages obtenus par numérisation sont souvent de résolution très fine (moins d'un millimètre entre deux sommets, en moyenne). Par rapport à l'échelle d'observation, cette résolution est suffisamment petite pour que nous puissions nous permettre de reconstruire

l'information d'apparence uniquement au niveau des sommets. Nous souhaitons donc estimer les paramètres $\{a_i\}_{1 \leq i \leq 6}$ en chaque sommet de la surface à partir de l'ensemble de photographies calibrées.

Pour commencer, nous extrayons les échantillons de luminance associés à chaque sommet. Pour ce faire, une projection du modèle 3D dans l'espace image de chaque photographie est effectuée. Afin de détecter et d'éliminer les parties non visibles de l'objet, cette phase de projection est accomplie en deux temps :

- une première reprojection est effectuée pour générer un tampon de profondeur.
- une seconde reprojection nous permet alors de distinguer les sommets qui sont visibles de ceux qui sont occultés en testant ce tampon de profondeur.

Ainsi, si un sommet n'est pas visible depuis un point de vue donné, aucun échantillon de luminance ne sera créé pour la photographie correspondante. À l'inverse, chaque point de vue pour lequel un sommet est visible donne lieu à un échantillon de luminance pour ce sommet.

À ce sujet, il est important de remarquer que l'information récupérée dans chaque photographie ne correspond pas directement à la luminance renvoyée par la surface observée. En effet, cette information tiens également compte d'une courbe de sensibilité dépendant de la caméra d'acquisition¹. Le passage de l'information capturée à l'information de luminance réelle doit donc se faire par un calibrage photométrique de cette caméra. Nous n'avons malheureusement pas pu nous en préoccuper pour le moment, notamment du fait de l'absence du matériel nécessaire à sa mise en œuvre. Dorénavant, lorsque nous parlerons de luminance, nous ferons en fait référence à l'information directement extraite des photographies.

Disposant donc de n échantillons de luminance $\{\widehat{L}_r(\vec{v}_j)\}_{1 \leq j \leq n}$, où $n \geq 6$, pour un sommet du maillage, les paramètres $\{a_i\}_{1 \leq i \leq 6}$ du polynôme en ce sommet peuvent alors être aisément estimés en résolvant le système linéaire suivant :

$$\underbrace{\begin{bmatrix} v_{1,x}^2 & v_{1,y}^2 & v_{1,x}v_{1,y} & v_{1,x} & v_{1,y} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{n,x}^2 & v_{n,y}^2 & v_{n,x}v_{n,y} & v_{n,x} & v_{n,y} & 1 \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_6 \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} \widehat{L}_r(\vec{v}_1) \\ \vdots \\ \widehat{L}_r(\vec{v}_n) \end{bmatrix}}_{\mathbf{l}} \quad (8.3)$$

où la matrice \mathbf{V} et le vecteur \mathbf{l} peuvent être construits directement à partir des échantillons, et où le vecteur \mathbf{a} constitue alors l'inconnue à résoudre.

Dellepiane *et al.* [39], qui ont travaillé sur l'acquisition de textures polynomiales hautes résolutions, ont noté, lors d'une tentative de diminution du nombre de lampes nécessaires à leur capture, l'importance de l'échantillonnage dans la qualité de l'estimation. Cela est tout à fait logique, dans la mesure où cette estimation tente de faire coller au mieux les paramètres aux données, quitte à ce que le modèle aboutisse à une extrapolation aberrante là où aucune information n'a été fournie pour l'estimation. C'est pourquoi les travaux s'attachant à la capture de textures polynomiales utilisent généralement des dispositifs matériels permettant de récupérer un échantillonnage dense [39, 109].

¹Voir annexe A.

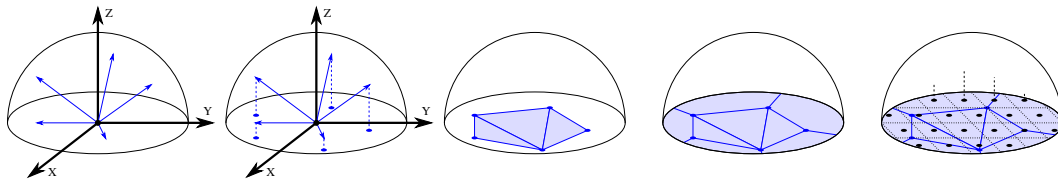


Figure 8.1 – Rééchantillonnage des directions d’observation. De gauche à droite : 1) échantillons initiaux. 2) projection sur le plan tangent. 3) triangulation de Delaunay des échantillons projetés. 4) extrusion des bords de la triangulation pour couvrir tout l’hémisphère d’observation. 5) génération de nouveaux échantillons par interpolation des valeurs aux sommets.

De notre côté, si nous sommes capables de fournir des échantillons de luminance dépendant du point de vue, nous ne sommes en revanche pas en mesure de garantir une densité d’échantillonnage suffisante, ni même homogène, des directions d’observation. Comme nous le montrerons dans les résultats, nous avons travaillé sur deux copies numériques, composées respectivement de 27 et 42 photographies. Étant donné que ces points de vue échantillonnent la luminance pour l’ensemble de l’objet, on peut considérer qu’il y a au moins la moitié d’entre eux pour lesquels un point quelconque de la surface n’est pas visible (car appartenant à l’hémisphère situé sous la surface par rapport à ce point). Cela signifie que chaque sommet se voit assigner, au grand maximum, 14 ou 21 échantillons de luminance, selon l’objet considéré. Ce nombre, comparé aux récents travaux sur les textures polynomiales, où le nombre de directions d’illumination en chaque point se situe entre 60 et 100, est clairement insuffisant pour correctement contraindre le processus d’estimation, d’autant plus que la distribution de ces points de vue sur l’hémisphère d’observation peut être totalement arbitraire.

Nous commençons donc par rééchantillonner les données en chaque sommet, selon un schéma similaire à celui proposé par Chen *et al.* [26], de manière à obtenir un nouvel échantillonnage beaucoup plus dense et mieux réparti sur l’hémisphère d’observation :

1. La direction d’observation de chaque échantillon est projetée sur le plan tangent du sommet considéré. Nous obtenons donc un ensemble de points 2D exprimés dans ce plan (figure 8.1-2). À chacun de ces points est associé la valeur de luminance de l’échantillon correspondant.
2. Une triangulation de Delaunay est alors générée à partir de cet ensemble de points (figure 8.1-3).
3. Le bord de cette triangulation est alors extrudé de manière à couvrir tout le domaine des directions d’observation projetées, c’est-à-dire de manière à englober le disque unité centré sur le sommet traité (figure 8.1-4).
4. Un nouvel échantillonnage est alors généré à partir de cette triangulation, la valeur de luminance de chaque nouvel échantillon étant obtenue par interpolation des valeurs aux sommets du triangle dans lequel il se trouve (figure 8.1-5).

Pour la génération des nouveaux échantillons à partir de la triangulation, nous avons utilisé le pipeline de rendu OpenGL, en associant à chaque sommet une couleur correspondant à sa valeur de luminance. La phase d’interpolation est donc totalement prise en charge par le matériel graphique. Le tampon dans lequel se fait le rendu est alors récupéré, et chaque

pixel nous fournit alors un nouvel échantillon. Dans notre implantation, nous avons utilisé un tampon de 64×64 , ce qui génère environ 3200 échantillons par sommet (ceux situés à l'extérieur du disque unité ne sont pas conservés), un nombre plus que suffisant pour une estimation correcte de la texture polynomiale.

Cette méthode de rééchantillonnage ne fonctionne évidemment que si le nombre d'échantillons initiaux est au moins de trois, sans quoi la triangulation de Delaunay n'aboutit à rien. Nous traitons les cas particuliers (nombre d'échantillons < 3) de la manière suivante :

- **aucun échantillon** : ce cas de figure se produit rarement, mais il peut tout de même survenir pour des sommets se trouvant dans des zones géométriques difficiles d'accès qui ne sont visibles depuis aucune des photographies. Actuellement, nous ne gérons pas ce problème : la couleur est systématiquement mise à zéro (noir). Afin que ce cas de figure soit correctement géré, une technique de propagation de l'information chromatique comme celle proposée par Debevec *et al.* [38] peut cependant être employée.
- **un seul échantillon** : la couleur du sommet est alors considérée comme constante. Dans ce cas, les coefficients du polynôme peuvent être directement générés en donnant à a_6 (coefficient constant) la valeur de l'échantillon, et en mettant à zéro tous les autres.
- **deux échantillons** : dans le repère local du sommet, nous construisons deux plans passant chacun par le sommet considéré et étant respectivement incident à l'un et l'autre des deux échantillons. Ces plans sont orientés de manière à ce que leurs distances orthogonales au point sommet de l'hémisphère soient maximales. Un nouvel échantillonnage est alors généré : si la direction de l'un des nouveaux échantillons se trouve entre les deux plans, sa valeur de luminance est obtenue par interpolation des deux échantillons initiaux. Sinon, elle prend la valeur de celui dont le plan est le plus proche.

Une fois que le nouvel échantillonnage des directions d'observation a été obtenu, les coefficients (a_1, \dots, a_6) de texture polynomiale sont déterminés en résolvant le système linéaire de l'équation 8.3. Dans la mesure où le nombre d'échantillons utilisés pour ce faire est largement supérieur au nombre d'inconnues, le système en question est sur-contraint. L'inversion de la matrice \mathbf{V} est alors accomplie grâce à une *décomposition en valeurs singulières* [144].

■ 1.3 Visualisation temps réel

Pour pouvoir tirer parti du matériel graphique lors de l'affichage, les coefficients de textures polynomiales doivent être stockés sous forme d'images de texture. Nous rappelons que nous disposons, à l'issue de la phase de reconstruction précédente, de trois ensembles de coefficients par sommet : (a_1^R, \dots, a_6^R) , (a_1^V, \dots, a_6^V) et (a_1^B, \dots, a_6^B) , soit un par canal de couleur du modèle trichromatique RVB.

Nous utilisons alors un total de six images de texture, chacune d'elles stockant les trois coefficients RVB représentant le même monôme dans la texture polynomiale. Le modèle 3D est alors stocké en espace texture de manière éclaté, c'est-à-dire que chacun de ses triangles est considéré indépendamment des autres, en utilisant trois *texels*² dans un carré

²De l'anglais *texture element*, similaire à *pixel* pour *picture element*.

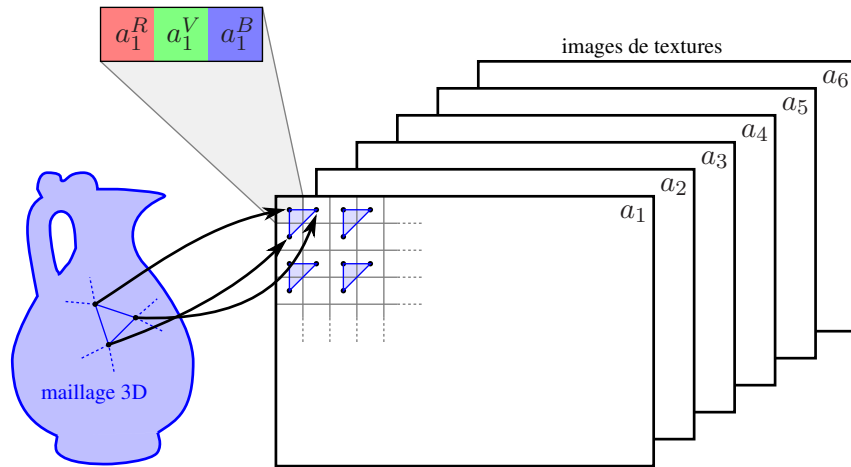


Figure 8.2 – Schéma de stockage de la texture polynomiale en vue de son évaluation temps réel par le processeur graphique.

de 2×2 pour stocker les coefficients à ses sommets, comme le montre la figure 8.2.

Au moment de l’affichage, si un triangle s’affiche sur plusieurs pixels, les coefficients de textures polynomiales pour les pixels représentant l’intérieur du triangle sont automatiquement déterminés par la carte graphique à l’aide d’une interpolation bilinéaire en espace texture. Or, cette interpolation utilise les quatre texels les plus proches, comme le définit l’équation 8.4 et comme l’illustre la figure 8.3.

$$I_{bil}(\alpha, \beta) = (1 - \alpha)(1 - \beta) t_{00} + \alpha(1 - \beta) t_{10} + \alpha\beta t_{11} + (1 - \alpha)\beta t_{01} \quad (8.4)$$

Ainsi, le schéma de stockage que nous utilisons va générer des artefacts du fait que le quatrième texel de chaque carré 2×2 (celui ne correspondant à aucun sommet du triangle) sera tout de même utilisé par le calcul d’interpolation.

Désactiver l’interpolation permettrait, bien entendu, de résoudre le problème, mais au risque de générer d’importants effets de «pixelisation». En revanche, il est possible de donner à ce quatrième texel une valeur qui soit telle que l’interpolation bilinéaire ne tienne compte que des trois texels où l’information utile est définie. En effet, si la valeur du texel t_{11} dans l’équation précédente est remplacée par le calcul donné par l’équation 8.5, on obtient le schéma d’interpolation défini par l’équation 8.6, qui n’est rien d’autre que la formule de l’interpolation barycentrique.

$$t_{11} = t_{01} + t_{10} - t_{00} \quad (8.5)$$

$$I_{bar}(\alpha, \beta) = \alpha t_{10} + \beta t_{01} + (1 - \alpha - \beta) t_{00} \quad (8.6)$$

Une fois modifiée de la sorte, la valeur initialement indéfinie de ce quatrième texel n’interfère plus dans le mécanisme d’interpolation de la carte graphique. La modification est donc effectuée pour tous les carrés de 2×2 , et pour chacune des six images de texture utilisées.

Bien entendu, une autre solution pour résoudre ce problème d’interpolation, plus simple et plus économique en mémoire, consiste à placer deux triangles adjacents (partageant une

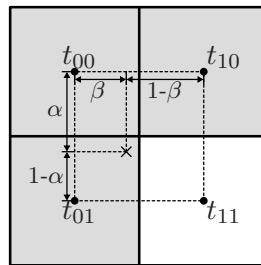


Figure 8.3 – Interpolation bilinéaire utilisée lors de l’indexation texture par la carte graphique. Les quatre texels les plus proches sont considérés.

même arête sur le maillage) dans chaque carré de 2×2 . Nous avons préféré utiliser notre correction du fait qu’elle ne nécessite pas d’établir préalablement l’adjacence entre les triangles, ce qui rend donc sa mise en œuvre plus immédiate.

En plus des coefficients de textures polynomiales, les repères locaux permettant de déterminer en chaque sommet les coordonnées v_x et v_y associées à une direction d’observation \vec{v} donnée doivent également être mis à disposition du processeur graphique. Pour ce faire, nous convertissons la matrice 3×3 représentant le repère local d’un sommet sous la forme du quaternion correspondant. Les quatre coordonnées de ce quaternion sont alors passées en attributs de ce sommet lors du rendu.

Le processeur graphique est alors reprogrammé pour effectuer les traitements suivants :

– **en chaque sommet** :

1. reconstruire le repère local à partir du quaternion passé en attribut,
2. déterminer la direction locale d’observation, et donc les coordonnées v_x et v_y de cette direction dans le plan tangent.

– **pour chaque pixel affiché** :

1. récupérer les coefficients $(a_1^R, a_1^V, a_1^B), \dots, (a_6^R, a_6^V, a_6^B)$ à partir des six textures,
2. évaluer le polynôme de l’équation 8.2 en utilisant ces coefficients ainsi que les coordonnées v_x et v_y obtenues par interpolation des valeurs aux sommets,
3. donner comme couleur au pixel écran actuel le résultat de cette évaluation.

■ 1.4 Résultats expérimentaux

Qualité visuelle

Pour évaluer la qualité de la reconstruction par textures polynomiales, nous avons comparé le rendu obtenu à des photographies réelles ayant été utilisées lors de l’estimation de ce modèle. Les résultats de ces comparaisons sont données par les figures 8.4 et 8.5 pour les modèles de la statue africaine et du vase grec.



Figure 8.4 – Comparaison entre le rendu par textures polynomiales dépendant du point de vue (ligne du bas) et des photographies réelles (ligne du haut) pour le modèle de la statue africaine.



Figure 8.5 – Comparaison entre le rendu par textures polynomiales dépendant du point de vue (ligne du bas) et des photographies réelles (ligne du haut) pour le modèle du vase grec.

La statue africaine a été reconstruite à partir de 6 images télémétriques et 42 photographies, et le vase grec à partir de 5 images télémétriques et 27 photographies. Le faible nombre d'images télémétriques utilisées dans les deux cas explique les trous apparents au niveau de la géométrie, correspondant à des zones qui n'ont pas été mesurées lors de la numérisation 3D.

En regardant d'abord les images synthétiques uniquement, on peut voir que l'apparence est globalement bien rendue, tenant compte de certains effets spéculaires en fonction du point de vue. Cependant, en les comparant aux photographies réelles, nous pouvons faire deux constats :

- on peut voir qu'un effet de flou a fait son apparition : dans le cas de la statue africaine, la majorité des veinules du bois sont en effet bien moins nettes que sur les photographies. Ce flou est également très apparent au niveau des détails picturaux du vase grec.

Modèle	Nombre de triangles	Consommation mémoire (Mo)	Fréquence d'affichage (Hz)
Statue africaine	184K	~10	22
Vase grec	362K	~20	11

Tableau 8.1 – Fréquence d'affichage et consommation mémoire pour le rendu par textures polynomiales dépendant du point de vue.

- on constate également une altération flagrante des reflets spéculaires : ceux-ci sont, d'une manière générale, très lissés. Mais on peut également remarquer que la forme de ces reflets, ainsi que les points de vue sous lesquels ils apparaissent, ont été modifiés.

Concernant l'effet de flou, il n'est dû ni au modèle de textures polynomiales, ni au fait que la texture ne soit définie qu'aux sommets (le maillage est en effet suffisamment fin) : il provient simplement du fait que les différents clichés utilisés pour la reconstruction ne sont pas toujours parfaitement ajustés. Des décalages de un pixel par-ci par-là engendrent forcément un effet de moyennage local de l'information de texture lors de la reconstruction. Sur ce point, il faut bien reconnaître que nous sommes désavantagés par rapport à l'usage classique des textures polynomiales : utilisées pour faire du rééclairage d'images, toutes les photographies sont prises depuis le même point de vue, éliminant ainsi les besoins en recalage d'images. Dans notre cas, les légers défauts de précision que nous avons présentés dans le chapitre 6 concernant le calibrage des points de vue ont nécessairement un impact majeur sur la présence de ce flou.

Pour ce qui est de la non-préservation des reflets spéculaires, rappelons que le modèle de textures polynomiales est connu pour lisser l'information qu'il représente. Au final, la luminance n'est représentée que par une quadrique exprimée dans le repère local de chaque point, ce qui limite forcément le degré de complexité que peut atteindre la reconstruction. Cependant, en regardant les résultats pour la statue africaine, force est de constater que les reflets spéculaires sont toujours présents. Il sont certes beaucoup moins tranchés, et ne coïncident pas toujours exactement avec ceux observables dans la réalité, mais il suffisent tout de même à donner une «intuition» à l'observateur quant aux matériaux qui composent l'objet affiché.

Performances

Les performances de rendu pour l'algorithme des textures polynomiales sont reportées dans le tableau 8.1. Elles ont été mesurées sur un ordinateur de bureau muni d'un processeur AMD Athlon 3800+ et d'une carte graphique NVidia GeForce 8800 GTS.

Il faut noter que les faibles vitesses de rendu par rapport à ce que l'on aurait été en droit d'attendre ne sont pas nécessairement à mettre en lien avec le modèle de textures polynomiales en lui-même : nous avons mesuré exactement les mêmes temps d'exécution en désactivant son évaluation par la carte graphique. Cette évaluation n'a donc qu'un coût très négligeable en termes de temps de calcul.

En revanche, entre les deux objets 3D, on peut observer une chute des performances d'un facteur deux alors que le nombre de triangles a, quant à lui, doublé. Ainsi, ce qui semble nuire le plus aux performances d'affichage est la forte densité des maillages. Des questions quant à l'intérêt d'utiliser un algorithme de simplification de maillages dans le pipeline de reconstruction se posent donc. La mise en œuvre d'un tel algorithme demandera alors de revoir le processus de reconstruction de la texture, dans la mesure où celle-ci devra être définie non plus par sommet mais par triangle, afin de permettre de travailler à des résolutions différentes pour la géométrie et pour la texture.

2 Application 2 : simulation de la réfringence

Dans cette section, nous proposons une seconde application des champs lumineux surfaciques visant à simuler le phénomène complexe de réfringence se produisant dans le cas d'objets transparents.

Contrairement à l'application précédente, nous n'allons travailler ici qu'avec des données synthétiques, la principale raison étant que la numérisation d'objets transparents est, à l'heure actuelle, un procédé encore très mal maîtrisé. Les seules méthodes qui abondent en ce sens sont les algorithmes de *matting* [29, 212], assimilables à des techniques de ré-éclairage d'images, et qui, de ce fait, ne sont capables de générer de nouvelles images que depuis un point de vue exactement identique à celui ayant servi à leur acquisition. Une autre approche, très aboutie, est celle de Matusik *et al.* [115], qui capturent à la fois la géométrie, les propriétés des matériaux, et les phénomènes de réflexion et de réfraction. Cependant, le matériel utilisé pour l'acquisition est d'une complexité incroyable, la taille des données pour chaque copie numérique est énorme, même après traitement (plusieurs Go), et plusieurs minutes sont nécessaires à la génération d'une seule image. Notre intérêt ne se situant pas, dans le cas présent, au niveau de l'acquisition, nous nous plaçons purement et simplement dans le cas d'objets de synthèse.

La méthode que nous proposons représente le phénomène de réfringence comme un champ de distorsions dépendant du point de vue et stocké sur la surface d'un objet. Cette approche permet de prendre en compte des objets de géométries quelconques, et pouvant même être composés de différents indices de réfraction (un verre rempli d'eau, par exemple), ce qui n'est pas le cas des méthodes orientées temps réel existant au moment où nous avons développé la nôtre. Pour s'en rendre compte, nous allons commencer par un bref état de l'art concernant quelques travaux sur le rendu des objets réfringents, en section 2.1. Nous y ferons notamment les quelques rappels d'optique nécessaires à la compréhension du phénomène. Nous présentons ensuite le principe de notre méthode, en section 2.2, en décrivant, notamment, comment le phénomène peut être représenté par un champ lumineux surfacique. La construction de ce champ passe par une étape de prétraitements qui va évaluer de manière exhaustive l'apparence que doit prendre l'objet en fonction du point de vue, avant de compresser les données recueillies afin qu'elles demeurent exploitables dans le cas d'une

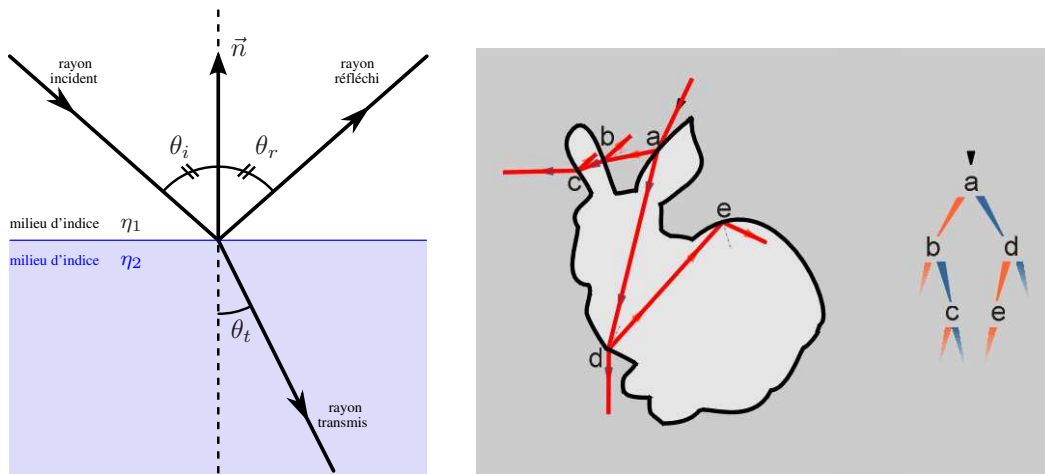


Figure 8.6 — **Gauche** : réflexion et transmission selon les lois de Snell-Descartes. **Droite** : chaque rayon lumineux incident à un objet transparent génère un arbre binaire infini de rayons.

visualisation temps réel. Ces prétraitements sont présentés en section 2.3. L'algorithme de rendu, conçu pour exploiter au maximum les fonctionnalités de la carte graphique, est quant à lui détaillé dans la section 2.4, avant d'en compléter la description à l'aide de certaines améliorations (section 2.5) et de certains effets visuels additionnels (section 2.6). Des résultats concernant ces travaux sont finalement présentés et étudiés en section 2.7.

■ 2.1 Rappels d'optique et travaux antérieurs

À l'interface³ entre deux milieux d'indices de réfraction η_1 et η_2 différents, un rayon lumineux incident se divise en deux nouveaux rayons : un rayon *réfléchi* et un rayon *transmis* (ou *réfracté*), comme le montre la figure 8.6-gauche. Les directions que prennent ces deux nouveaux rayons, dictées par les lois de Snell-Descartes, sont telles que :

- l'angle θ_r entre le rayon réfléchi et la normale \vec{n} au point d'incidence est égal à l'angle θ_i entre le rayon incident et cette même normale,
- si θ_t est l'angle entre la normale \vec{n} et le rayon transmis, la relation suivante est vérifiée :

$$\eta_1 \cdot \sin(\theta_i) = \eta_2 \cdot \sin(\theta_t) \quad (8.7)$$

De plus, l'intensité lumineuse portée par le rayon incident est répartie entre les deux nouveaux rayons selon une relation dépendant des conditions d'incidence. Cette relation est donnée par les équations de Fresnel, dont les formules sont fournies dans l'annexe B. Intuitivement, les équations de Fresnel sont responsables du fait que si l'on observe une surface d'eau de front, celle-ci est totalement transparente, alors que si on l'observe selon un angle très rasant, elle se comporte comme un miroir. Dans le premier cas, les coefficients de Fresnel donnent plus de poids au rayons transmis, alors que dans le second cas, c'est le rayon réfléchi qui a le plus d'importance.

³Nous considérons ici le cas d'une interface parfaite (interface sans rugosité).

L'ensemble de ces règles s'applique à chaque changement de milieu, c'est-à-dire à chaque fois que la lumière passe du milieu extérieur au milieu intérieur à un objet, et vice versa. Notamment, le rayon réfracté peut lui-même rencontrer la paroi de l'objet dans lequel il se propage, auquel cas il va à son tour générer deux nouveaux rayons, et ainsi de suite. Il en va bien entendu de même pour le rayon réfléchi. Un seul rayon lumineux incident va donc générer un arbre binaire infini de rayons, comme on peut le voir sur la figure 8.6-droite, avec notamment des effets de réflexions internes qui ajoutent à la complexité visuelle du phénomène.

La seule méthode, en informatique graphique, qui soit capable de réellement simuler le phénomène de réfraction est le lancer de rayons [70, 198]. En effet, de par son mode de fonctionnement, la réfraction peut être gérée correctement en déployant simplement l'arbre binaire de rayons évoqués à l'instant, jusqu'à ce qu'un niveau donné de récursivité ait été atteint. Certaines extensions de cet algorithme, telles que le lancer de faisceaux [71] ou le lancer de rayons sur des objets volumiques [100, 152], se sont également intéressés à la simulation de ce phénomène, avec pour résultats des images de très bonnes qualités. Cependant, ces méthodes restent, aujourd'hui encore, très coûteuses en temps de calcul.

À l'inverse, les techniques de rendu basées polygones que l'on trouve dans les bibliothèques graphiques actuelles permettent des rendus de très hautes performances, principalement du fait du support matériel sous-jacent, mais ne permettent pas une simulation simple de phénomènes optiques complexes tels que la réfringence. Des techniques hybrides ont donc été développées, combinant rendu basé polygones et lancer de rayons [66, 133]. Le principe général de ces méthodes est alors d'effectuer un lancer de rayon non pas par pixel écran, mais en chaque sommet de l'objet 3D, et d'interpoler les directions trouvées pour les rayons transmis. Étant donné que, contrairement à la réflexion, la réfraction a un comportement non linéaire [71], cette interpolation des directions aux sommets n'aboutit qu'à une grossière approximation si la résolution du maillage sur lequel elle est appliquée est trop fine. Ainsi, Hakura *et al.* [66] proposent un mécanisme adaptatif de subdivision qui s'applique aux triangles pour lesquels les directions des rayons transmis aux sommets sont trop différentes l'une de l'autre. Ils proposent également de simplifier l'étape de lancer de rayons en élaguant l'arbre de rayons généré de manière à ne conserver qu'un unique chemin. Pour ce faire, seul celui du rayon transmis ou du rayon réfléchi dont le coefficient de Fresnel est le plus élevé est suivi, l'autre étant simplement ignoré. Malgré toutes ces simplifications, plusieurs dizaines de secondes sont tout de même nécessaires pour la génération d'une seule image.

Dans le but d'atteindre le temps réel, certaines méthodes se sont focalisées sur des scènes ou des situations spécifiques. Tso *et al.* [188], par exemple, se placent dans le cas de la restitution de surfaces océaniques, où les rayons lumineux ne sont déviés qu'une seule fois, à l'interface entre l'air et l'eau, avant de venir heurter le fond marin. Pour ce même type de scènes, Baboud *et al.* [5] proposent une implémentation sur processeur graphique, utilisant deux champs de hauteurs pour représenter respectivement le fond marin et la surface d'eau. Le temps réel est atteint, avec la possibilité supplémentaire de simuler dynamiquement les caustiques produits sur le fond lors d'un éclairage de la surface d'eau. Un autre exemple d'applications dédiées est celle de Guy *et al.* [65], qui ont développé un logiciel

de rendu réaliste interactif de pierres précieuses, gérant de nombreux effets optiques parmi tous ceux pouvant se produire avec de tels objets. Basée sur un lancer de faisceaux sur carte graphique, cette méthode n'est réellement conçue que pour le rendu de pierres précieuses, objets composés uniquement de quelques centaines de facettes planaires.

Dans le cas d'objets quelconques, des simplifications ont été proposées afin de permettre l'utilisation du matériel graphique. Wyman [200] propose un modèle considérant deux intersections du rayon lumineux avec la géométrie, c'est-à-dire un point d'entrée et un point de sortie, et considère la déviation qui a lieu pour chacune de ces intersections. Cette technique est formulée de manière à permettre de correctement utiliser le matériel graphique, et aboutit ainsi à un rendu temps réel et très convaincant du phénomène de réfraction. En revanche, elle repose sur un précalcul qui ne permet, *in fine*, de ne travailler que sur des objets non déformables. Cette contrainte peut éventuellement être levée grâce à l'approximation proposée par Seipel *et al.* [165] qui, même si elle est beaucoup moins précise, ne nécessite quant à elle aucun précalcul. Cependant, pour l'une ou l'autre de ces deux méthodes, seules deux intersections du rayon avec l'objet sont considérées, ce qui implique que les modèles 3D comportant de nombreuses concavités, ou bien composés de différents objets imbriqués les uns dans les autres, ne peuvent pas être restitués convenablement. Ihrke *et al.* [83] ont présenté un moyen de tenir compte de tels objets, et de simuler de nombreux effets liés au phénomène de réfringence, grâce à une structure représentant l'évolution dans l'espace des fronts d'ondes d'énergie lumineuse. Il faut cependant remarquer que cette approche a été développée ultérieurement aux travaux que nous allons présenter ici.

Enfin, on peut noter l'approche purement basée images de Heidrich *et al.* [73], basée sur un champ lumineux (voir rappel en chapitre 5, section 4.2) pour simuler le phénomène de réfraction. Au lieu d'encoder une luminance réfléchie en fonction d'un rayon d'observation, le champ lumineux encode la direction du rayon de sortie, après réfraction à l'intérieur de l'objet. Ce rayon de sortie est alors utilisé pour indexer un environnement, décrit soit par une texture d'environnement, soit par un second champ lumineux permettant de gérer correctement la parallaxe (tenir compte de la proximité relative des différents éléments de cet environnement). Cette méthode est, bien entendu, sujette à tous les problèmes liés aux champs lumineux dans le cas général : artefacts visuels, grande consommation mémoire, restriction du point de vue, limitation de la résolution d'affichage.

Cette dernière méthode, du fait qu'elle se base sur un précalcul exhaustif de toutes les directions d'observation possibles, permet forcément une restitution très réaliste du phénomène de réfraction. En ce sens, notre approche est similaire : elle se base également sur un précalcul exhaustif. Cependant, comme nous stockons l'information directement sur la surface du modèle 3D (ce qui est le principe des champs lumineux surfaciques), les problèmes communément rencontrés avec les champs lumineux standards sont ainsi évités.

■ 2.2 Principe de la méthode

Notre méthode se base sur un modèle de champs lumineux surfaciques pour représenter le phénomène de réfringence. Ainsi, au même titre que toutes les méthodes de cette catégorie

[26, 121, 199] , une information directionnelle (dépendant du point de vue) est définie en chaque point de la surface d'un modèle 3D.

Dans notre cas, au lieu de représenter une intensité lumineuse réfléchie, l'information directionnelle encode, pour toutes les directions d'observation possibles, la direction du rayon lumineux sortant de l'objet après en avoir parcouru l'intérieur par rebonds successifs. Étant donné que le rayon de sortie n'est pas censé être unique, comme cela a été rappelé précédemment, nous ne gardons que celui transportant l'énergie lumineuse la plus importante, à l'instar de Hakura *et al.* [66]. Le détail de cette méthode de sélection sera d'ailleurs explicité. Cette information est d'abord échantillonnée, de manière exhaustive, pour de nombreux points de la surface et pour un grand nombre de directions d'observation lors d'une phase de précalculs que nous allons détailler en section 2.3. Les données recueillies à l'issue de ce prétraitement, initialement très denses, sont alors compressées à l'aide d'une représentation en harmoniques sphériques, dont la théorie sera rappelée dans cette même section. Cet outil mathématique, qui est le pendant des séries de Fourier sur la sphère, permet alors de représenter l'information directionnelle en chaque point de la surface à l'aide de quelques coefficients réels seulement. Ces coefficients sont alors stockés dans des images de texture afin de pouvoir être utilisés *a posteriori* par le processeur graphique.

Lors de la phase de visualisation, dont le détail sera donné dans la section 2.4, les coefficients d'harmoniques sphériques sont récupérés depuis les textures, permettant ainsi de reconstruire l'information directionnelle. Celle-ci est alors évaluée pour la direction d'observation courante, fournissant en chaque point de la surface la direction du rayon quittant l'objet après avoir été réfracté et réfléchi un certain nombre de fois à l'intérieur de celui-ci. Cette direction est finalement utilisée pour indexer une texture d'environnement, qui décrit ce qui se trouve au-delà de l'objet par rapport au point de vue, simulant ainsi l'effet de transparence.

■ 2.3 Prétraitements

Échantillonnage spatial et directionnel

Contrairement au cas précédent de la visualisation d'œuvres d'art, nous ne souhaitons pas nécessairement travailler sur des maillages de résolutions fines. À ce moment là, nous avons opté pour cette solution du fait que les maillages fournis par la numérisation 3D sont, par nature, très denses. Dans le cas présent, puisque nous travaillons sur des données synthétiques, il est tout à fait possible de s'abstraire de ce prérequis, mais il faut alors que l'information directionnelle ne soit plus simplement définie au niveau des sommets.

Nous avons choisi d'employer le même schéma éclaté de représentation de l'objet 3D en espace texture que précédemment, sauf que chaque triangle est cette fois empaqueté dans un carré de $n \times n$ texels au lieu de 2×2 . Cela permet d'obtenir l'information de texture à un niveau de résolution différent de celui de la géométrie. Bien entendu, dans le cas d'un maillage fin, il est toujours possible de revenir à une information définie par sommet en

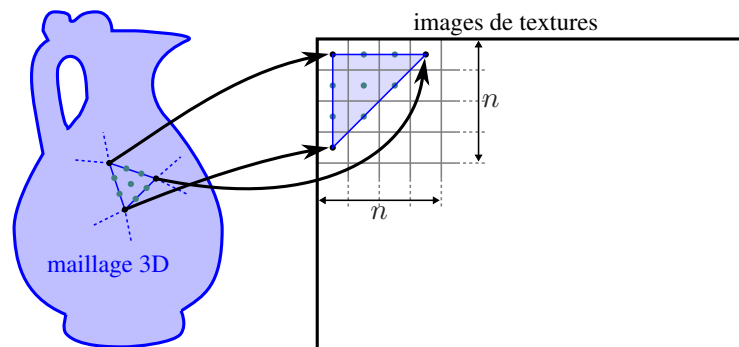


Figure 8.7 – Construction de l'échantillonnage spatial. Chaque triangle est stocké en espace texture sur un carré de $n \times n$ texels, et les points 3D associés aux texels qu'il couvre constituent les échantillons spatiaux pour ce triangle.

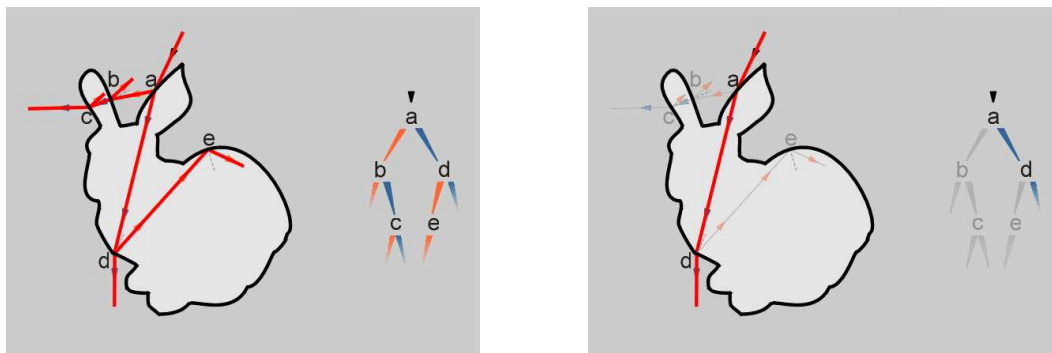


Figure 8.8 – Simplification de l'arbre binaire de rayons engendré par le phénomène de réfraction (Hakura *et al.* [66]) : à chaque interface, seul le rayon dont le coefficient de Fresnel est le plus élevé est conservé.

fixant $n = 2$. Pour une meilleure optimisation de l'espace texture, on peut noter que les triangles peuvent également être rangés tête-bêche. Une fois cette «paramétrisation» dans l'espace texture effectuée, chaque texel ainsi couvert peut être associé à un point 3D sur le maillage et l'ensemble de ces points constitue l'échantillonnage spatial au niveau duquel l'information directionnelle va être définie, comme l'illustre la figure 8.7.

Cette information directionnelle est alors échantillonnée à l'aide d'un algorithme de lancer de rayons qui va chercher, pour chaque échantillon spatial, la direction du rayon sortant de l'objet après réfraction pour un grand nombre de directions d'observation possibles. Lors des quelques rappels optiques de la section 2.1, nous avons expliqué qu'un rayon lumineux intersectant un objet transparent génère deux nouveaux rayons, l'un réfléchi, l'autre transmis. Si l'un ou l'autre de ces deux rayons heurte également la surface de l'objet, il donne lui-même naissance à deux nouveaux rayons, et ainsi de suite, de telle manière que notre rayon initial donne naissance à un arbre binaire infini. Ainsi, pour une direction d'observation donnée, la direction de sortie n'est pas unique mais il en existe une infinité, ce qui n'est évidemment pas envisageable dans la pratique.

Afin de ne conserver qu'une unique direction de sortie pour chaque direction d'observation,

nous avons employé la simplification proposée par Hakura *et al.* [66], dont le principe, illustré sur la figure 8.8, est de ne conserver que le rayon, parmi les deux obtenus à chaque interface, pour lequel le coefficient de Fresnel est le plus élevé. Ainsi, nous n’obtenons plus un arbre mais un chemin de rayons, correspondant au trajet le long duquel la quantité d’énergie lumineuse reste la plus significative.

Une fois tous ces traitements terminés, nous obtenons, pour un grand nombre de points sur la surface de l’objet, un échantillonnage dense de la direction du rayon quittant l’objet en fonction de la direction d’observation. Les données correspondantes représentent alors plusieurs Go, ce qui n’est évidemment pas exploitable en l’état pour une visualisation temps réel. Nous allons donc maintenant voir comment compresser la composante directionnelle de notre champ lumineux surfacique.

Compression de l’information directionnelle par harmoniques sphériques

Les harmoniques sphériques [106], notées Y_l^m , constituent une famille de fonctions de fréquences croissantes définies sur la sphère S^2 . Ces fonctions, à valeurs complexes, sont définies pour chaque direction (θ, ϕ) de S^2 par l’équation 8.8 :

$$Y_l^m(\theta, \phi) = K_l^m P_l^m(\cos \theta) e^{im\phi} \quad (8.8)$$

où P_l^m correspond au polynôme de Legendre associé, et K_l^m est un coefficient de normalisation défini par l’équation suivante :

$$K_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \quad (8.9)$$

Les fonctions harmoniques peuvent également être définies dans le cas de fonctions à valeurs réelles à partir de leurs équivalents complexes Y_l^m . Ainsi, les harmoniques sphériques réelles y_l^m sont définies par l’équation 8.10 :

$$y_l^m = \begin{cases} \sqrt{2} \operatorname{Re}(Y_l^m) & m > 0 \\ \sqrt{2} \operatorname{Im}(Y_l^m) & m < 0 \\ Y_l^0 & m = 0 \end{cases} \quad (8.10)$$

Dans la mesure où ces harmoniques forment une famille complète de polynômes orthogonaux, n’importe quelle fonction sphérique f de carré intégrable peut être représentée par une combinaison linéaire des fonctions harmoniques Y_l^m :

$$f = \sum_{l=0}^{\infty} \sum_{m=-l}^l C_l^m \cdot Y_l^m \quad (8.11)$$

où chaque coefficient C_l^m impliqué est alors défini par la convolution de la fonction f avec la fonction harmonique Y_l^m correspondante, comme le montre l’équation 8.12 :

$$C_l^m = \int_{S^2} f(\theta, \phi) \cdot Y_l^m(\theta, \phi) d\theta d\phi \quad (8.12)$$

Dans le cas pratique, la somme infinie de l'équation 8.11 ne peut évidemment pas être évaluée. Elle doit nécessairement être tronquée, ce qui ne permet d'obtenir qu'une reconstruction partielle de la fonction f , que nous noterons \hat{f} :

$$f \approx \hat{f} = \sum_{l=0}^n \sum_{m=-l}^l C_l^m \cdot Y_l^m \quad (8.13)$$

La qualité de l'approximation de la fonction f par la fonction \hat{f} dépend alors totalement de l'ordre n de reconstruction choisi : plus cet ordre est grand, plus la fonction \hat{f} tend vers la fonction f . Cependant, plus n est grand, plus le nombre de coefficients C_l^m nécessaires à la représentation de la fonction \hat{f} devient lui aussi important, d'autant plus qu'il croît de manière quadratique en n (un ordre de reconstruction n signifie en fait $(n+1)^2$ coefficients, comme cela peut être déduit de l'équation 8.13). La valeur de n doit donc être choisie de manière à établir un bon compromis entre la qualité de la reconstruction et l'espace de stockage qu'elle nécessite.

De nombreux travaux en synthèse d'images ont utilisé les harmoniques sphériques. Le récent papier de Sloan [169] contient, à ce propos, une liste exhaustive de références vers ces travaux. Au vu de la littérature, force est de constater que des ordres de reconstruction supérieurs à 7 (soit 64 coefficients) ne sont pratiquement jamais utilisés dans le cas d'applications de rendu temps réel. En ce qui nous concerne, nous avons travaillé jusqu'à un ordre 8 (81 coefficients). Nous discuterons, dans la section résultats (section 2.7), de l'importance de la valeur de n .

Ainsi, nous employons cette représentation pour compresser l'information directionnelle que nous avons préalablement évaluée de manière exhaustive. Les coefficients C_l^m sont alors déterminés jusqu'à l'ordre de reconstruction n donné pour chaque échantillon spatial en évaluant l'intégrale de l'équation 8.12 à l'aide des échantillons directionnels précalculés. Cela peut être fait par une intégration de Monte-Carlo. Pour plus de détails à ce sujet, l'article de Green [61] peut être consulté : il s'agit d'un très bon manuel sur l'implantation logicielle des harmoniques sphériques pour une utilisation en informatique. Il faut cependant noter que nous avons adopté une autre tactique de compression, permettant d'accélérer drastiquement la détermination des coefficients C_l^m . Celle-ci est décrite en annexe C.

Empaquetage des données

Pour pouvoir être utilisés par le matériel graphique lors du rendu, les coefficients d'harmoniques calculés en chaque échantillon spatial sont stockés sous la forme d'images de texture. Nous avons déjà discuté de l'agencement du maillage 3D en espace texture. En reprenant cet agencement, nous allons générer $(n+1)^2$ images de texture, chacune d'elles étant associée à l'un des coefficients d'harmoniques $\{C_0^0, C_1^{-1}, C_1^0, C_1^1, \dots, C_n^{n-1}, C_n^n\}$, comme on peut le voir sur la figure 8.9. Ainsi, les coefficients d'harmoniques encodant l'information directionnelle pour un échantillon spatial donné peuvent être retrouvés en regardant les valeurs stockées dans le même pixel des $(n+1)^2$ images de texture. Il est à noter que le mécanisme

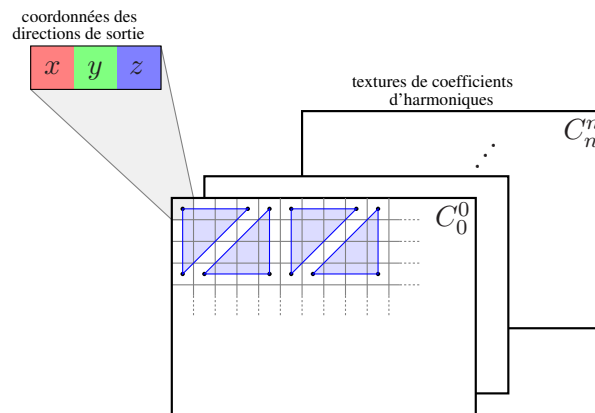


Figure 8.9 – Stockage des coefficients d'harmoniques dans des images de texture.

de correction de l'interpolation bilinéaire décrit dans le cas de la restitution d'œuvres d'art est également employé ici.

Étant donné que nous encodons une direction, qui est donc une donnée vectorielle dans \mathbb{R}^3 et non un scalaire, non pas une mais trois séries de coefficients d'harmoniques doivent être déterminées pour chaque échantillon spatial : une par coordonnées de ce vecteur. Ainsi, les séries de coefficients associées aux coordonnées x , y et z sont respectivement stockées dans les canaux R, G, et B des images de texture.

Notons, enfin, que nous aurons à nouveau besoin des repères locaux en chaque échantillon spatial. Ceux-ci sont également stockés sous forme de textures, en respectant toujours le même agencement.

■ 2.4 Algorithme de base pour le rendu temps réel

La phase de visualisation du champ lumineux surfacique est résumée sur la figure 8.10. Comme on peut le voir, elle se décompose en deux étapes :

1. **Pré-rendu** (ou phase de rendu hors écran) : cette étape a pour but de générer une texture dépendante du point de vue qui définit, pour l'ensemble de la surface du modèle 3D considéré, les directions de sortie des rayons d'observation. Cette étape utilise les textures de coefficients d'harmoniques créées précédemment, et passe par une évaluation à la volée des fonction harmoniques y_l^m par le matériel graphique.
2. **Rendu final** : la texture générée à l'étape précédente va être utilisée pour indexer une texture d'environnement. Le pixel atteint correspond alors à la couleur perçue par transparence, après réfraction à travers l'objet. D'autres effets peuvent alors être ajoutés, comme nous le verrons, pour produire la couleur finale de chaque pixel écran.

Nous allons maintenant détailler les différents points de ce pipeline de rendu. Celui-ci est, pour le moment, volontairement simplifié. Nous allons effectivement voir que des étapes intermédiaires peuvent (ou doivent, pour certaines) être ajoutées pour améliorer soit les performances, soit la qualité du rendu. Une version complète de ce pipeline sera donc présentée

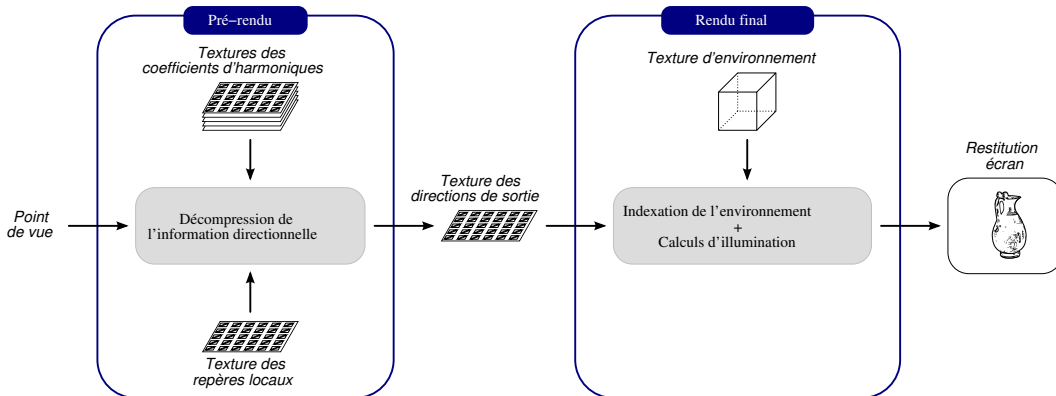


Figure 8.10 – Principe général de l'algorithme utilisé pour le rendu de la réfraction à l'aide de champs lumineux surfaciques compressés par harmoniques sphériques.

ultérieurement.

Décompression à la volée des harmoniques sphériques

La formulation des fonctions harmoniques présentée dans l'équation 8.8 est relativement compacte, mais elle se prête mal à une décompression à la volée par le matériel graphique pour les raisons suivantes :

- le polynôme de Legendre associé est défini de manière récursive,
- il fait appel à des fonctions trigonométriques qui risquent d'être coûteuses,
- la sphère S^2 est paramétrée par des coordonnées sphériques alors que les processeurs graphiques manipulent préférentiellement des vecteurs.

En utilisant le changement de coordonnées défini par l'équation 8.14 :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{bmatrix} \quad \text{avec} \quad \begin{cases} \theta \in [0, \pi] \\ \phi \in [0, 2\pi[\end{cases} \quad (8.14)$$

il est possible de récrire les fonctions harmoniques Y_l^m à l'aide de coordonnées Cartésiennes, ce qui aboutit à la formulation de l'équation 8.15 :

$$Y_l^m(x, y, z) = K_l^m \sum_{p,q,s} \frac{1}{p!q!s!} \left(-\frac{x+iy}{2} \right)^p \left(\frac{x-iy}{2} \right)^q z^s$$

$$\text{avec} \quad \begin{cases} p, q, s \in \mathbb{N} \\ p + q + s = l \\ p - q = m \end{cases} \quad (8.15)$$

Grâce à cette nouvelle écriture, toutes les harmoniques sont représentées par des polynômes dépendants des variables x , y et z qui peuvent être facilement évalués par le processeur graphique. Nous utilisons la version réelle de ces polynômes, qui peut être obtenue à partir de la définition précédente grâce à l'équation 8.10.

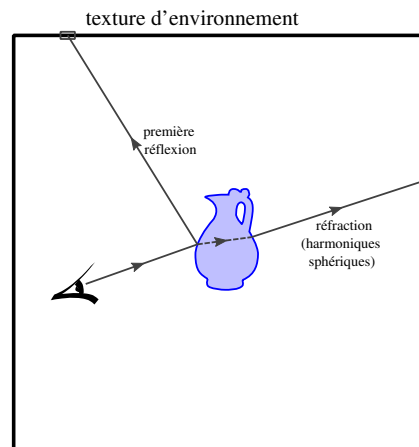


Figure 8.11 – Rendu final. Les couleurs obtenues par indexation de la texture d'environnement pour la direction de réflexion (première réflexion uniquement) et de réfraction (obtenue par la décompression des harmoniques sphériques) sont combinées linéairement à l'aide de leurs coefficients de Fresnel respectifs.

Il faut noter que les polynômes représentant les harmoniques d'ordres plus élevés demandent plus de calculs pour être évalués. Or, le nombre d'instructions pouvant être exécutées par le processeur graphique au cours d'une seule passe de rendu est limité. Ainsi, sauf pour des ordres de reconstruction très petits, tous les polynômes ne peuvent pas être évalués en une fois. Nous devons donc effectuer plusieurs passes de rendu. Pour ce faire, nous utilisons un algorithme glouton qui construit le programme à charger dans le processeur graphique pour chaque passe de rendu en y empaquetant le plus grand nombre de polynômes possible, réduisant ainsi le nombre de passes nécessaires. Étant donné que la reconstruction finale correspond à une combinaison linéaire de ces polynômes avec les coefficients stockés en texture, il nous suffit alors d'accumuler (additionner) le résultat de ces passes successives de rendu⁴.

La décompression de l'information directionnelle se fait donc en chaque échantillon spatial de la manière suivante :

1. récupérer le repère local de l'échantillon spatial à partir de la texture correspondante,
2. déterminer la direction locale d'observation dans ce repère,
3. utiliser cette direction pour évaluer les polynômes de l'équation 8.15, en une ou plusieurs passes, et effectuer la combinaison linéaire avec les coefficients d'harmoniques stockés en textures,
4. écrire le vecteur ainsi obtenu dans la texture cible.

La texture cible en question contiendra donc, *in fine*, les directions des rayons sortant de l'objet pour chaque point de la surface. Cette texture est dynamique en ce sens que son contenu est systématiquement recalculé en fonction de la position courante du point de vue.

⁴Il faut, bien entendu, noter que le matériel a évolué depuis la réalisation de ces travaux, et que les restrictions actuelles sur la quantité d'instructions exécutables en une seule passe sont certainement beaucoup moins contraignantes.

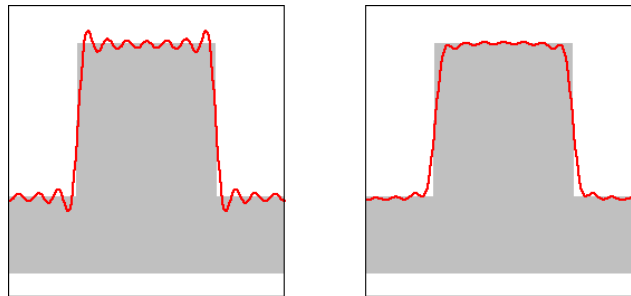


Figure 8.12 — Gauche : phénomène de Gibbs observable dans le cas des séries de Fourier. Droite : atténuation du phénomène de Gibbs par application d'un filtre semi-Gaussien sur la série de fonctions.

Rendu final

Une fois que la texture définissant les directions de sortie est disponible, elle peut être utilisée pour produire le rendu final. Pour chaque pixel écran couvert par le modèle 3D, cette texture est indexée pour trouver la direction de sortie correspondante, qui est alors utilisée à son tour pour indexer une texture d'environnement (*cubemap*).

Afin d'augmenter le réalisme visuel, notre algorithme effectue également un calcul d'illumination local en chaque pixel tenant compte de la réflexion issue du premier impact du rayon d'observation avec la surface. La direction du rayon réfléchi est alors calculée sur le processeur graphique, et utilisée pour indexer une nouvelle fois la texture d'environnement. Les couleurs obtenues par réflexion et réfraction sont alors combinées linéairement à l'aide de leurs coefficients de Fresnel respectifs dans le but de produire la couleur finale du pixel, comme l'illustre la figure 8.11.

■ 2.5 Améliorations

Filtrage fréquentiel

Étant donné que la série infinie de l'équation 8.11 doit être tronquée dans le cas pratique, la qualité de la reconstruction s'en trouve nécessairement altérée, comme nous l'avons déjà évoqué précédemment. Cette suppression des hautes fréquences aboutit à des oscillations dans le signal reconstruit, appelées «phénomène de Gibbs», qui sont identiques à celles que l'on peut observer lorsque l'on tente de représenter un signal carré à l'aide d'une série de Fourier (figure 8.12-gauche).

Les artefacts visuels engendrés par ce phénomène peuvent être atténués en appliquant un filtre passe-bas semi-Gaussien sur la série d'harmoniques (figure 8.12-droite) plutôt que de la tronquer purement et simplement [196]. Nous avons donc implanté un tel filtrage. S'il est appliqué directement sur les coefficients d'harmoniques lors du précalcul, son impact sur les performances de rendu devient alors nul.

Le réglage des paramètres de ce filtre semi-Gaussien demande de trouver un bon compromis entre l'atténuation des oscillations et la préservation du signal reconstruit. Ce réglage est actuellement effectué de manière empirique, et dépend totalement de la complexité du signal (liée à la complexité de la géométrie de l'objet dans notre cas) et de l'ordre de reconstruction utilisé.

Filtrage spatial

Dans la mesure où nous faisons appel à une méthode de compression à perte, du bruit peut apparaître dans le rendu final de la réfraction, même après correction du phénomène de Gibbs. Pour éviter ce problème, il est également possible d'appliquer un filtre Gaussien à la texture des directions de sortie une fois que celle-ci a été créée par décompression des harmoniques. Même si une perte de précision est nécessairement induite par ce filtrage, nous avons tout de même remarqué que les résultats étaient beaucoup plus plaisants à l'œil. Nous illustrerons les effets de ces deux filtrages (spatial et fréquentiel) dans la section résultats (section 2.7).

Il faut noter que l'application du filtrage spatial n'est pas immédiate : en effet, le stockage éclaté en espace texture nous fait perdre les relations d'adjacence entre les échantillons spatiaux. Nous avons donc ajouté une phase de précalcul qui détermine, pour chaque échantillon, l'ensemble des échantillons voisins. Cette relation de voisinage, ainsi que le poids Gaussien à appliquer à chaque voisin, est alors elle-même stockée dans une texture, et utilisée pour effectuer le filtrage spatial sur la carte graphique. L'emplacement dans le pipeline de rendu où ce filtrage est introduit est illustré sur la figure 8.13.

Réarrangement des échantillons spatiaux

On peut remarquer, toujours à cause de notre éclatement du maillage triangulaire en espace texture, que de nombreux échantillons spatiaux parmi tous ceux générés sont redondants (en fait, tous ceux situés sur les arêtes des triangles). Pour éviter de multiplier les traitements, sachant que l'évaluation des fonctions harmoniques par le matériel graphique est relativement coûteuse, la décompression n'est pas directement faite dans cette texture. Nous utilisons d'abord une texture dans laquelle chaque échantillon n'est représenté qu'une seule fois. L'évaluation des harmoniques se fait alors dans cette texture, qui est ensuite expansée de manière à générer la texture finale (agencement éclaté par triangle dont nous avons parlé jusqu'à présent) à l'aide d'une texture supplémentaire d'indirection. Cette suppression de la redondance est en réalité introduite dès le début de la chaîne de traitements, de manière à ce que l'échantillonnage exhaustif par lancer de rayons en bénéficie également.

Le pipeline complet de rendu, intégrant toutes ces modifications, se présente alors comme sur la figure 8.13.

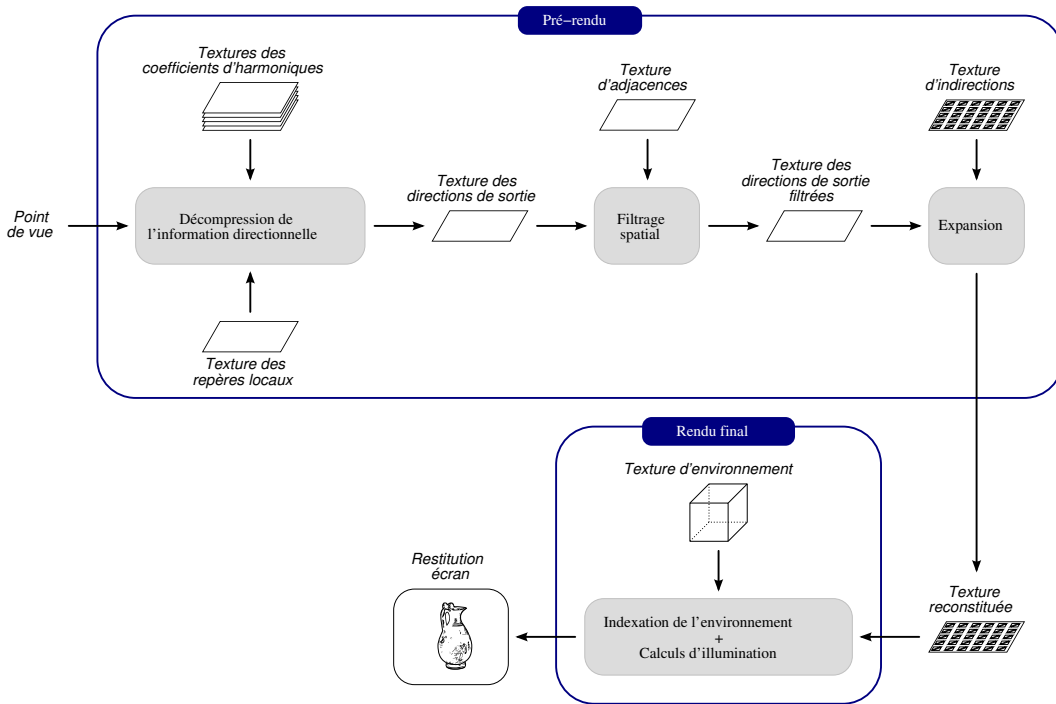


Figure 8.13 – Principe détaillé de l'algorithme utilisé pour le rendu de la réfraction à l'aide de champs lumineux surfaciques compressés par harmoniques sphériques.

■ 2.6 Gestion de l'opacité

L'opacité d'un matériau translucide homogène peut être définie à partir de sa *profondeur optique* τ , elle-même dépendante de la distance d parcourue par la lumière à l'intérieur de l'objet :

$$\tau = K d \quad (8.16)$$

où K est alors appelé *coefficient d'extinction*, sa valeur étant propre à chaque matériau.

L'opacité est alors définie de la façon suivante. Soient I_0 l'intensité lumineuse émise depuis un point source, et I la luminance perçue à une distance d de ce point. L'atténuation subie par la lumière tout au long de cette distance est alors caractérisée par l'équation suivante :

$$I = I_0 e^{-\tau} = I_0 e^{-Kd} \quad (8.17)$$

Ainsi, en fixant un coefficient d'extinction K à notre objet, la couleur de l'environnement réfractée vers l'œil peut facilement être modulée grâce à cette équation dès lors que la distance parcourue dans l'objet est connue. Celle-ci peut aisément être déterminée lors du précalcul par lancer de rayons, puis compressée comme une information directionnelle en chaque point de la surface, au même titre que les directions de sortie.

Par ailleurs, le stockage de cette information de distance n'implique aucune contrainte supplémentaire : les coefficients d'harmoniques encodant la variation des directions de sortie étant, pour le moment, stockées dans des textures RVB, le canal alpha de chaque texture est

actuellement libre, et peut donc être utilisé pour y placer les coefficients de cette nouvelle information. De la même façon, aucun coût supplémentaire n'est requis pour sa reconstruction, étant donné que les fonctions harmoniques doivent de toute façon être évaluées.

■ 2.7 Résultats

Résultats visuels et comparaisons

La figure 8.14 fournit différents exemples d'objets rendus à l'aide de la méthode qui vient d'être présentée. Comme on peut le voir, des objets de géométries quelconques peuvent être restitués. On peut d'ailleurs y voir le modèle de la tasse, composé de plusieurs milieux différents : le verre de la tasse, ainsi que l'eau qu'il contient. Notre méthode permet de tenir compte de ce changement de milieu, contrairement aux méthodes orientées rendu temps réel existant au moment du développement de cette approche.

La figure 8.15 illustre, quant à elle, l'effet obtenu grâce à l'opacité dépendant du point de vue pour différentes valeurs du coefficient d'extinction. Comme on peut le voir, l'opacité devient plus importante au niveau de la perle contenue dans le coquillage, c'est-à-dire dans la région où la distance traversée par le rayon d'observation à l'intérieur de l'objet est la plus importante.

La figure 8.16 permet de comparer nos résultats à d'autres méthodes existantes, lesquelles sont les suivantes :

- le lancer de rayons, qui nous sert en quelque sorte de vérité terrain,
- une approximation très couramment utilisée pour les applications temps réel, consistant à ne considérer que la déviation se produisant entre le rayon d'observation et la première surface qu'il rencontre,
- la méthode récemment présentée par Wyman [200], considérant deux déviations successives : l'une pour la toute première surface rencontrée et la seconde pour la toute dernière.

À l'exception de notre méthode (et, bien entendu, du lancer de rayons), la perle se trouvant à l'intérieur du coquillage n'est pas du tout prise en compte par la simulation du phénomène de réfringence. Cet objet constitue, en effet, un jeu de données test particulièrement difficile pour les algorithmes de rendu de réfraction, du fait des différentes «couches» que doivent traverser les rayons lumineux. Même la méthode de Wyman est ici mise en défaut par la concavité importante du coquillage, ce qui en révèle les limites pour des géométries complexes. À l'inverse, même si notre méthode n'atteint pas la qualité visuelle du lancer de rayons, et ce du fait que les harmoniques hautes fréquences sont ignorées, elle permet néanmoins de prendre en compte les objets de ce genre, où certaines parties de la géométrie peuvent être complètement occultées.

De même, puisque nous nous basons sur un précalcul intensif, des effets particuliers peuvent être observés, comme dans le cas de la tasse de la figure 8.17-gauche, où des bandes circulaires apparaissent au niveau de la paroi lorsque le point de vue est proche de la verticale. Ces bandes, que l'on observe également dans la réalité (figure 8.17-milieu), sont dues à des

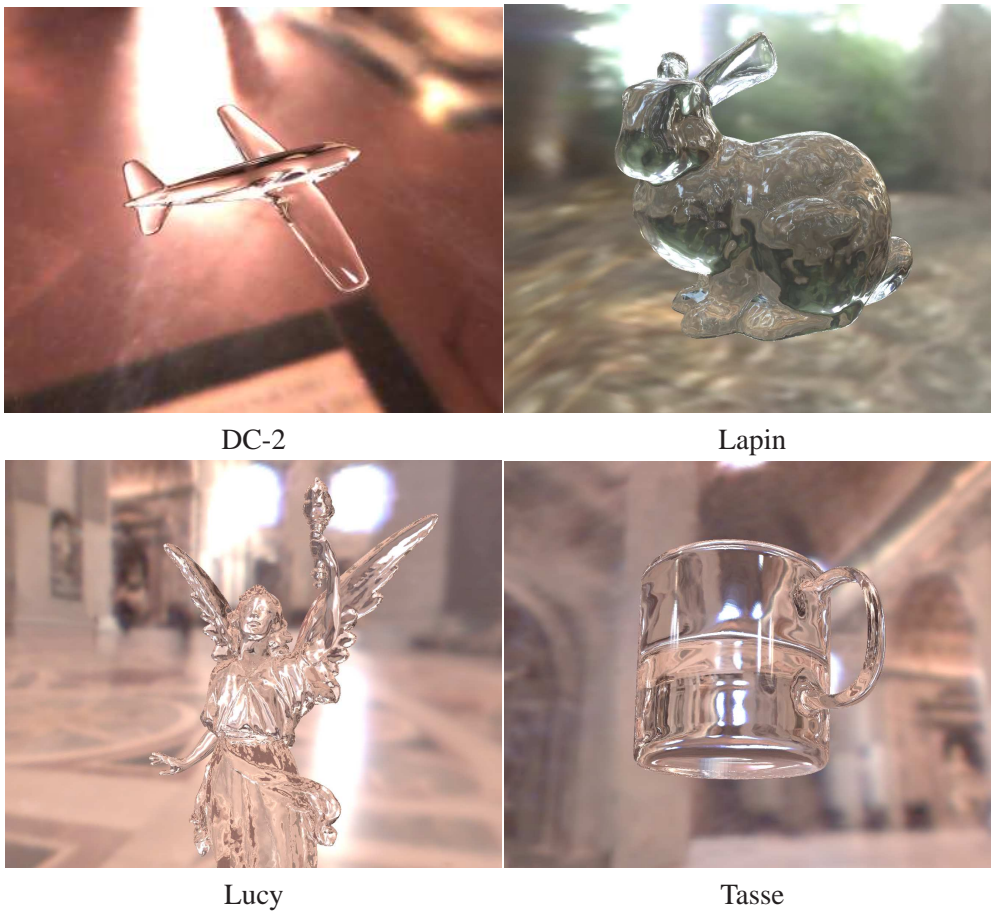


Figure 8.14 – Exemples d'objets réfringents restitués à l'aide de notre méthode.

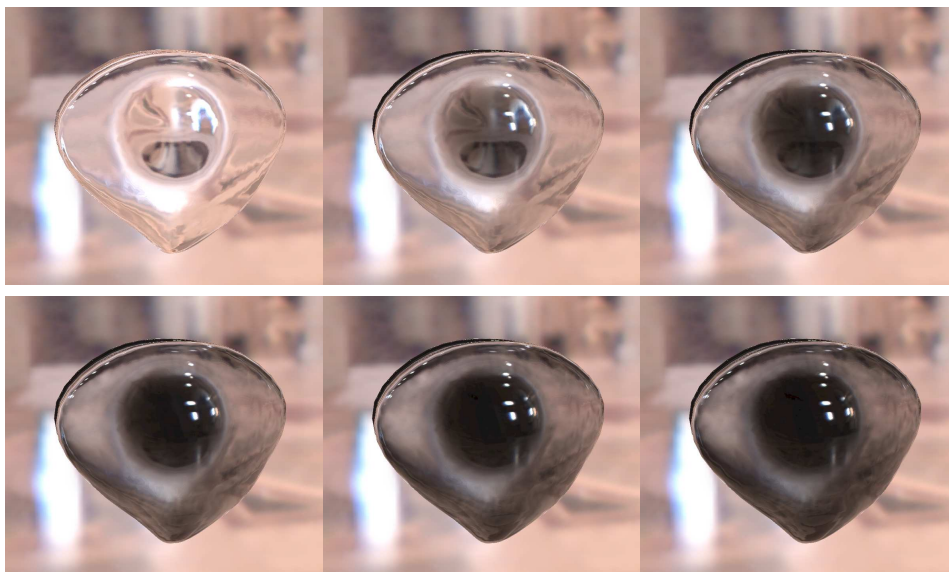


Figure 8.15 – Différentes opacités obtenues pour le modèle du coquillage en jouant sur le coefficient d'extinction.

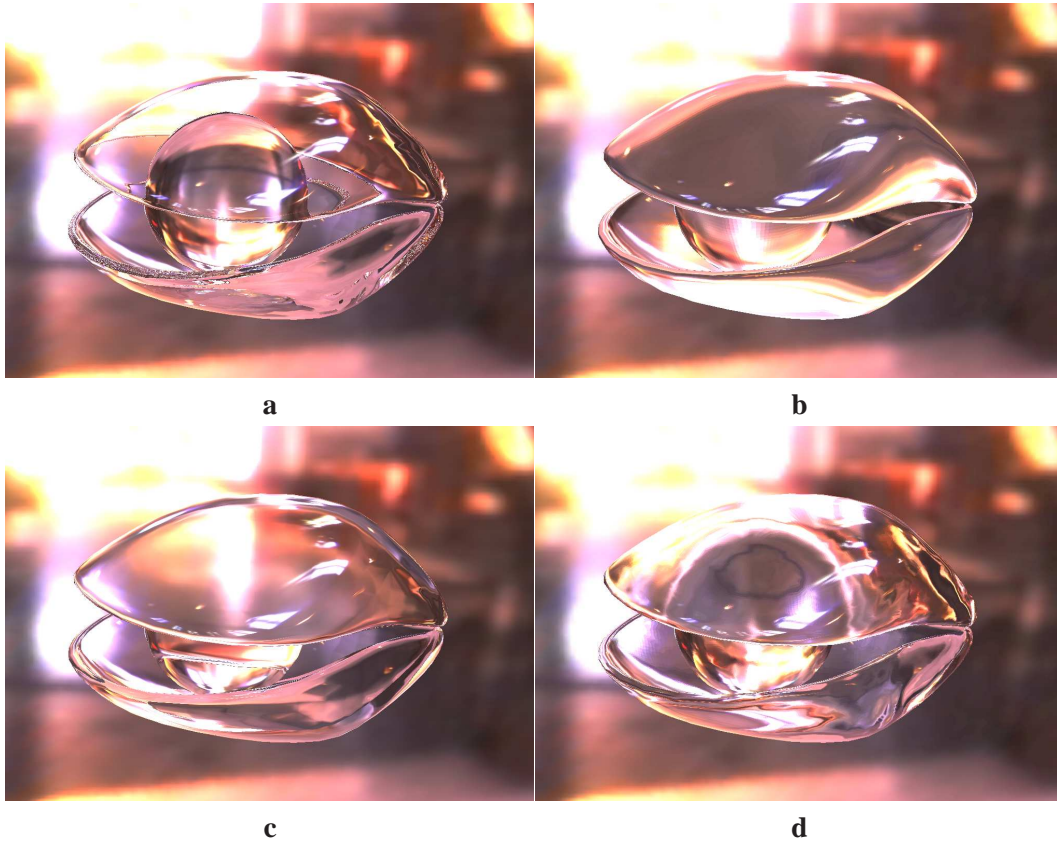


Figure 8.16 – Comparaison aux méthodes existantes. a) Lancer de rayons. b) Approximation par une seule déviation. c) Réfraction double de Wyman [200]. d) Notre méthode basée sur les harmoniques sphériques (reconstruction d'ordre $n = 8$).

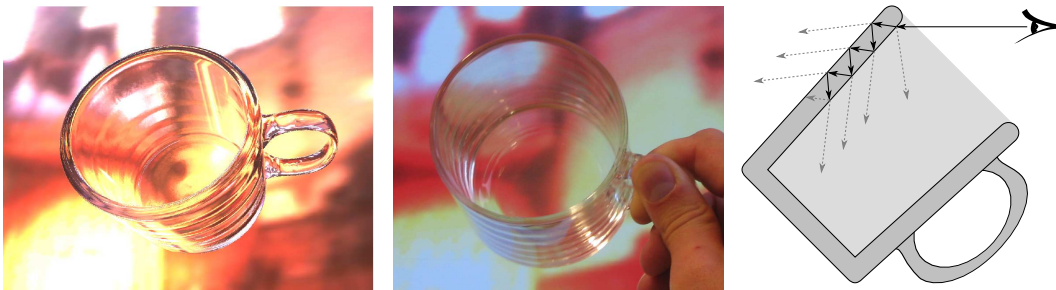


Figure 8.17 – Effets visuels liés aux réflexions internes des rayons lumineux dans l'objet. Le fait que notre méthode se base sur un précalcul intensif nous permet de les simuler.

Modèle	# triangles	# échantillons spatiaux	Consommation mémoire (Mo)	Fréquence d'affichage (Hz)
Coquillage	4288	34K	17	52
Lapin	69451	35K	35	40
Tasse	9184	42K	24	43
Théière	2256	41K	23	45
Lucy	238734	119K	121	7

Tableau 8.2 – Statistiques de rendu pour différents exemples. Dans tous les cas, les harmoniques sphériques sont utilisées jusqu'à l'ordre 8 (91 coefficients par échantillon spatial).

Modèle	# triangles	# échant. spatiaux	Ordre 2		Ordre 5		Ordre 8	
			Mém. (Mo)	Fréq. (Hz)	Mém. (Mo)	Fréq. (Hz)	Mém. (Mo)	Fréq. (Hz)
DC-2	3874	7K	2,5	190	3,3	165	4,5	126
Lapin	69451	35K	26,2	44	29,2	39	35,9	30
Lucy	238734	119K	90,0	18	102,4	15	122,9	11

Tableau 8.3 – Consommation mémoire et performances en fonction de l'ordre de reconstruction (la consommation mémoire inclut la géométrie du modèle).

effets de réflexions internes (figure 8.17-droite) qui ne peuvent pas être pris en compte par des méthodes telles que celles auxquelles nous venons de nous comparer.

Étude des performances et de la consommation mémoire

Les mesures de performances qui vont être présentées ont été obtenues sur un ordinateur de bureau doté d'un processeur AMD Athlon XP1800+ et d'une carte graphique ATI Radeon 9700 Pro 128Mo. Les résultats sont reportés dans le tableau 8.2.

Pour tous ces exemples, un ordre 8 (81 coefficients) a été utilisé pour l'évaluation des harmoniques sphériques, et les dimensions de la fenêtre d'affichage sont de 1350×1100 pixels. Cette seconde donnée n'est cependant pas aussi importante que dans le cas d'autres techniques de rendu, étant donné que l'évaluation des harmoniques est effectuée en espace texture. Le nombre de traitements ne dépend donc pas du nombre de pixels écran couvert par l'objet affiché, mais du nombre d'échantillons spatiaux générés sur sa surface lors des prétraitements. Comme on peut le voir, les vitesses de rendu obtenues restent interactives, malgré l'ordre de reconstruction élevé.

En effet, cet ordre de reconstruction a un impact sur les performances : plus l'ordre est élevé, plus le nombre de polynômes à calculer pour évaluer les fonctions harmoniques sera important (et plus ces polynômes seront gros, d'ailleurs). Le tableau 8.3 nous donne la consommation mémoire et les performances de rendu pour différents objets et pour différents ordres de reconstruction. Étant donné que le nombre de coefficients à stocker est

quadratique en fonction de l'ordre, la consommation mémoire augmente de manière non négligeable lorsque l'ordre devient élevé. Il faut d'ailleurs noter que la mémoire requise est de la mémoire vidéo (stockage direct sur la carte graphique), ce qui est assez problématique pour des objets volumineux dans la mesure où la quantité de mémoire vidéo est généralement moindre par rapport à la quantité de mémoire centrale. On peut notamment voir que le modèle Lucy consomme déjà plus de 120Mo à lui seul, alors qu'il n'est constitué que de 238 000 triangles.

Ce constat doit cependant être modéré : il est en effet important de constater qu'il y a un surcoût mémoire engendré par notre schéma de stockage éclaté (texture d'indirection, texture d'adjacence, etc.). Entre 30 et 40% de la consommation mémoire est dédiée au stockage des données auxiliaires nécessaires à la bonne mise en place de ce schéma. Il est évident que si une bonne paramétrisation du maillage en espace texture peut être trouvée, la taille des données pourrait être réduite. Évidemment, cela n'enlève rien au fait que la méthode reste gourmande en ressources en tous genres.

Étude de l'impact visuel de l'ordre de reconstruction

La figure 8.18 montre l'exemple du coquillage restitué pour différents ordres de reconstruction. Comme on peut le voir, l'utilisation des coefficients d'harmoniques de plus hautes fréquences est indispensable, sans quoi les déviations induites par la perle disparaissent totalement, ou ne sont que trop grossièrement approximées.

Mais cela signifie également que pour des objets très simples (de géométries relativement convexes), des ordres de reconstruction très bas peuvent être utilisés. On peut s'en convaincre grâce à la figure 8.19, où il devient difficile de dire, sans élément de comparaison, dans quelle mesure l'effet de réfraction est bien approximé ou non. Le phénomène de réfringence est en fait si difficile à appréhender que ses résultats visuels ne peuvent être prédits avec certitude : autant sommes-nous en mesure d'affirmer que la perle du coquillage est ignorée, autant il est impossible de dire à quoi est censé ressembler la thère dans la réalité.

En ce sens, le phénomène n'a pas nécessairement besoin d'être systématiquement simulé avec la même précision : il doit être adapté au cas. Pour ce faire, les harmoniques sphériques sont un bon outil, puisqu'elles permettent une reconstruction progressive en fonction de l'ordre demandé.

Étude du filtrage

Nous avons précédemment introduit deux types de filtrage : un filtrage fréquentiel, agissant sur les coefficients d'harmoniques, et un filtrage spatial, consistant à appliquer un filtre Gaussien aux directions de sortie obtenues après décompression de l'information directionnelle.

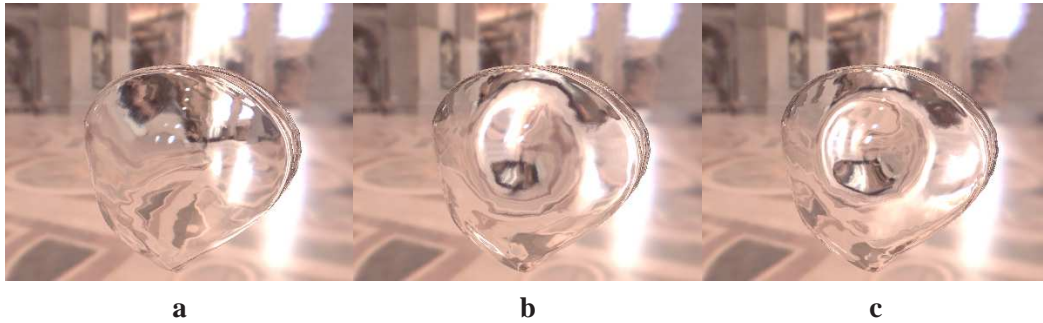


Figure 8.18 – Différents ordres de reconstruction dans le cas d'un objet complexe. a) Ordre 1. b) Ordre 4. c) Ordre 8. L'ajout des coefficients d'ordre plus élevés est essentiel à la capture des déviations hautes fréquences.



Figure 8.19 – Différents ordres de reconstruction dans le cas d'un objet «simple». a) Ordre 1. b) Ordre 4. c) Ordre 8. L'impact de l'ordre de reconstruction sur l'aspect visuel n'est pas aussi flagrant que dans le cas précédent.



Figure 8.20 – Impact visuel du filtrage. **Gauche** : pas de filtrage. **Milieu** : filtrage fréquentiel, uniquement. **Droite** : filtrage fréquentiel et spatial.

Comme le montre la figure 8.20-gauche, le rendu présente du bruit lorsque aucun des deux filtres n'est activé. Nous l'avons déjà dit : la principale cause de ce bruit est le phénomène de Gibbs, dû au fait que la série infinie permettant de reconstruire une fonction à partir des harmoniques doit nécessairement être tronquée. Ainsi, le filtre fréquentiel permet d'éliminer la plus grande partie de ce bruit, comme le montre la figure 8.20-milieu. L'ajout du filtre spatial (figure 8.20-droite), dont la justification est cette fois d'ordre purement empirique, permet cependant d'adoucir davantage l'information reconstruite.

On peut néanmoins remarquer que si des contraintes de performances sont imposées, l'application de ce second filtre peut éventuellement être retirée du rendu, dans la mesure où il n'apporte pas nécessairement un gain aussi significatif que le premier.

On pourra également remarquer, sur certaines des images fournies dans cette section, des effets prononcés de scintillement ou d'aliassage. Ces effets, beaucoup plus visibles quand l'objet est en mouvement, apparaissent en fait lorsque la densité de l'échantillonnage spatial devient supérieure à la densité de pixels de l'écran. Ce genre de problèmes est généralement résolu à l'aide de techniques de filtrage câblées, exploitant plusieurs niveaux de résolutions de textures (*mipmapping*). Ces techniques sont cependant sans effet dans notre cas : la texture finale des directions de sortie est utilisée comme une texture d'indirection, pour indexer un environnement. De ce fait, lui appliquer un tel filtrage ne change rien au fait que deux pixels écran voisins peuvent servir à indexer des zones très éloignées l'une de l'autre dans la texture d'environnement, ce qui est justement la cause du scintillement dans notre cas. Ceci constitue en réalité le principal défaut de notre méthode, d'autant plus que nous n'avons pour le moment aucune idée de la façon dont il pourrait être résolu, à moins de l'effectuer de manière explicite au risque d'entraîner un surcoût supplémentaire important.

3 Bilan

Dans ce chapitre, nous avons présenté deux applications des champs lumineux surfaciques. La première est une utilisation de ces champs au sens classique du terme, c'est-à-dire pour représenter la luminance quittant la surface d'un objet placé dans un environnement lumineux fixe. L'objectif est alors de permettre la visualisation des copies numériques que nous avons mesurées et reconstruites au cours des chapitres précédents. Pour ce faire, nous avons utilisé un modèle de textures polynomiales pour représenter la dépendance au point de vue. Ce modèle a l'avantage d'être simple, et permet une visualisation temps réel tout en ne nécessitant que peu de ressources (autant en termes de temps de calcul que de consommation mémoire).

Bien que les textures polynomiales soient connues pour ne permettre de représenter que des variations basses fréquences, nous avons constaté, au cours de cette application, que cela suffisait pour donner une bonne intuition du comportement de la surface face aux déplacements de l'observateur. Une bonne intuition, certes, car les variations en question ne sont

pas forcément fidèles à la réalité : les textures polynomiales tendent à lisser drastiquement l'information. De plus, il faut remarquer que les exemples sur lesquels nous avons travaillé constituent un cadre plus favorable pour ce modèle dans la mesure où seuls des environnements lumineux se limitant à une unique source lumineuse réellement significative ont été considérés. Ainsi, le modèle n'a besoin d'encoder, en chaque point de la surface, qu'un seul lobe spéculaire. Pour des environnements lumineux plus complexes, les limitations imposées par ce modèle risquent d'être beaucoup plus préjudiciables.

Au vu de ces résultats, nous sommes forcément en droit de nous poser des questions quant à la validité de ce modèle pour le cas de la visualisation d'œuvres d'art : peut-on se satisfaire d'une simple « intuition » de l'aspect visuel pour de tels objets ? Un modèle permettant une plus grande précision est largement préférable si l'objectif est de convaincre des visiteurs virtuels potentiels.

De plus, ces résultats ont à nouveau soulevé un point déjà abordé dans le bilan du chapitre 6 : la nécessité d'un algorithme permettant l'ajustement des photographies à partir desquelles la texture doit être reconstruite. Nous avons en effet observé des effets de flou dans la texture finale. Quelque soit la méthode utilisée pour représenter le champ lumineux surfacique, de tels effets seront systématiquement présents si un tel ajustement n'est pas envisagé. Les travaux de Bernardini *et al.* [10] peuvent alors constituer un bon point de départ pour le mettre en œuvre.

Enfin, nous avons également constaté que le modèle 3D fourni par la fusion des acquisitions géométriques est généralement bien trop gros pour convenir en l'état à une visualisation temps réel. Le goulet d'étranglement semble, en effet, se situer au niveau du nombre de primitives géométriques qu'il est nécessaire de faire traiter par la carte graphique. Cela met en exergue la nécessité d'ajouter à la chaîne d'acquisition/reconstruction une étape de simplification de maillages. La densité de l'information géométrique recueillie par les dispositifs de mesures de formes actuels est en effet bien supérieure à ce qu'il est nécessaire de conserver pour une application de visualisation temps réel.

La seconde application des champs lumineux surfaciques que nous avons proposée est la visualisation temps réel d'objet réfringents. La composante directionnelle du champ est alors une application qui associe à chaque rayon d'observation incident la direction du rayon quittant l'objet après en avoir parcouru l'intérieur. Elle est construite par un précalcul intensif à l'aide d'un lancer de rayons simplifié, qui choisit, à chaque interface, celui du rayon réfléchi ou réfracté dont le coefficient de Fresnel est le plus élevé plutôt que de considérer l'arborescence complète des rayons lumineux. Cet échantillonnage exhaustif est finalement compressé par projection dans la base des harmoniques sphériques, grâce auxquelles l'information pour tout l'hémisphère d'observation peut être approximée à l'aide de quelques coefficients réels, seulement. La visualisation temps réel est finalement obtenue grâce à une décompression à la volée de ces fonctions harmoniques par le matériel graphique, permettant de retrouver les directions de sorties, pour tout l'objet, en fonction du point de vue courant. Connaissant ces directions de sortie, une texture représentant l'environnement peut être indexée afin de déterminer quelle est la couleur de cet environnement qui est perçue par transparence.

Bien que coûteuse en temps de calcul et en consommation mémoire, cette méthode est capable de traiter avec des objets de géométries complexes, ou composés de milieux d'indices de réfraction différents, contrairement à la plupart des méthodes récentes orientées rendu temps réel. Les résultats produits sont visuellement convaincants. Elle présente cependant un certain nombre de limitations :

- le fait que la méthode se base sur un précalcul interdit l'usage d'objets déformables. Nous devons donc nous cantonner à la représentation de géométries statiques.
- à l'instar de toutes les méthodes utilisant des textures d'environnement, nous partons de l'hypothèse que l'environnement en question est situé à une distance infinie de l'objet. C'est grâce à cette hypothèse qu'il est en effet possible de ne considérer que la direction des rayons de sortie tout en ignorant leurs positions. Il n'est donc pas possible, avec notre méthode, de gérer le proche voisinage de l'objet visualisé. Une approximation visant à calculer la texture d'environnement à partir d'une boîte positionnée autour de l'objet, comme cela est souvent fait, est bien entendu envisageable, mais elle ne permettra en aucun cas de tenir compte des effets de parallaxe liés aux distances relatives des objets avoisinants.
- la dernière limitation, très contraignante pour la qualité visuelle, est l'impossibilité d'utiliser des techniques de *mipmapping* pour filtrer les textures en fonction de la distance de l'objet au point de vue. Cette impossibilité provient du fait que la texture finale n'est pas directement employée pour produire une couleur, mais sert d'indirection pour indexer l'environnement. Ainsi, lorsque l'échantillonnage spatial de la texture devient plus dense que la résolution de l'écran, des phénomènes de scintillement surviennent.

Concernant ce dernier point, il semble que la résolution de ce problème de scintillement n'est malheureusement pas possible à l'heure actuelle (contraintes matérielles liées aux fonctionnalités câblées des cartes graphiques).

Pour ce qui est de la gestion de la parallaxe, l'encodage de l'environnement sous forme d'un champ lumineux [73] ou l'utilisation de textures d'environnements multiples [66] peuvent être envisagés. Quelque soit la technique employée, il faudra nécessairement déterminer la position des rayons de sortie en plus de leurs orientations.

Pour ce qui est de la gestion des objets déformables, il faut commencer par noter la similitude de notre approche avec les travaux de Sloan *et al.* [170] : dans les deux cas, les harmoniques sphériques sont utilisées pour représenter une information directionnelle à la surface d'un objet. Cependant, ces deux travaux diffèrent au niveau de l'usage qu'il est fait de ces harmoniques : l'approche de Sloan *et al.* exploite l'orthogonalité de la base pour effectuer une convolution entre deux fonctions sphériques à l'aide d'un simple produit scalaire entre les vecteurs de coefficients qui les représentent, alors que dans notre cas, cette propriété d'orthogonalité n'est absolument pas considérée : nous effectuons directement l'évaluation des fonctions harmoniques. Quoi qu'il en soit, la géométrie est considérée comme statique dans les deux cas. Cependant, Sloan *et al.* ont proposé une extension de leur méthode permettant de tenir compte de déformations locales de la géométrie [171]. Il serait donc intéressant de voir si une telle approche peut également être mise en œuvre dans notre cas.

Enfin, remarquons que la formulation des fonctions harmoniques que nous avons présentées n'est prévue que pour la représentation de données scalaires. Or, les applications en informatique graphique utilisant des harmoniques sphériques travaillent souvent avec des données vectorielles (que ce soit pour représenter une direction, comme dans notre cas, ou bien une couleur dans un espace trichromatique quelconque). Ces «harmoniques scalaires» sont alors utilisées de manière indépendante sur chacune des composantes des vecteurs manipulés. Il existe pourtant une forme vectorielle des harmoniques sphériques. Celle-ci demande donc d'être étudiée, afin de voir si un gain quelconque (en termes de précision ou de stockage) peut être obtenu grâce à une telle représentation. Les nombreuses applications en informatique graphique pourraient alors, elles aussi, en bénéficier.

Les deux applications que nous avons présentées utilisent donc des modèles de champs lumineux surfaciques différents. Nous n'avons cependant pas confronté ces deux modèles l'un à l'autre, principalement parce qu'ils ne répondent pas aux mêmes objectifs. Dans le cas des œuvres d'art, nous avons opté pour un modèle simple permettant une visualisation très rapide tout en donnant une bonne intuition des variations lumineuses se produisant à la surface des objets (même si celles-ci ne sont pas totalement fidèles à la réalité). De plus, l'intensité lumineuse est une information qui varie de manière relativement douce, à condition qu'aucun objet réfléchissant ne soit présent dans la scène (ce qui est le cas pour nous, étant donné que ce type d'objets ne peut pas être numérisé par la technologie que nous employons). À l'inverse, dans le cas de la réfraction, l'information encodée a un comportement beaucoup plus chaotique, présentant des variations potentiellement très hautes fréquences. Les harmoniques sphériques sont donc plus adaptées, même si pour cela un ordre de reconstruction élevé doit être utilisé.

Un dernier mot, enfin, sur le schéma éclaté de rangement des triangles en espace texture. Celui-ci a bien entendu été utilisé pour éviter d'avoir à se soucier de problèmes de paramétrisations de maillages. Ce problème, non trivial en soi, n'étant pas notre préoccupation première lors de la réalisation de ces travaux, nous ne nous y sommes pas particulièrement attachés. Cependant, si un tel algorithme est disponible, il peut évidemment être utilisé pour obtenir un rangement mieux optimisé, et ainsi éviter les traitements et le stockage supplémentaires auxquels notre schéma «simplifié» aboutit. Il faut cependant noter que ces algorithmes sont souvent obligés de découper les objets afin de pouvoir les déplier pour les mettre à plat. Les arêtes au niveau desquelles à lieu ce découpage sont alors définies en deux endroits différents de l'image de texture. Si la densité et la répartition de l'échantillonnage n'est pas la même de part et d'autre, des discontinuités peuvent apparaître pour ces arêtes au moment de la visualisation, du fait, une fois encore, de l'interpolation bilinéaire employée pour indexer la texture. Nous avons constaté que l'utilisation de *gouttières* pour étendre le maillage en espace texture, solution couramment utilisée dans le domaine pour palier à ce problème, n'éliminait pas totalement ces problèmes de continuité. Ces problèmes sont d'ailleurs davantage visibles du fait que l'information que nous représentons n'est pas statique, mais dépendante du point de vue. En utilisant notre schéma, tous les triangles sont plaqués sur le même nombre de pixels, garantissant que l'échantillonnage de la texture pour deux triangles situés de part et d'autre d'une même arête est bien le même. C'est pourquoi aucun problème de continuité n'est visible dans notre cas, malgré les traitements supplémentaires que cela demande.

- Cas d'étude - Numérisation de tableaux

Chaîne de traitements dédiée à la numérisation et au rendu réaliste de peintures d'art

Au cours des chapitres précédents, nous avons abordé le problème de l'acquisition et de la visualisation d'objets réels dans le cas le plus général possible, c'est-à-dire en réduisant au maximum le nombre d'hypothèses de départ. Malgré les quelques solutions que nous avons tenté d'apporter aux problématiques que cela implique, la chaîne complète de traitements demeure particulièrement complexe, comme aura pu en juger le lecteur. La nécessité d'apporter des simplifications à différents niveaux de cette chaîne s'impose d'elle-même, et c'est d'ailleurs ce que nous nous sommes finalement résolus à faire en choisissant d'acquérir des champs lumineux surfaciques : nous avons ainsi fixé certains degrés de liberté dans l'information d'apparence en choisissant délibérément d'effectuer l'acquisition dans des conditions d'éclairage fixes.

Vouloir développer des systèmes complets tout en se plaçant dans le cas général, même si cela permet de lever de nombreuses limitations, entraîne malheureusement un accroissement drastique du niveau de complexité, autant pour ce qui est des technologies employées que des connaissances techniques humaines requises. Ainsi, du fait de la grande variété d'objets susceptibles d'être numérisés, il est intéressant de se poser la question de l'usage de méthodes d'acquisition et de rendu dédiées à certaines classes d'objets. En plus du protocole que nous avons proposé au cours des chapitres précédents, nous nous sommes également intéressés à l'élaboration d'une chaîne de traitements permettant, spécifiquement, la numérisation et la visualisation temps réel et réaliste de peintures d'art. Nous souhaitons offrir une méthode qui soit simple d'un point de vue technique mais qui préserve tout de même un certain degré de précision visuelle, afin de permettre la conception de galeries d'art numériques, ou même l'étude de tableaux via leurs copies virtuelles.

Si l'on considère l'état de l'art sur la numérisation de tableaux, on peut se rendre compte que la géométrie de la toile est généralement approchée par un simple plan, l'accent étant mis de préférence sur la précision de l'acquisition de l'information chromatique. Cependant, dans

la réalité, la surface d'une toile peinte peut présenter de nombreuses aspérités, dont certaines sont suffisamment significatives pour devoir être prises en compte lors de la visualisation. En fait, le relief joue un rôle à ce point important que les peintres eux-mêmes exploitent les effets visuels qu'il induit pour «accrocher» davantage la lumière sur la toile, comme dans le cas de la peinture au couteau. Ainsi, l'information géométrique doit être considérée avec autant de soin que l'information chromatique/photométrique.

Ainsi, nous partons du constat que la toile d'un tableau présente souvent des variations géométriques qui, même si elles sont généralement faibles, restent toutefois non négligeables au niveau visuel. Contrairement à la plupart des méthodes existantes, nous proposons donc d'acquérir la géométrie en plus de la texture couleur. En complément de cette phase d'acquisition, nous proposons également un algorithme pour la visualisation des copies numériques obtenues. Concernant la couleur, nous utilisons un modèle analytique estimé à partir des mesures pour représenter les variations induites sur la toile par les changements d'éclairage et de point de vue (information bidirectionnelle complète, donc). Pour la géométrie, une approche hybride combinant deux techniques de rendu de relief va être présentée. Elle se base sur un schéma adaptatif pour permettre un rendu très performant en choisissant celle des deux techniques qui semble la plus adaptée selon le niveau d'échelle sous lequel la toile est observée.

Ainsi, nos contributions dans ce domaine sont les suivantes :

- nous proposons une chaîne de traitements destinée à l'acquisition du relief et de la texture de toiles peintes. Cette chaîne comporte notamment un jeu d'outils spécifiquement conçus pour le traitement automatique des données géométriques acquises.
- nous proposons également un algorithme permettant de préserver le relief de la toile là où les autres méthodes l'ignorent. Cet algorithme considère l'information chromatique de cette toile sous la forme d'une texture bidirectionnelle, augmentant ainsi le réalisme. Un rendu très performant est alors obtenu grâce à un modèle hybride et un mécanisme adaptatif dédiés, et pour un coût de stockage faible.

Nous commencerons donc ce chapitre par un bref état de l'art, en section 1, concernant la numérisation de tableaux d'une part, puis les techniques de rendu pouvant présenter un intérêt pour la restitution d'une telle catégorie d'objets d'autre part. Nous venons effectivement de dire que le relief, même s'il doit être considéré avec attention, reste toutefois très faible pour ce genre d'objets. Or, il existe justement des techniques qui ont été spécifiquement développées pour le rendu de ce genre de reliefs. C'est pourquoi nous souhaitons avant tout les présenter. Nous verrons ensuite le principe général de notre chaîne de traitements (section 2), comportant les deux étapes distinctes de numérisation et de visualisation, respectivement décrites dans les sections 3 et 4. Nous proposerons une étude des résultats obtenus en section 5 avant de conclure ce chapitre en section 6.

1 Travaux antérieurs

Au sujet de la numérisation de tableaux, nous n'avons trouvé que peu de travaux s'intéressant à l'acquisition de peintures en vue de leur visualisation. En réalité, la grande majorité de ces travaux s'attachent principalement à l'étude des peintures d'art à des fins de conservation et de restauration.

Pour n'en citer que quelques uns, l'ouvrage publié par Christian Lahanier [94] présente des travaux menés dans le but de révéler certains des secrets de Léonard de Vinci concernant la réalisation de la Joconde. Une numérisation géométrique de la toile a été réalisée, mais surtout une analyse multispectrale de la peinture, permettant de faire apparaître, de manière non-invasive, la succession des différentes couches de peintures qui se superposent. Il est ainsi possible d'observer, à travers toutes ces couches, les esquisses originelles qui ont initialement servi de support à l'élaboration du tableau, de même que certaines rectifications apportées par Léonard lui-même dans le but de renforcer les impressions suscitées par son œuvre.

Dans le même ordre d'idées, Guidi *et al.* [64] ont numérisé l'«Adorazione dei Magi», autre peinture de Léonard De Vinci, à l'aide d'un scanner à lumière structurée. Le support de la peinture en question est un ensemble de 2,40m sur 2,40m de panneaux de bois attachés entre eux par deux grandes traverses situées à l'arrière du tableau. L'objectif du projet est alors d'utiliser la géométrie pour fournir un diagnostic sur l'état du support, sachant que l'assemblage des panneaux a été particulièrement détérioré, au fil du temps, par l'humidité et les nombreux déplacements. En revanche, la visualisation n'est pas du tout considérée : une information chromatique est belle et bien acquise, mais elle n'est utilisée que pour faciliter le recalage géométrique des différentes acquisitions nécessaires à la numérisation complète du support.

Il existe, bien évidemment, beaucoup d'autres travaux développés à des fins de restauration, et l'article de Guidi *et al.* [64] contient de nombreuses références sur le sujet. Cependant, à notre connaissance, les travaux concernant la numérisation dans un but de visualisation sont beaucoup plus rares.

Celui qui se rapproche le plus de ce que nous proposons est celui de Tominaga *et al.* [186], qui ont développé une technique permettant l'acquisition, à l'aide d'une caméra multispectrale, de la réflectance et de la normale en chaque point de la surface de la toile. L'objectif est clairement la restitution des copies numériques acquises. Les normales calculées permettent alors de simuler une information de relief lors des calculs d'illumination. Cependant, l'utilisation d'une simple texture de normales pour la restitution du relief ne permet pas de tenir compte des effets de parallaxe qui peuvent survenir lorsque le relief devient important, ou lorsque la toile est observée selon un angle rasant.

Ainsi, au vu de la littérature, on constate que la géométrie n'est généralement utilisée qu'à des fins de diagnostics et d'études, mais jamais pour la restitution sur écran.

Il existe cependant plusieurs algorithmes permettant la restitution temps réel de reliefs de faible amplitude, et qui pourraient donc être utilisés dans le cas qui nous intéresse. L'objectif de ces algorithmes est, bien entendu, d'éviter d'avoir à manipuler des maillages trop détaillés, ce qui est généralement peu attrayant lorsqu'il s'agit de faire du rendu hautes performances. Ainsi, des techniques comme le *bump mapping* [15] ont été introduites, utilisant une texture de normales pour simuler le relief en perturbant le calcul d'illumination. Le relief n'est donc pas réel, mais simulé, ce qui permet d'obtenir de très bonnes performances. En revanche, comme nous l'avons dit plus haut en présentant les travaux de Tominaga *et al.* [186], l'absence de gestion de la parallaxe engendre une altération de la qualité visuelle lorsque la surface est observée sous un angle rasant. Ainsi, si le relief devient quelque peu significatif, la méthode n'est plus du tout adaptée.

Des techniques comme le *displacement mapping* [32] ont alors été proposées, utilisant, pour pallier à ces problèmes, des méthodes d'application de textures visant à modifier directement la géométrie sous-jacente. Malheureusement, l'emploi de telles techniques signifie un retour à des modèles surfaciques extrêmement détaillés.

Plus récemment, le *relief mapping* [6, 136, 141, 143] et le *parallax mapping* [181] ont été introduits pour ajouter, de manière purement visuelle, des détails géométriques à l'aide de techniques avancées d'application de textures. Ainsi, un relief complexe et bien géré au niveau de la parallaxe peut être généré sur des objets 3D dont la complexité géométrique réelle reste très simple.

Outre la géométrie, la seconde information importante pour la visualisation réaliste de copies virtuelles de tableaux est, bien entendu, la réponse photométrique de la toile. Celle-ci doit également être acquise avec précision afin de pouvoir être restituée plus tard de manière fidèle. Comme nous l'avons expliqué dans le chapitre 5, les variations de couleur observables sur une surface sont liées à la réflectance du matériau, et sont donc dépendantes à la fois du point de vue et des conditions d'illumination. Dans le cas de tableaux, nous avons affaire à des objets dont la variation spatiale du matériau est très importante, du fait du mélange et de la superposition des différentes peintures utilisées. Nous avons discuté, toujours dans le chapitre 5, des différentes techniques qui pouvaient être utilisées dans un tel cas, aussi nous invitons le lecteur à revenir vers ce chapitre pour plus d'informations. Sans vouloir faire à nouveau le détail de ces méthodes, nous attirons tout de même l'attention sur l'approche de McAllister *et al.* [116] : les auteurs utilisent un modèle de Lafortune [93] pour représenter la réflectance en chaque point d'une surface. Nous avons en fait choisi ce même modèle pour représenter l'information bidirectionnelle de la toile des tableaux que nous numérisons. Nous rappellerons le principe de ce modèle un peu plus tard.

Le dernier point de ce bref état de l'art concerne les algorithmes hybrides de rendu. La plupart des techniques précédemment citées ne sont pas forcément adaptées à tous les cas de figure, du fait de leur coût en temps de calcul ou de leur qualité de rendu inadéquate. Les méthodes hybrides peuvent alors fournir une alternative intéressante. De tels mécanismes consistent à utiliser des algorithmes de rendu différents en fonction de certains paramètres. Becker *et al.* [7], par exemple, utilisent alternativement trois algorithmes différents, selon l'orientation et la distance du point de vue, pour restituer des objets dont la surface contient

de nombreux détails. Le principal problème posé par la mise en place de mécanismes hybrides se situe au niveau des transitions entre les différents modes de rendu. Ces transitions sont, en effet, souvent enclines à générer des artefacts visuels, qui proviennent majoritairement, comme le soulignent Heidrich *et al.* [72], des différences d'illumination existant entre un algorithme de rendu et un autre. Des mécanismes assurant des transitions douces entre les différents algorithmes doivent alors être introduits afin de limiter l'impact visuel de tels inconsistances [56].

Quoi qu'il en soit, si le problème des transitions est correctement géré, les techniques hybrides constituent des outils puissants pour accélérer le processus de rendu. Il n'existe cependant pas de solutions générales : chaque technique est adaptée à une situation spécifique. C'est pourquoi nous avons choisi d'introduire notre propre algorithme, particulièrement seyant au cas de figure qui nous intéresse.

2 Principe

La méthode que nous proposons se base sur le fait que la plupart des peintures d'art présentent des caractéristiques géométriques qui méritent d'être préservées. Une partie du relief de la toile est, en effet, suffisamment significative au niveau visuel pour mériter d'être prise en compte. Toutefois, la grande majorité du relief reste néanmoins négligeable et peut donc être approchée par une technique de rendu très simple. Nous proposons alors de découpler le relief significatif du reste de la toile.

La première étape de notre chaîne de traitements consiste à acquérir les données géométriques et photométriques de la toile. Cela signifie, notamment, gérer les problèmes de mise en correspondance entre la géométrie acquise et les différentes photographies utilisées pour en échantillonner l'apparence.

En second lieu, des traitements spécifiques sont proposés pour permettre :

- de reconstruire une texture bidirectionnelle représentant la réponse photométrique de la toile à partir des différentes photographies.
- de séparer, dans les données géométriques, le cadre du tableau de la toile en elle-même. Cela nous permet alors de ne travailler que sur l'information qui nous intéresse réellement.
- de générer un *champ de hauteurs* à partir des données de la toile, c'est-à-dire une carte 2D définissant, en chaque point, une élévation par rapport à un plan de référence qui, dans notre cas, représente le support du tableau. Cette représentation est, en effet, particulièrement adaptée à la forme généralement planaire de la catégorie d'objets considérée.
- de classer ce champ de hauteurs en deux parties distinctes : le relief *significatif* et le relief *négligeable*.

Lors de la phase de rendu, le relief négligeable est alors affiché à l'aide d'un algorithme peu coûteux de *bump mapping*, et le relief significatif, quant à lui, est restitué à l'aide de

l'algorithme plus complexe, mais également plus précis, du *relief mapping*. Ce dernier nous permet alors de gérer correctement les effets de masquage liés à la parallaxe pour les zones où le relief devient important. Les principes de ces deux algorithmes de rendu seront rappelés ultérieurement.

Dire si un relief est significatif ou non dépend totalement du niveau d'échelle sous lequel il est considéré. Nous proposons donc un mécanisme adaptatif permettant de n'utiliser l'algorithme le plus complexe que lorsque cela est réellement nécessaire. Ainsi, la partie du relief précédemment classifiée comme significative est répartie en un ensemble de petites boîtes 3D. Pour chacune d'elles, un calcul simple nous permet de déterminer, en fonction de la position de l'observateur, si cela présente un intérêt ou non d'utiliser l'algorithme de *relief mapping*. Ainsi, la totalité du relief significatif n'est pas systématiquement affichée : seule la partie réellement intéressante par rapport au point de vue courant est considérée. Nous verrons que ce mécanisme adaptatif, qui constitue la principale contribution de ce travail en termes de rendu, aboutit à une accélération drastique de la vitesse d'affichage.

L'apparence est finalement rendue grâce à la texture estimée précédemment, représentant l'information bidirectionnelle en chaque point de la toile à l'aide d'un modèle de Lafortune [93]. La couleur est reconstruite à la volée à partir de ce modèle grâce au matériel graphique.

3 Acquisition et prétraitements

Dans cette section, nous explicitons nos choix concernant l'acquisition de la géométrie et de la texture. Nous présentons également les prétraitements nécessaires à la génération des copies numériques qui seront utilisées ultérieurement par l'algorithme de rendu hybride (voir section 4).

■ 3.1 Configuration matérielle pour l'acquisition

Pour établir les copies numériques dont nous avons parlé, il nous faut capturer deux informations : la géométrie de la toile, ainsi que sa texture. Concernant l'acquisition de l'information géométrique, nous avons utilisé le même scanner à lumière structurée que précédemment. Nous l'avons placé de manière à ce qu'il regarde le tableau le plus de front possible. Compte tenu de la forme particulière de la classe d'objets que nous nous proposons de numériser, ce seul point de vue, quasi perpendiculaire au plan de la toile, est utilisé pour la numérisation 3D. Étant donné que la variation des détails picturaux d'un tableau peuvent présenter de forts contrastes, certaines zones captées par la caméra embarquée risquent d'être soit trop sombres, soit trop saturées, faisant alors échouer la mesure de la géométrie dans ces zones. Nous capturons donc plusieurs images télémétriques, prises depuis le même point de vue, mais avec différents temps d'exposition, afin de garantir la bonne acquisition des régions

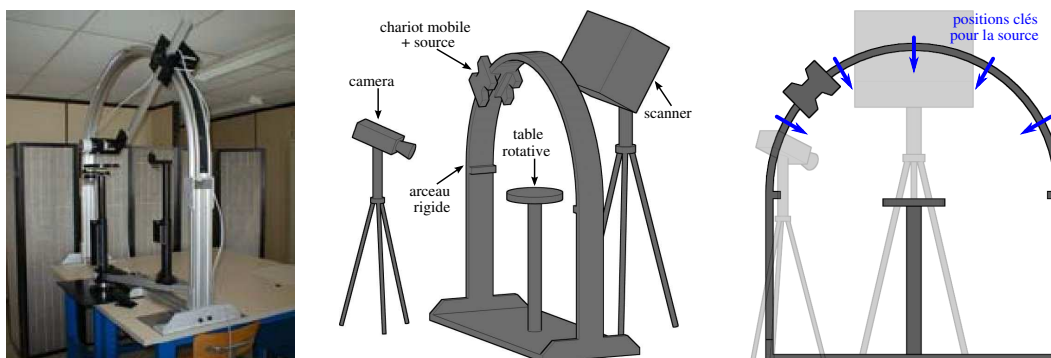


Figure 9.1 – Configuration matérielle utilisée pour la numérisation de tableaux.

claires et sombres à la fois. Les différentes images télémétriques obtenues sont alors fusionnées par une simple opération d'union sur l'information géométrique qu'elles contiennent.

Acquérir la seconde information, c'est-à-dire la texture, signifie collecter différents échantillons d'une fonction bidirectionnelle en chaque point de la toile. Chacun de ces échantillons doit correspondre à une valeur caractérisant la réponse du matériau pour un couple donné de directions d'incidence lumineuse et d'observation. Concernant les directions d'observation, il suffit de prendre une multitude de clichés, capturés depuis des points de vue différents. Concernant les directions d'incidence, nous devons utiliser une source la plus ponctuelle possible, afin de réduire au maximum l'angle solide et ainsi s'assurer que chaque échantillon correspond bien, dans une certaine mesure, à une unique direction d'illumination. Pour ce faire, nous nous sommes dotés d'une lampe à incandescence *Jocker-Bug 200* de la marque *K5600 lighting*, ayant la particularité de posséder un filament extrêmement fin.

Pour pouvoir déterminer les directions d'incidence et d'observation localement par rapport à la géométrie acquise, il nous faut pouvoir localiser les différents dispositifs entre eux. Étant donné les difficultés liées à la localisation d'un illuminant dans une scène¹, nous avons adopté une approche matérielle pour régler le problème de la localisation de la source : un arceau rigide a été utilisé, muni d'un chariot mobile sur lequel nous avons fixé la lampe. Le banc complet de numérisation mis en place pour ce faire est illustré sur la figure 9.1. Sous cet arceau se trouve une table rotative sur laquelle est posé le tableau à numériser. En déplaçant le chariot et en faisant tourner la table, il est ainsi possible de couvrir tout l'hémisphère d'incidence situé au-dessus de la toile. La localisation effective de la source par rapport au scanner se fait alors de la manière suivante :

1. une caméra numérique est temporairement placée sur le chariot mobile, à la place de la source.
2. cette caméra est localisée par rapport au scanner à l'aide de la paramétrisation par phase présentée au chapitre 6 pour un certain nombre de positions clés du chariot sur l'arceau (figure 9.1-droite).

¹Voir chapitre 5, section 2.2.

3. la lampe est alors replacée sur le chariot mobile et, lors de la mesure, seules les directions d'illumination associées à ces positions clés seront considérées.

Bien évidemment, cette technique de localisation demande à ce que le scanner reste statique par rapport à l'armature pendant la totalité de la mesure. De plus, cela présuppose que, une fois monté sur le chariot mobile, le filament de la lampe et le centre optique de la caméra sont confondus. Or, garantir qu'un tel prérequis est satisfait est impossible dans la pratique. Cependant, le fait que l'armature rigide ait été réalisée par un bureau d'étude nous permet d'affirmer que la différence entre ces deux positions n'excède pas quelques millimètres. Étant donné que la distance du chariot à la table rotative est d'environ un mètre, l'erreur commise dans la localisation de la source par rapport au repère objet du tableau peut être considérée comme négligeable.

Concernant la localisation des points de vue, la caméra est fixée sur un trépied et positionnée manuellement autour de l'objet. Les points de vue sont choisis de manière à couvrir de nombreuses directions d'observation, et chacun d'eux est localisé par rapport au scanner grâce à la paramétrisation par phase, une fois encore. Chaque point de vue acquis est immédiatement contrôlé, de manière à s'assurer en temps réel que les données récupérées sont correctes.

Bien entendu, lorsque la table rotative est tournée, le repère objet du tableau change par rapport au scanner et, par conséquent, le positionnement de la caméra et de la source ne sont plus valables dans ce repère. Pour déterminer le changement de repère objet, nous acquérons une image télémétrique pour chaque position de la table rotative. Le recalage géométrique de ces différentes images télémétriques, réalisé dans notre cas à l'aide de la paramétrisation par phase, nous permet alors de déterminer les matrices de changement de repères nécessaires au remplacement de toutes les données dans un même repère commun.

■ 3.2 Reconstruction de la texture bidirectionnelle

Grâce à la localisation des différents dispositifs, les photographies acquises peuvent être recalées sur la géométrie de la toile, permettant ainsi l'extraction de nombreux échantillons de texture dépendant à la fois du point de vue et de l'éclairage. Chacun de ces échantillons est composé de la valeur de luminance perçue par la caméra au point considéré, ainsi que des directions d'illumination et d'observation définies dans le repère local de ce point.

Du fait que la géométrie ne soit pas parfaitement plane, il se peut qu'un point de la toile ne soit pas visible depuis un point de vue donné, ou même qu'il ne soit pas éclairé pour une position de source donnée. Or, certains modèles analytiques de représentation de texture sont sensibles, en termes de précision, aux fortes discontinuités que peuvent représenter de tels effets de masquage. Comme l'information de relief est connue dans notre cas, nous choisissons d'éliminer les échantillons qui ne correspondent pas à de l'illumination ou à de l'observation directe à l'aide d'un algorithme de lancer de rayons. Il faut noter que cette étape n'est réellement nécessaire que si la toile présente vraiment des empâtements très prononcés.

Un modèle de Lafortune est finalement estimé en chaque point à partir des échantillons restant. Le modèle exact que nous utilisons est l'approximation proposée par McAllister *et al.* [116], donnée par l'équation 9.1 :

$$f_r(\vec{v}, \vec{l}) = \rho_d + \rho_s (C_x v_x l_x + C_y v_y l_y + C_z v_z l_z)^k \quad (9.1)$$

où ρ_d et ρ_s sont deux couleurs RVB décrivant respectivement les contributions diffuses et spéculaires à l'énergie lumineuse sortante, et \vec{v} et \vec{l} sont les directions locales d'observation et d'incidence lumineuse. La définition des autres quantités de cette équation peut être retrouvée dans la section 1.3 du chapitre 5. Ce modèle n'utilise donc qu'un seul lobe spéculaire, contrairement à d'autres modèles qui en définissent un par canal de couleur. Il faut noter que les données que nous estimons à l'aide de ce modèle ne représentent pas une BRDF à part entière : certes, nous estimons une information bidirectionnelle, mais l'information en question ne consiste pas uniquement en une fonction de réflectance puisqu'elle contient également des interrélaxions potentielles (il ne s'agit d'ailleurs pas non plus d'une BTF, en ce sens que la géométrie mésoscopique est représentée explicitement dans notre cas, et non pas encodée de manière purement basée images par la texture bidirectionnelle).

Nous avons choisi ce modèle pour deux raisons :

- sa simplicité, qui nous permet d'en effectuer l'évaluation directe par le matériel graphique et ainsi de reconstruire l'apparence de la toile à moindre coût,
- sa compacité, puisque le modèle peut être empaqueté dans seulement trois images de textures 2D : deux pour les coefficients trichromatiques ρ_d et ρ_s et une pour les paramètres (C_x, C_y, C_z, k) du lobe spéculaire.

Évidemment, d'autres modèles de représentation auraient pu être utilisés. Cependant, ces modèles ne présentent pas forcément une formulation aussi simple, ni aussi compacte, que celui-ci. De plus, le modèle de Lafortune permet de représenter une grande catégorie de matériaux, ce qui en fait un très bon candidat pour l'estimation à partir de données réelles.

■ 3.3 Extraction automatique de la toile

Nous allons maintenant voir comment générer, à partir de l'information géométrique acquise, le champ de hauteurs qui sera utilisé par notre algorithme de visualisation. Comme nous l'avons expliqué plus haut, la géométrie consiste en une unique image télémétrique, acquise selon un angle de vue quasi orthogonal au plan de la toile. Ainsi, cette image télémétrique constitue déjà une représentation très proche du champ de hauteurs que nous désirons obtenir. Nous allons donc maintenant traiter le contenu de cette image afin de générer le champ de hauteurs final.

Étant donné que notre méthode de rendu ne s'intéresse qu'à la toile, nous cherchons d'abord à isoler cette partie du reste du tableau. En d'autres termes, nous souhaitons éliminer de l'image télémétrique toute information géométrique relative au cadre du tableau ou à ce qui se trouve au-delà.

Du fait de sa forme hautement planaire, la géométrie du tableau peut être vue comme une élévation par rapport à un plan de référence. Nous allons donc déterminer ce plan de réf-

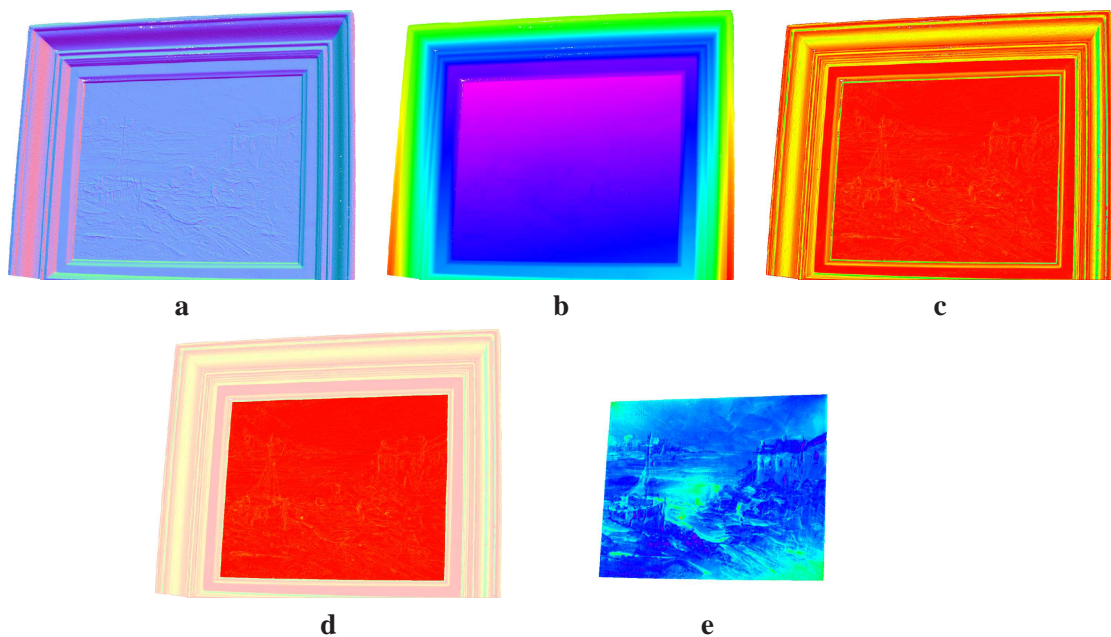


Figure 9.2 – Extraction automatique de la toile. a) Image télémétrique initiale, affichée ici en fausses couleurs. b) Calcul d'un premier champ de hauteurs par estimation du plan moyen des données géométriques. Le gradient de couleurs représente l'élévation par rapport à ce plan. c) Extraction de contours appliquée au champ de hauteurs. d) Utilisation de ces contours pour détecter l'interface entre la toile et le cadre du tableau. e) Calcul du champ de hauteurs final à partir des seules données de la toile.

rence en prenant, dans un premier temps, l'ensemble des données contenues dans l'image télémétrique. Pour ce faire, nous effectuons une *analyse en composantes principales (ACP)* des données géométriques afin d'extraire un nouveau repère local $(\vec{X}, \vec{Y}, \vec{Z})$ aligné sur le plan moyen passant par ces données. \vec{X} et \vec{Y} correspondent alors aux deux vecteurs tangents au plan du tableau (les deux premières composantes principales, représentant les axes de plus grande dispersion des données), et \vec{Z} représente le vecteur orthogonal au plan (axe de moindre dispersion). Ce troisième vecteur est bien entendu celui qui va nous permettre de construire le champ de hauteurs.

Les points de l'image télémétrique sont alors exprimés par rapport à ce nouveau repère, et la coordonnée associée à l'axe \vec{Z} est appelée *coordonnée de hauteur* (figure 9.2-b). À la suite de ce changement de repère, l'interface entre la toile et le cadre est clairement caractérisée par un saut important dans la coordonnée de hauteur comparé aux petits détails géométriques de la toile. Ainsi, nous commençons par détecter ces sauts grâce à l'application d'un opérateur de Sobel sur les coordonnées de hauteur, afin de mettre en valeur ces fortes discontinuités (figure 9.2-c). L'extraction de la toile se fait alors par un algorithme de croissance de régions qui s'arrête lorsqu'une zone de gradient trop important est rencontrée (figure 9.2-d). Nous choisissons comme point de départ de cet algorithme le pixel central de l'image télémétrique, dans la mesure où il semble logique qu'un tableau correctement mesuré se trouve bien centré par rapport au champ de vision du dispositif de numérisation.

Ainsi, toute l'information géométrique non pertinente pour notre cas est éliminée en ne conservant que les pixels de l'image télémétrique qui ont été parcourus par l'algorithme de croissance de régions. Une érosion de quelques pixels est ensuite appliquée afin d'éviter la présence d'éventuels artefacts résiduels. Une fois que seule l'information pertinente demeure, une nouvelle ACP est appliquée aux données de manière à calculer un nouveau repère local $(\vec{X}', \vec{Y}', \vec{Z}')$ qui soit plus représentatif des seules données de la toile. Une fois exprimées dans ce nouveau repère, les coordonnées des points de l'image télémétrique le long de l'axe \vec{Z}' représentent l'élévation par rapport au plan de la toile, et donc notre champ de hauteurs final (figure 9.2-e).

■ 3.4 Classification du relief

Notre but est maintenant de séparer du reste de la toile la partie du relief qui est jugée suffisamment significative. Nous définissons pour cela un seuil δ_0 en-dessous duquel la hauteur est considérée comme négligeable. Actuellement, ce seuil est fourni par l'utilisateur. Durant nos tests, nous lui avons donné une valeur située entre 20% et 40% de la hauteur maximale de la toile, mais ce choix reste très dépendant des données.

Tous les pixels de l'image télémétrique dont la hauteur est supérieure à δ_0 sont marqués *valides*, et les autres *non valides*, de manière à construire le masque binaire \mathcal{M} de la figure 9.3-b. Ce masque est alors nettoyé en supprimant les composantes connexes valides qui sont trop petites, de manière à éliminer la possible influence du bruit de numérisation. Cela est accompli en fixant un seuil minimum, en nombre de pixels, sur l'aire tolérée pour ces composantes. Les pixels valides restants correspondent alors aux pixels de l'image télémétrique qui représentent le relief significatif de la toile (figure 9.3-c).

La grille de pixels de l'image télémétrique est ensuite subdivisée en un ensemble de cellules de $n \times n$ pixels. Les cellules qui ne contiennent aucun pixel valide dans \mathcal{M} sont supprimées. Pour chacune de celles restantes, une boîte 3D est créée, dont la longueur et la largeur sont égales aux dimensions de la cellule, et dont la hauteur correspond à l'altitude h_{max} du pixel p_{max} le plus haut parmi tous ceux qu'elle contient. À cette boîte sont également associées les coordonnées (u_{max}, v_{max}) dans l'espace de l'image télémétrique du pixel p_{max} . Ces informations nous seront utiles pour la mise en place de notre mécanisme adaptatif, décrit en section 4.4.

L'ensemble de boîtes ainsi créé englobe donc totalement le relief segmenté, comme l'illustre la figure 9.3-d. Cet ensemble sera utilisé pour afficher séparément la partie significative du relief. Il est évident, mais important, de noter que plus n est petit, plus le nombre de boîtes créées est grand. Ainsi, n nous permet de contrôler la granularité du mécanisme adaptatif, comme nous le verrons plus tard. Le choix de sa valeur sera d'ailleurs discuté dans la section résultats (section 5).

À l'issue de tous ces traitements, nous obtenons donc les données suivantes :

- un champ de hauteurs, construit directement à partir de l'image télémétrique acquise et ne contenant que les données géométriques relatives à la toile du tableau,

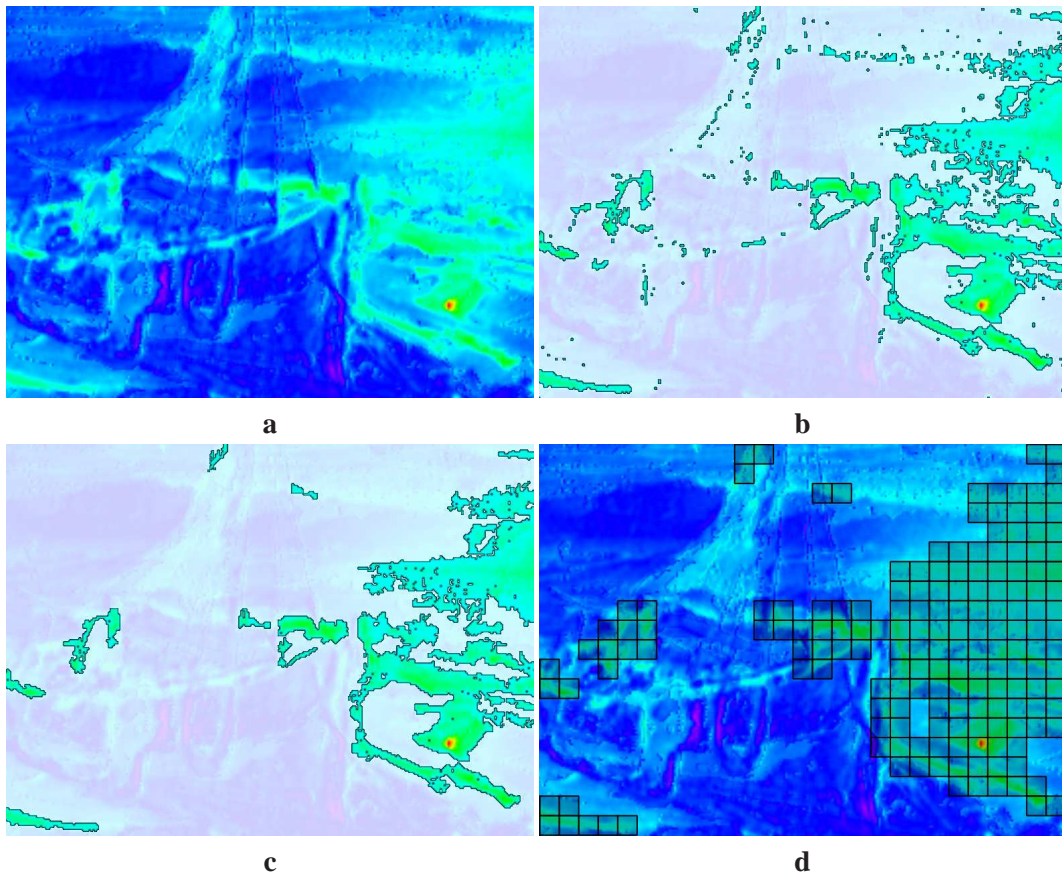


Figure 9.3 – Classification du relief. a) Champ de hauteurs construit à partir de l’image télémétrique acquise, mise en valeur par un gradient de couleur. b) Seuillage des coordonnées de hauteurs. c) Nettoyage pour réduire l’influence du bruit de numérisation. d) Boîtes créées autour des composantes connexes restantes et utilisées plus tard pour le rendu du relief significatif.

- un seuil δ_0 définissant, dans ce champ de hauteurs, l’altitude au-delà de laquelle le relief est considéré comme significatif,
- un ensemble de boîtes 3D délimitant la zone contenant le relief significatif, et qui servira de support pour en effectuer l’affichage,
- la hauteur h_{max} du pixel le plus haut de chaque boîte, ainsi que les coordonnées (u_{max}, v_{max}) de ce pixel dans l’espace image du champ de hauteurs,
- trois images contenant les coefficients de texture bidirectionnelle de la toile :
 - deux images RVB stockant les composantes trichromatiques des couleurs diffuses et spéculaires, respectivement,
 - une image RVBA stockant les quatre coefficients C_x, C_y, C_z et k définissant la forme du lobe spéculaire de Lafortune.

L’ensemble de ces données constituent alors la copie numérique complète du tableau.

4 Rendu

À partir des données obtenues précédemment, nous avons développé un algorithme de rendu permettant d'afficher, avec de bonnes performances et une bonne qualité visuelle, les copies numériques ainsi constituées. Cet algorithme travaille en deux passes de rendu :

- une première passe pour le relief négligeable, qui utilise une technique de rendu très basique mais très performante, le *bump mapping*, simulant le relief sans gérer la parallaxe (le relief, dans ce cas, est trop faible pour que l'on ait besoin de s'en soucier),
- une passe pour le relief significatif, qui utilise l'ensemble de boîtes préalablement déterminé comme support pour l'algorithme plus complexe du *relief mapping*. Ce dernier corrige alors les erreurs de parallaxe introduites lors de la passe précédente là où cela est nécessaire.

Ces deux passes de rendu, ainsi que des rappels sur les algorithmes qu'elles utilisent, sont respectivement décrites dans les sections 4.1 et 4.2.

Nous parlons ensuite, dans la section 4.4, du mécanisme adaptatif que nous avons ajouté à ce rendu hybride. L'affichage du relief significatif étant plus coûteux, ce mécanisme nous permet d'en réduire l'usage au maximum en tenant compte du point de vue pour déterminer dans quelle mesure la parallaxe nécessite d'être corrigée.

■ 4.1 Rendu du relief négligeable

Le *bump mapping* consiste à appliquer sur un modèle 3D une image contenant non pas de la couleur, mais un ensemble de vecteurs 3D. Cette image de vecteurs permet alors de décrire la variation de la normale sur la surface de manière beaucoup plus fine que ne le permet la résolution du maillage 3D en lui-même. Cette texture de normales est alors utilisée pendant le rendu pour perturber la vraie normale géométrique lors des calculs d'illumination en chaque pixel, ce qui a pour effet de donner une impression de relief à la surface sans pour autant recourir à un maillage 3D détaillé.

Cette technique de plaquage de texture n'implique que des calculs très simples, et permet donc un rendu très performant. Cependant, elle ne sied qu'à la restitution de reliefs de faible amplitude : étant donné que la géométrie sous-jacente n'est pas réellement modifiée, tous les effets de parallaxe normalement visibles pour des angles de vue rasant sont totalement absents. Ces raisons expliquent pourquoi nous avons choisi d'utiliser cet algorithme pour la restitution du relief négligeable.

Nous commençons donc par une première passe de rendu, durant laquelle l'intégralité de la toile est affichée à l'aide d'un simple plan texturé grâce à l'algorithme du *bump mapping*. La texture de normales utilisée pour ce faire est construite en appliquant un calcul de gradient à notre champ de hauteurs. Étant donné que la géométrie du tableau est statique, ce calcul n'est évidemment effectué qu'une seule fois, lors d'une phase d'initialisation.

Plutôt que d'utiliser plusieurs images de texture, chacune contenant une information différente, nous empaquetons la texture de normale et le champ de hauteurs au sein d'une unique image : pour chaque triplet RVB, R et V contiennent alors les deux premières composantes n_x et n_y du vecteur normal, et B la hauteur. La troisième composante n_z du vecteur normal est, quant à elle, systématiquement recalculée à la volée lors du rendu.

■ 4.2 Rendu du relief significatif

Le rendu du relief significatif s'effectue au cours d'une seconde passe de rendu, durant laquelle l'ensemble de boîtes 3D déterminées précédemment est utilisé comme support pour l'application d'un algorithme de *relief mapping*. Cette passe de rendu se superpose alors à la précédente, de manière à corriger les erreurs de parallaxe introduites par le *bump mapping* lorsque le relief devient trop important par rapport au point de vue.

L'algorithme du *relief mapping*, initialement décrit par Policarpo *et al.* [143], est en fait une technique avancée de plaquage de textures, à l'instar du *bump mapping*. L'idée est d'ajouter du relief à un modèle géométrique très grossier en lui appliquant une image de texture représentant ce relief sous la forme d'un champ de hauteurs, c'est-à-dire sous la forme d'une carte décrivant la profondeur à laquelle se trouve la surface réelle par rapport à la surface du modèle 3D.

Pour obtenir une bonne gestion de la parallaxe, un véritable test d'intersection entre le rayon d'observation et le champ de hauteurs est alors exécuté à la volée par le processeur graphique pour chaque pixel écran couvert par l'objet affiché (voir la figure 9.4-gauche). Pour ce faire, le point d'entrée du rayon d'observation dans l'objet est déterminé, ainsi que ses coordonnées de texture. Ces coordonnées de texture nous donnent alors le pixel du champ de hauteurs à la verticale duquel se trouve le point d'entrée. Connaissant la profondeur maximale du champ de hauteurs, les coordonnées de texture du point de sortie du rayon d'observation peuvent également être déterminées. La recherche d'intersection entre le rayon d'observation et le champ de hauteurs se fait alors en deux étapes :

1. **recherche linéaire** : la recherche s'effectue en avançant, en espace texture, selon un pas constant entre le point d'entrée et le point de sortie. Si, à la suite de l'un de ces pas, nous aboutissons à un point situé sous le champ de hauteurs, cela signifie que l'intersection se trouve entre le point actuel et le point atteint au pas précédent. La recherche linéaire s'arrête alors à ce moment.
2. **recherche binaire** : la recherche précédente nous donne donc un intervalle à l'intérieur duquel se trouve la véritable intersection. Les bornes de cet intervalle sont alors raffinées à l'aide d'une recherche dichotomique.

Les données texture (normale, couleur, etc.) finalement utilisées pour la suite du rendu sont alors celles situées aux nouvelles coordonnées de texture déterminées par cette recherche d'intersections.

Une fois encore, on peut dire que le relief est simulé, en ce sens qu'il n'existe pas réellement sur le modèle 3D. En revanche, et contrairement au *bump mapping*, ce relief est visuellement

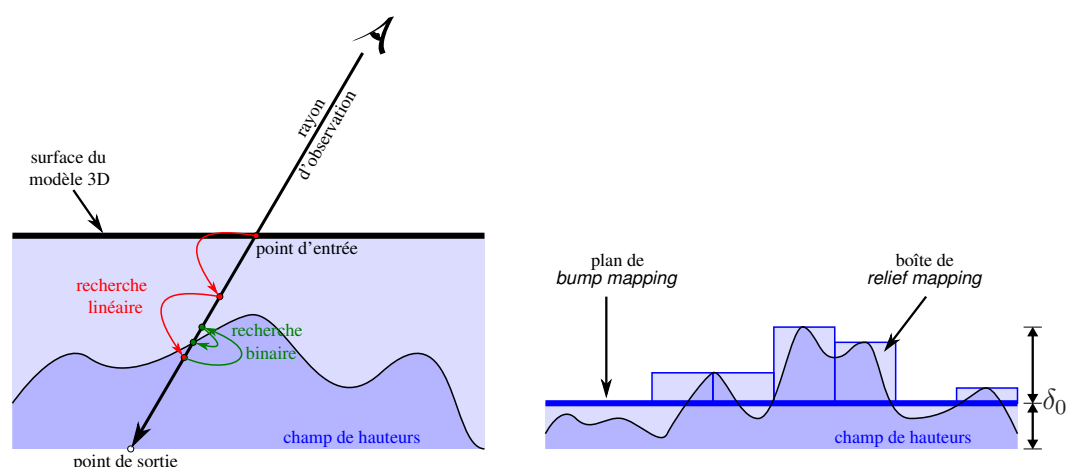


Figure 9.4 — **Gauche** : algorithme du *relief mapping*. La vraie surface est représentée par un champ de hauteurs située sous la surface de l'objet 3D. Une recherche d'intersections entre le rayon d'observation et ce champ de hauteurs effectuée sur le processeur graphique au moment du rendu. **Droite** : notre algorithme hybride. Positions relatives du plan de *bump mapping* et des boîtes englobant le relief significatif par rapport à l'altitude réelle du plan de la toile.

cohérent, étant donné que les effets de masquage liés à la parallaxe sont entièrement pris en compte par la recherche d'intersections entre le rayon d'observation et le champ de hauteurs. De plus, il est possible de stocker dans le tampon de profondeurs des valeurs qui soient cohérentes par rapport à ce relief, exactement comme si celui-ci existait réellement.

Dans notre cas, nous utilisons les boîtes 3D comme support de rendu pour cet algorithme. Chaque boîte est associée à une portion de la texture contenant le champ de hauteurs et les normales. Cette texture est, par conséquent, partagée entre toutes les boîtes. Étant donné que les boîtes sont positionnées de manière à ne représenter que la partie significative du relief de la toile, leurs bases se situent à l'altitude δ_0 . Ainsi, pour éviter qu'un écart ne subsiste entre les boîtes et le plan de *bump mapping*, ce dernier est déplacé jusqu'au seuil δ_0 . Ainsi, le niveau zéro de la vraie toile se trouve en réalité sous le plan de *bump mapping*, comme l'illustre la figure 9.4-droite.

■ 4.3 Calcul d'illumination

Comme nous l'avons décrit dans la section 3.2, nous utilisons un modèle de Lafortune pour calculer l'illumination en chaque point. L'équation 9.1 est alors directement évaluée par le processeur graphique.

Dans le cas du *relief mapping*, les coordonnées de texture utilisées pour récupérer les coefficients du modèle de Lafortune correspondent, évidemment, à celles obtenues par le calcul d'intersection entre le rayon d'observation et le champ de hauteur. Cette intersection est d'ailleurs également utilisée pour déterminer la direction locale d'illumination qui servira à

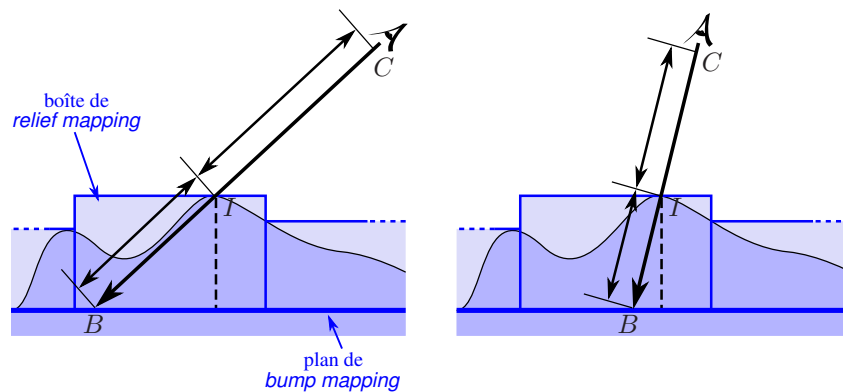


Figure 9.5 – Calcul du coefficient de transition α_{boite} à partir du point I le plus haut de chaque boîte. Le rapport des distances $\|\vec{IB}\|/\|\vec{CI}\|$ dépend à la fois de l'angle de vue et de la distance de l'observateur à la surface.

évaluer le modèle de Lafortune.

Concernant le *bump mapping*, les directions d'illumination et d'observation sont toutes deux corrigées à l'aide de la coordonnée de hauteur, de manière à produire une illumination qui tienne compte de la vraie position de la surface, et ainsi garantir une cohérence entre les illuminations calculées pour chacun des deux algorithmes. Cela a en effet son importance pour la suite, comme nous allons le voir.

■ 4.4 Mécanisme adaptatif dépendant du point de vue

Si l'on observe une peinture sous un angle perpendiculaire au plan de sa toile, on remarque que la différence visuelle entre les deux algorithmes de rendu est quasiment inexistante. Il en va de même lorsque le point de vue est très éloigné du tableau, c'est-à-dire lorsque les variations de hauteurs deviennent négligeables par rapport à leurs tailles projetées à l'écran. Dans ces deux cas de figure, les différences introduites par la parallaxe entre la surface réelle et le plan de *bump mapping* deviennent alors trop petites pour être encore perceptibles. Dans de telles circonstances, l'utilisation de l'algorithme de *relief mapping* pour corriger la parallaxe devient alors obsolète.

Nous avons donc développé un algorithme permettant de déterminer si chacune de nos boîtes doit être affichée ou non, en fonction du point de vue courant. Afin d'éviter d'éventuels effets de *popping*² lors des transitions entre les états *visible* et *non visible* d'une boîte, cet algorithme propose également d'utiliser une composition alpha pour que ces transitions soient progressives.

L'état d'une boîte est contrôlé par un coefficient α_{boite} , calculé en considérant le point le plus haut de la boîte. Soit I les coordonnées de ce point dans la scène, aisément calculées à partir des valeurs u_{max} , v_{max} et h_{max} déterminées en section 3.4 et associées à chaque

²Apparitions ou disparitions brutales d'éléments de la scène.

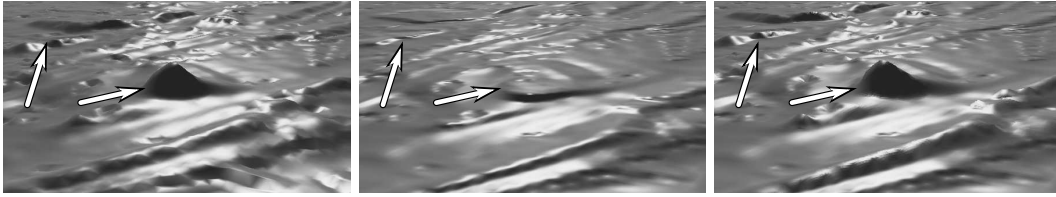


Figure 9.6 — Comparaison entre différents modes de rendu. **Gauche** : rendu basé maillage. **Milieu** : rendu utilisant exclusivement du *bump mapping*. **Droite** : notre rendu hybride, mélangeant de *bump mapping* et *relief mapping* pour tenir compte de la parallaxe lorsque le relief devient significatif.

boîte. Soient C la position actuelle du point de vue et B l'intersection avec le plan de *bump mapping* du rayon d'observation passant par I (voir le schéma de la figure 9.5). Considérons maintenant le rapport r des distances situées de part et d'autre du point I , c'est-à-dire le ratio obtenu par le calcul suivant :

$$r = \frac{\|\vec{IB}\|}{\|\vec{CI}\|} \quad (9.2)$$

Comme on peut le voir sur la figure 9.5, ce ratio dépend directement de l'angle sous lequel la surface est observée, mais également de la distance qui la sépare du point de vue : pour les points de vue éloignés ou frontaux (c'est-à-dire ceux pour lesquels l'erreur de parallaxe devient négligeable), la valeur de r tend vers zéro. À l'inverse, celle-ci augmente lorsque l'observateur se rapproche de la surface, ou que l'angle de vue devient rasant, c'est-à-dire pour les situations où l'effet de parallaxe devient important. Ce ratio est donc pertinent pour notre mécanisme adaptatif.

Le coefficient α_{boite} est alors calculé pour chaque boîte grâce à l'équation 9.3 :

$$\alpha_{boite} = \begin{cases} 0 & \text{si } \kappa r < \epsilon_{min} \\ 1 & \text{si } \kappa r > \epsilon_{max} \\ \frac{\kappa r - \epsilon_{min}}{\epsilon_{max} - \epsilon_{min}} & \text{sinon} \end{cases} \quad (9.3)$$

où ϵ_{min} et ϵ_{max} sont deux valeurs seuil telles que $0 < \epsilon_{min} \ll \epsilon_{max} \leq 1$, et κ est un facteur défini par l'utilisateur pour contrôler la sensibilité de la transition entre les deux algorithmes de rendu. Le seuil ϵ_{max} sert à mettre une limite supérieure au coefficient α_{boite} , de manière à ce que celui-ci reste toujours dans l'intervalle $[0, 1]$. Concernant le seuil ϵ_{min} , étant donné que le point I ne peut jamais être incident au plan de *bump mapping* (sans quoi la boîte en question n'aurait jamais été créée), la valeur de r ne sera jamais nulle. Ce seuil nous permet donc de forcer α_{boite} à prendre une valeur nulle lorsque le ratio r devient vraiment très petit. Dans notre implantation, nous avons fixé $\epsilon_{min} = 0, 1$ et $\epsilon_{max} = 1$.

Ainsi, selon la valeur de α_{boite} , trois cas de figure sont à considérer :

- si α_{boite} est nul, la boîte n'est pas affichée du tout.
- si $\alpha_{boite} = 1$, la boîte est affichée sans aucune composition alpha.
- si $\alpha_{boite} \in]0, 1[$, la boîte est composée avec le plan de *bump mapping* sous-jacent en utilisant le coefficient α_{boite} comme valeur de transparence.

Ces trois cas de figure permettent donc de n'afficher les boîtes contenant le relief significatif que lorsque cela est nécessaire, tout en assurant des transitions visuelles douces lors des phases d'apparition et de disparition des boîtes.

Comme cela a déjà été évoqué dans ce chapitre, des inconsistances lumineuses peuvent parfois apparaître lorsque différents algorithmes de rendu sont combinés. Dans notre cas, la même normale, la même information de couleur et le même modèle d'illumination sont utilisés pour les deux algorithmes. De plus, des corrections de hauteurs sont appliquées aux directions locales d'incidence lumineuse et d'observation, comme nous l'avons expliqué dans la section 4.3. Ainsi, aucune différence d'illumination n'est visible au niveau des transitions, comme on peut le constater grâce à la figure 9.6.

5 Résultats expérimentaux

Nous avons testé notre méthode sur la peinture que l'on peut voir sur la figure 9.7. Cette figure compare deux méthodes de rendu à des photographies réelles : notre rendu hybride utilisant une texture bidirectionnelle et le rendu obtenu à l'aide d'une simple texture diffuse. On voit clairement que l'apport de la texture bidirectionnelle sur le réalisme est manifeste. Dans cet exemple, la texture a été reconstruite à partir de 340 photographies capturées pour différentes conditions d'éclairage et d'observation.

Pour ce qui est du mécanisme adaptatif, la figure 9.8 illustre la prise en compte du point de vue lors du rendu pour automatiquement sélectionner les boîtes représentant les zones pour lesquelles l'erreur de parallaxe introduite par le *bump mapping* doit être corrigée. Le résultat de la composition des deux algorithmes de rendu a déjà été montré lors de la section précédente, sur la figure 9.6. On peut notamment voir que les erreurs de parallaxe sont effectivement corrigées, et qu'aucun artefact n'est visible au niveau de la jonction des deux modes de rendu. Nous allons donc maintenant discuter de l'intérêt de notre rendu hybride adaptatif en termes de performances.

■ 5.1 Étude des performances de l'algorithme de rendu

Nous avons testé notre méthode pour différentes conditions d'observation, c'est-à-dire pour des points de vue plus ou moins proches du tableau, et des angles de vue plus ou moins rasants. Quatre modes de rendu ont alors été comparés :

- un rendu purement basé maillages, se plaçant dans le cas où la géométrie du tableau est intégralement conservée et affichée,
- un rendu intégralement effectué par *relief mapping*, ce qui revient à ne considérer qu'une seule grande boîte couvrant la totalité de la toile,
- notre rendu hybride mais sans activation de l'adaptativité (le plan de *bump mapping* et



Figure 9.7 — Gauche : photographies du tableau test. Milieu : rendu par plaquage de texture classique sur un simple plan. Droite : rendu à l'aide d'une texture bidirectionnelle et de notre méthode hybride.



Figure 9.8 — Rendu adaptatif. Pour les points de vue proches (gauche) ou les angles d'observation rasants (droite), un nombre plus important de boîtes est automatiquement sélectionné afin de corriger l'erreur de parallaxe introduite par le *bump mapping*.

Distance d'observation	Angle de vue	Maillage	Relief mapping	Hybride sans adaptatif	Hybride avec Adaptatif
proche	rasant	71	36	33-48	36-57
	frontal	85	32	39-48	56-72
cadrée	rasant	60	100	61	330
	frontal	52	74	60	350
éloignée	rasant	96	280-450	82	650
	frontal	96	155-230	80	1250

Tableau 9.1 — Vitesse d'affichage (Hz) obtenue pour différents algorithmes de rendu et pour différentes conditions d'observation.

- toutes les boîtes sont systématiquement affichés),
- notre rendu hybride avec adaptativité au point de vue.

Les performances enregistrées pour ces différents cas de figure sont reportées dans le tableau 9.1. Ces mesures ont été effectuées sur un ordinateur de bureau doté d'un processeur AMD Athlon X2 4200+ ainsi que d'une carte graphique NVIDIA GeForce 7900GTX.

Comme on peut le voir, les performances obtenues par la méthode hybride adaptative permettent une visualisation temps réel. Ce constat plutôt positif est cependant à nuancer si on le compare à d'autres modes de rendu. On peut d'abord voir que le rendu hybride sans le mécanisme adaptatif ne présente aucun intérêt, étant donné que les vitesses de rendu, dans ce cas, sont systématiquement les moins bonnes. Elles sont d'ailleurs également moins bonnes que le rendu utilisant du *relief mapping* pour l'intégralité de la toile, ce qui est assez surprenant dans la mesure où le rendu hybride, même sans adaptativité, est censé limiter l'usage de cet algorithme plus coûteux à un nombre plus restreint de pixels écran.

Nous avons plusieurs idées sur les causes de cet état de faits :

- tout d'abord, force est de constater qu'il existe des pixels sur lesquels plusieurs boîtes 3D se projettent au moment du rendu. Ainsi, pour les pixels incriminés, l'algorithme de *relief mapping* est calculé à plusieurs reprises.
- ensuite, il ne faut pas oublier que nous effectuons deux passes de rendu : une pour chacun des deux algorithmes utilisés. Or, la seconde passe étant superposée à la première, les pixels atteints par les deux passes à la fois engendrent forcément un surcoût de calculs. Et ceci est d'autant plus vrai lorsque l'on considère le calcul d'illumination : l'évaluation du modèle de Lafortune n'est pas une opération bénigne, même pour un processeur graphique. Effectuer cette opération deux fois par pixel (voir plus, si plusieurs boîtes s'y projettent) engendre donc forcément une baisse de performance.
- enfin, notre méthode demande d'effectuer une composition alpha, ce qui est forcément un peu plus coûteux que si cette composition n'a pas à être faite. L'impact de ce dernier point est cependant discutable.

En revenant, cette fois, au cas adaptatif, on constate également que notre approche demeure moins efficace qu'un rendu basé maillages lorsque le point de vue est très proche de la toile. Cela s'explique par le fait que le rendu utilise alors plus de boîtes dans le but de corriger les erreurs de parallaxe. On constate évidemment le même phénomène lorsque l'angle de vue devient rasant. Toutefois, la distance de l'observateur à la toile a un impact plus fort sur les performances dans la mesure où, en plus d'utiliser plus de boîtes, la proportion de pixels écran à traiter par *relief mapping* est très importante.

Nous attirons l'attention sur le fait que nous n'ayons volontairement pas comparé notre méthode à un rendu purement basé sur du *bump mapping*. Il y a deux raisons à cela :

- au niveau des performances, il est inutile d'effectuer des mesures pour savoir que le *bump mapping* surclasse forcément tous les autres algorithmes, étant donné qu'il est le plus léger de tous à calculer.
- nous n'avons voulu comparer entre elles que les méthodes permettant d'obtenir le même degré de qualité visuelle. Or, le *bump mapping* ne peut clairement pas prétendre à faire aussi bien que les méthodes que nous avons considérées, du fait de l'absence de gestion

Taille des boîtes	Nombre de boîtes	Angle de vue		
		Frontal	Intermédiaire	Rasant
5 × 5	2073	305	142	18
10 × 10	714	300	185	38
20 × 20	253	330	250	75
50 × 50	81	240	175	145
100 × 100	35	135	130	125

Tableau 9.2 – Vitesse d'affichage (Hz) de notre méthode hybride adaptative pour différentes tailles de boîte et différents angles d'observation.

de la parallaxe.

La comparaison avec cet algorithme n'est donc pas pertinente.

■ 5.2 Étude de l'influence de la taille des boîtes

Le niveau de granularité de l'algorithme adaptatif est dicté par la taille donnée lors des prétraitements pour la construction des boîtes. Plus cette granularité est fine (taille plus petite), meilleure est l'approximation du relief significatif par l'ensemble de boîtes, mais plus coûteux est le calcul du coefficient α_{boite} , qui doit alors être déterminé pour un plus grand nombre de boîtes. Compte tenu de ces deux facteurs contradictoires, nous avons voulu vérifier les performances de rendu pour différentes tailles de boîtes. Les mesures obtenues sont reportées dans le tableau 9.2.

Regardons d'abord ce qui se passe pour les angles de vue frontaux. Une nette baisse des performances peut être observée pour les tailles de boîtes les plus importantes (50×50 et 100×100). Cela est justement dû au fait que l'ensemble de boîtes approche le relief significatif de manière beaucoup trop grossière, comme nous venons de l'évoquer. Ainsi, la proportion de ce relief qui est affichée est largement supérieure à ce qu'elle devrait être.

Si l'on considère maintenant le cas des angles de vue rasants, on observe une chute drastique des performances, mais cette fois-ci pour les boîtes de petites tailles (5×5 et 10×10). Cette chute ne semble en revanche pas imputable au calcul du coefficient α_{boite} , contrairement à ce que nous avons suggéré plus haut : ce coefficient doit en effet être calculé dans tous les cas, puisque c'est sa valeur qui permet de définir l'état des boîtes. Or, en regardant les performances pour les autres angles de vue, l'utilisation de petites boîtes reste tout à fait compétitif par rapport aux tailles de boîtes supérieures. Ce calcul ne peut donc être mis en cause.

Ces résultats tendent à corroborer notre supposition précédente quant à l'influence néfaste des calculs redondants pour les pixels dans lesquels se projettent plusieurs boîtes, phénomène devenant forcément plus important lorsque le nombre de boîtes augmente.

Au vu du tableau 9.2, les tailles de boîte intermédiaires semblent donc proposer le meilleur compromis entre les différentes conditions d'observation. N'ayant basé ce constat que sur

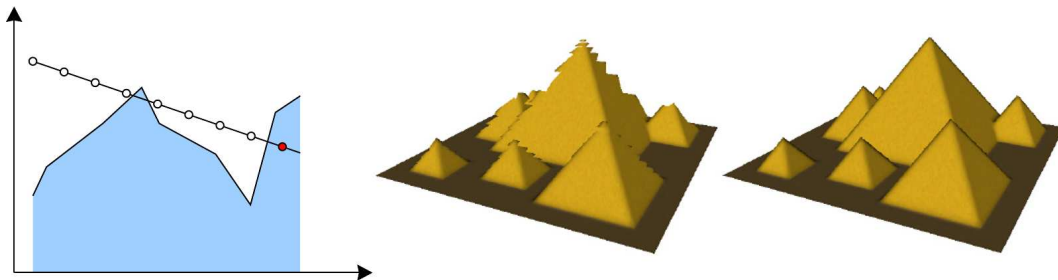


Figure 9.9 – Cause des artefacts visuels pouvant survenir avec le *relief mapping* (images tirées de [6]). **Gauche** : cas où l'étape de recherche linéaire manque une intersection. **Milieu** : artefacts visuels engendrés. **Droite** : version corrigée par la méthode de Baboud *et al.* [6].

un seul jeu de données, il nous est cependant difficile d'affirmer que le comportement de la méthode face à des tailles de boîtes très petites ou très grandes reste systématiquement le même. Nous ne savons d'ailleurs pas, à l'heure actuelle, comment déterminer la taille optimale de manière automatique.

■ 5.3 Discussion sur l'utilisation de champs de hauteurs

L'algorithme du *relief mapping* semble particulièrement adapté au rendu de tableaux : les peintures traditionnelles présentent, en effet, une forme fortement plane pouvant être représentées très correctement par un champ de hauteurs. Cependant, il est important de remarquer que le rendu de champs de hauteurs texturés impose certaines limitations. En effet, si un saut important se produit entre les hauteurs de deux pixels adjacents, la couleur le long de la falaise générée est simplement interpolée entre ces deux points. Ainsi, plus les discontinuités dans le relief sont importantes, plus la texture est étirée. Cette limitation rend difficile l'extension de notre méthode au rendu de fresques ou de bas-reliefs présentant une amplitude géométrique importante.

Un autre problème du *relief mapping* est l'apparition d'artefacts visuels dus à l'avancée pas à pas lors de la recherche d'intersections entre le rayon d'observation et le champ de hauteurs. La figure 9.9-gauche illustre l'exemple d'un cas problématique pour un tel algorithme : l'étape de recherche linéaire avance en espace texture selon un pas régulier jusqu'à se retrouver sous le champ de hauteurs. Cependant, si une partie de la géométrie représentée par ce champ est plus étroite que le pas utilisé, l'intersection peut être manquée. Cela aboutit à des artefacts du genre de ceux que l'on peut voir sur la figure 9.9-milieu, survenant principalement au niveau des arêtes, c'est-à-dire là où la géométrie est la moins épaisse.

Un moyen évident de réduire l'impact de ces artefacts est de choisir un pas plus petit. Cependant, un pas plus fin signifie également un nombre d'étapes beaucoup plus important avant de détecter les intersections. Heureusement, plusieurs travaux ont été développés dans le but de corriger ces défauts sans avoir à passer par cette solution plus coûteuse [6, 142]. Le principe de ces méthodes est de précalculer, en chaque pixel du champ de hauteurs, la distance maximale qui peut être parcourue en espace texture sans qu'aucun rayon ne

traverse deux fois la surface représentée. Ainsi, en utilisant ce pas adaptatif, la recherche est garantie de ne jamais manquer aucune intersection. De plus, le nombre d'étapes pour la recherche s'en trouve généralement réduit, ce qui entraîne une accélération du rendu.

Dans notre cas, l'amplitude du relief est vraiment faible, ce qui limite les cas de figure où de tels artefacts apparaissent. Nous n'avons donc, pour le moment, intégré aucune de ces optimisations. Les mesures de performances que nous avons proposées ont donc été effectuées avec l'algorithme de *relief mapping* standard. Cependant, étant donné que notre champ de hauteurs n'est pour le moment stocké que sur une texture RVB (hauteur+normale), une quatrième valeur peut encore être associée à chaque pixel en utilisant cette fois une texture RVBA. La quatrième composante pourrait alors servir à stocker le pas adaptatif utilisé par de telles méthodes.

■ 5.4 Discussion sur l'acquisition de la texture bidirectionnelle

Nous allons maintenant discuter de la reconstruction de la texture bidirectionnelle à partir des photographies acquises. Nous avons, en effet, rencontré d'importantes difficultés lors de l'estimation des paramètres du modèle de Lafortune. Cette estimation est traditionnellement effectuée à l'aide de l'algorithme de Levenberg-Marquardt, qui est un algorithme d'optimisation non linéaire. À ce titre, il est particulièrement sensible aux conditions initiales, et peut être sujet à des problèmes de stabilité concernant la convergence. Ainsi, nous nous sommes retrouvés dans le cas où de nombreux pixels de la toile n'ont pas pu être estimés correctement.

Bien que le modèle de Lafortune soit utilisé dans de nombreux travaux, il semblerait que nous ne soyons pas les premiers à rencontrer de tels problèmes : certaines publications semblent également y faire allusion [42, 74, 119]. Il semblerait même que l'estimation de modèles de Lafortune à plus de deux Lobes relève du «miracle» : le nombre de paramètres à estimer simultanément devient en effet très important, et l'algorithme devient alors de plus en plus instable et demande des temps de calcul à ce point faramineux qu'il en devient impraticable [119].

Heuberger *et al.* [74] rapportent à ce sujet les observations suivantes : de manière générale, alors que le paramètre de couleur diffuse converge très rapidement, le lobe spéculaire, quant à lui, met beaucoup de temps à converger, lorsqu'il y parvient. Apparemment, si l'exposant du lobe est initialisé à une valeur proche de la solution (ce qu'il est évidemment difficile de garantir *a priori*), le processus d'optimisation donnera un résultat correct. Dans le cas contraire, l'algorithme est incapable de converger.

De notre côté, nous pensons également que ces problèmes de stabilité proviennent, au moins en partie, du fait que le recalage des différentes photographies sur la géométrie ne soit pas suffisamment précis. En effet, des décalages d'un seul pixel peuvent engendrer un bruit important au niveau de l'information bidirectionnelle de chaque point. Nous en revenons donc à ce dont nous discutons dans le bilan du chapitre 6, sur la nécessité d'une méthode pour le raffinement des calibrages de points de vue.

McAllister *et al.* [116], qui travaillent avec un modèle de Lafortune identique au nôtre, ont utilisé une autre méthode d'estimation basée, entre autre, sur une linéarisation partielle du problème. Cette méthode est à la fois plus stable, mais également plus rapide que l'algorithme de Levenberg-Marquardt. Nous avons donc opté pour cette solution, même si cela n'a pas permis de résoudre le problème pour absolument tous les pixels.

■ 5.5 Discussion sur la compacité des données

L'un des avantages de notre méthode est l'absence de maillage pour représenter la géométrie, ce qui permet une réduction de la consommation mémoire. En effet, seules quatre textures flottantes 16bits (une RVBA et trois RVB) et une liste de boîtes sont nécessaires pour complètement représenter la copie numérique finale, texture bidirectionnelle incluse. La taille totale après traitement est d'environ 11Mo. À titre de comparaison, le maillage correspondant contient 815K triangles. En remplaçant la texture représentant le champ de hauteurs par ce maillage, nous aboutissons à une copie numérique de 28Mo pour représenter la même information.

On peut également remarquer que, puisque la quasi totalité des données est représentée sous forme d'images (à l'exception des boîtes, qui ne consistent qu'une proportion négligeable de la taille totale de la copie), des algorithmes classiques de compression d'images (JPEG, PNG, etc.) peuvent éventuellement être utilisés pour réduire davantage le coût de stockage.

6 Bilan

Dans ce chapitre, nous avons introduit une nouvelle chaîne de traitements permettant la numérisation et la visualisation de peintures d'art. Contrairement à la plupart des méthodes existantes, nous tenons compte de la géométrie de la toile au moment du rendu, étant donné que celui-ci peut être important lorsque le tableau présente des empâtements prononcés, comme dans le cas de la peinture au couteau.

Cette chaîne commence donc par l'acquisition de la géométrie et de la couleur du tableau, cette seconde information étant capturée pour de nombreux points de vue et de nombreuses directions d'incidence lumineuse.

Une phase de prétraitement prépare alors les données acquises de manière à générer la copie numérique finale, destinée à être visualisée de manière interactive. Ces traitements consistent à reconstruire la texture bidirectionnelle de la toile, afin d'en permettre la restitution pour n'importe quelle condition d'éclairage et d'observation, puis à générer un champ de hauteurs à partir des données géométriques. Ce champ de hauteurs est alors classifié en deux catégories : la première représentant le relief négligeable et l'autre, le relief significatif.

Au moment du rendu, un simple plan et l'algorithme du *bump mapping* sont alors utilisés pour afficher rapidement le relief négligeable. Le relief significatif est alors superposé à ce plan lors d'une seconde passe de rendu, durant laquelle l'algorithme du *relief mapping* est employé pour corriger les erreurs de parallaxe là où cela est nécessaire. Dans la mesure où ce second algorithme est plus coûteux, un mécanisme adaptatif a été mis en place afin d'en moduler l'usage en fonction du point de vue.

Nos résultats ont montré que les performances obtenues permettaient un rendu temps réel et de qualité, et que la représentation sous forme de champs de hauteurs aboutissait à des copies numériques plus compactes que lorsque des maillages 3D devaient être stockés.

Ces études ont cependant soulevé un certain nombre de questions. Commençons par quelques mots sur la texture bidirectionnelle. Depuis de nombreuses années, il est communément admis que le modèle de Lafortune n'est de loin pas le plus précis qui soit pour la représentation de fonctions de réflectances [45, 48]. Cependant, ce modèle est très pratique de par sa simplicité et sa modularité : on peut, en effet, modifier à loisir le nombre de lobes utilisés, en fonction de la précision désirée. Utilisé pour estimer des données issues de mesures réelles, il est évident que le modèle à un seul lobe spéculaire que nous avons utilisé est beaucoup trop approximatif. C'est cependant le seul, à l'heure actuelle, qui permette d'afficher une si grande variété de matériaux en même temps tout en garantissant d'excellentes performances. Nous pensons, toutefois, qu'une étude devrait être menée sur les caractéristiques photométriques des peintures et vernis les plus utilisés dans le domaine de l'art. D'une part, cela pourrait permettre de déterminer les modèles de réflectances les plus adaptés à notre cas. D'autre part, si les paramètres du modèle en question doivent être estimés par un processus d'optimisation, comme c'est le cas pour Lafortune, cela pourrait également permettre de définir un jeu de paramètres à utiliser comme conditions initiales pour garantir la convergence de ce processus.

Concernant le rendu de la géométrie, nous avons fait allusion aux limitations du *relief mapping*, aboutissant à la conclusion que notre technique ne conviendrait pas à des éléments artistiques présentant un relief plus important, tels que des bas-reliefs, par exemple. Nous pensons que cette assertion mérite d'être vérifiée, d'autant plus si l'on considère le papier original de Policarpo *et al.* [143], où le *relief mapping* est utilisé avec des objets de relief relativement important. Il faut d'ailleurs reconnaître que le tableau sur lequel nous avons travaillé n'est pas un bon sujet d'exemple : nous pensons que son relief n'est pas encore suffisamment prononcé pour justifier l'emploi d'une technique telle que la nôtre.

A l'heure actuelle, notre méthode ne permet de travailler que sur des tableaux de petites tailles : étant donné que nous construisons le champ de hauteurs directement à partir d'une image télémétrique, il faut impérativement que la toile soit entièrement visible par le scanner 3D. Il faudrait donc étendre le procédé de numérisation de manière à pouvoir traiter des grandes toiles, nécessitant une numérisation par morceaux. À première vue, les modifications nécessaires pour y arriver ne sont pas si évidentes, et risquent même de remettre totalement en cause les étapes d'acquisition et de prétraitement.

Un autre point qui peut prêter à interrogation est l'approximation de la toile par un plan lors de la création du champ de hauteurs. Il se peut que le support du tableau soit déformé, auquel

cas une grande partie de la toile sera classée comme relief significatif simplement parce que les distorsions auxquelles elle est soumise ne permettent pas de l'approximer correctement par un plan. De telles déformations sont couramment observées sur des œuvres anciennes [64, 94], aussi faudrait-il prendre en compte ce phénomène, soit en approximant la toile par un autre type de surfaces paramétriques (quadriques, par exemple), soit en considérant une représentation planaire par morceaux, auquel cas des techniques d'extraction de régions planaires dans des images télémétriques telles que celle de Stamos *et al.* [176] pourraient être employées.

Enfin, le dernier point qu'il nous semble important de remarquer est le fait que les performances soient moins bonnes pour notre méthode que pour un rendu basé maillages lorsque le point de vue est très proche de la toile. Bien que nous ayons apporté des explications à cet état de faits, nous nous demandons s'il ne faudrait pas introduire un troisième niveau dans le rendu hybride, utilisant le maillage en lui-même lorsque de telles conditions d'observation se produisent. Nous préfererions tout de même une méthode alternative, dans la mesure où utiliser le maillage remettrait en cause notre argument de compacité.

Troisième partie

— Conclusion —

- Conclusion générale -

Nous allons maintenant clore ce document par une conclusion générale portant sur l'ensemble des travaux qui ont été présentés. En premier lieu, la section 1 résume les quelques contributions auxquelles nos travaux ont donné lieu, celles-ci se situant plus particulièrement dans les domaines de la numérisation et de la visualisation. La section 2 débat ensuite des limitations de ces contributions, avant de déboucher, en section 3, sur les perspectives possibles pour ces travaux.

1 Rappel des contributions

Au cours des travaux qui ont été présentés, nous nous sommes intéressés à la chaîne de traitements allant de la numérisation d'objets réels jusqu'à leur visualisation. Nous avons adopté, pour cela, deux approches différentes : une première approche générale, visant à établir un protocole de numérisation utilisable dans des conditions arbitraires, et une seconde approche plus spécifique dédiée à la numérisation de tableaux, qui nous a servi à illustrer le fait que les techniques très ciblées ne doivent pas nécessairement être dévalorisées sous prétexte de vouloir rester coûte que coûte générales.

Concernant l'approche générale, notre principale contribution est un protocole¹ permettant l'acquisition de champs lumineux surfaciques, c'est-à-dire de copies numériques constituées de la géométrie de l'objet et d'une information d'apparence capturée en environnement lumineux fixe mais tenant compte des variations liées aux déplacements de l'observateur. Les deux problématiques sous-jacentes à la capture d'une telle information sont le recalage géométrique, survenant du fait que la géométrie de l'objet doit être capturée en plusieurs morceaux, et le calibrage de points de vue, qui permet d'associer à la géométrie acquise l'information chromatique contenue dans un ensemble de photographies capturées depuis des points de vue arbitraires.

¹cf. chapitre 6.

L'objectif de nos travaux étant de réaliser la mesure d'œuvres d'art sur site, nous avons proposé de résoudre ces deux problèmes grâce à un matériel peu onéreux et très mobile, et par une technique ne demandant ni contact avec l'objet, ni contrôle spécifique de l'environnement d'acquisition. De plus, ce protocole présente les atouts suivants :

- les tâches de recalage et de calibrage ne nécessitent aucune intervention de la part de l'opérateur chargé de la mesure, ce qui permet d'éviter le fastidieux travail que constitue généralement la réalisation manuelle de ces tâches.
- des résultats intermédiaires peuvent être fournis suffisamment rapidement pour permettre un contrôle interactif de la mesure, ce qui est un point essentiel pour éviter les mauvaises surprises *a posteriori*.

Ces deux points sont à mettre en opposition avec les travaux présentés dans la littérature, concernant les campagnes de numérisation sur site : soit la quasi totalité des traitements doit être effectuée manuellement, nécessitant parfois jusqu'à plusieurs dizaines d'heure de travail humain, soit l'automatisation repose sur du matériel complexe et coûteux, pas forcément évident à déployer. D'autres approches existent, bien entendu, pour résoudre ces problèmes automatiquement et de manière purement logicielle, mais soit elles nécessitent de contrôler l'environnement d'acquisition, ce qui n'est pas forcément possible sur site, soit elles s'exécutent en des temps de calcul rendant toute interactivité impossible. Notre protocole présente donc un intérêt significatif par rapport aux travaux antérieurs.

Concernant, cette fois-ci, l'approche dédiée, nous avons proposé un protocole spécifiquement conçu pour l'acquisition et la visualisation de peintures d'art². Ce protocole, bien entendu, s'intéresse à l'acquisition de l'apparence du tableau, en proposant notamment une capture et une restitution temps réel d'une texture bidirectionnelle complète, mais sa principale contribution par rapport aux travaux existants est la capture et l'utilisation du relief de la toile lors du processus de rendu, là où les autres méthodes se contentent généralement de l'approximer par un simple plan. De nombreuses toiles peintes présentent en effet un relief parfois prononcé qui doit être pris en compte lors du rendu, sous peine, à défaut, de nuire au réalisme de la restitution.

L'algorithme que nous proposons considère donc l'information géométrique, mais exploite le fait que la forme globale reste tout de même principalement planaire pour passer dans un mode de représentation plus adapté : un champ de hauteurs. Cette représentation permet d'éviter d'avoir à manipuler des maillages trop détaillés et, *de facto*, peu praticables, mais surtout d'utiliser des algorithmes de rendu *ad hoc*. Nous avons ainsi proposé de combiner deux de ces algorithmes : le *bump mapping* et le *relief mapping*, le choix entre l'un et l'autre étant guidé par un mécanisme adaptatif déterminant si le relief est suffisamment négligeable par rapport au point de vue courant pour être ou non simplement approximé. Ces travaux constituent une contribution intéressante, dans la mesure où nous avons pu constater que les travaux s'attachant à la numérisation de tableaux dans un but de visualisation sont relativement peu nombreux, et que les rares travaux en question ne se préoccupent généralement que de l'information picturale.

Sortant du cadre de la numérisation à proprement parler, nous avons également abordé des

²*cf.* chapitre 9.

problématiques liées à la visualisation temps réel³. Nous avons, bien entendu, commencé par proposer un moyen de visualiser les champs lumineux surfaciques acquis grâce à notre premier protocole (protocole général). Nous avons, pour cela, adapté le modèle des textures polynomiales existant dans la littérature de manière à prendre en compte les variations liées au point de vue. Nous avons alors détaillé la procédure permettant de reconstruire, à l'aide de ce modèle, une texture qui soit utilisable par le matériel graphique, et ainsi permettre une visualisation interactive et plausible par rapport aux comportements des matériaux composant l'objet.

Le second point qui a été abordé est l'utilisation de champs lumineux surfaciques pour simuler le phénomène de réfraction observable dans le cas d'objets transparents. Ce second point constitue en fait notre réelle contribution dans le domaine de la visualisation. Les champs lumineux surfaciques sont ici utilisés pour stocker, sur la surface même, la direction du rayon quittant l'objet après en avoir parcouru l'intérieur, et ce pour toutes les directions d'observation possibles. Nous avons, pour l'occasion, proposé un nouveau modèle de représentation pour ces champs, utilisant une compression de l'information directionnelle (dépendance au point de vue) par une projection dans la base des harmoniques sphériques. L'évaluation directe des fonctions harmoniques par la carte graphique permet d'obtenir une restitution temps réel du phénomène. Par rapport aux techniques temps réel existantes, notre méthode est capable de prendre en compte plusieurs déviations successives des rayons lumineux dans l'objet, ce qui permet de simuler la réfraction pour des géométries complexes, de même que pour des objets constitués de plusieurs milieux, avec chacun un indice de réfraction différent. Un autre avantage du modèle par harmoniques sphériques est la nature progressive de la compression : la précision du modèle ainsi que les ressources qu'il demande dépendent tous deux de l'ordre de reconstruction choisi. Ainsi, selon les besoins, il est possible de privilégier soit la précision, soit les performances.

Nous nous permettons de faire remarquer que toutes les contributions que nous venons de résumer ont donné lieu à des publications ou des communications diverses, dont la liste est fournie en annexe D.

2 Limitations et perspectives à courts termes

Nous allons discuter ici, point par point, des limitations des contributions précitées, ainsi que des difficultés rencontrées lors de leur mise en œuvre. La plupart de ces limitations ont déjà été évoquées dans les bilans des chapitres concernés, aussi cette section fait-elle principalement office de rappel et de synthèse. Nous allons également donner les pistes révélées par ces problèmes pour nous aider à poursuivre nos travaux sur le court et le moyen termes.

³ cf. chapitre 8.

■ 2.1 Acquisition et visualisation de champs lumineux surfaciques

Concernant l'acquisition, la procédure permettant d'automatiser et de rendre interactif le recalage géométrique et le calibrage de points de vue présente l'inconvénient d'augmenter le temps d'acquisition en lui-même. En effet, une image de phase étant associée à chaque donnée acquise, le temps nécessaire à son acquisition doit forcément être ajouté au temps global de mesure. Ce surcoût est particulièrement pénalisant pour le calibrage de points de vue : leur nombre peut rapidement devenir très important, ce qui ajoute un temps non négligeable à la mesure par rapport au fait de prendre un simple cliché à l'aide d'une caméra à l'épaule. Le surcoût en question n'est actuellement pas quantifiable : nous utilisons actuellement un logiciel de pilotage du scanner qui n'a pas été conçu spécifiquement pour effectuer les tâches que nous lui demandons. Aussi, de nombreuses manipulations sont actuellement nécessaires pour arriver à nos fins, ce qui multiplie d'autant les temps de mesure. Ces manipulations peuvent aisément être automatisées, mais cela attend encore d'être fait, et nous ne disposons pas des sources permettant d'effectuer nous-même les modifications dans le logiciel en question. Ce que nous pouvons dire, à l'heure actuelle, est que ce surcoût semble nécessaire : il n'existe pas à notre connaissance de méthodes capables d'effectuer ce calibrage de manière automatique et dans des conditions de mesure arbitraires. Il pourrait en revanche être réduit en utilisant, par exemple, un modèle de lumière structurée plus souple et plus interactif, comme ceux utilisés en vision pour le guidage de capteurs par asservissement visuel, ou en se penchant sur les travaux qui s'intéressent à la problématique de la numérisation 3D en temps réel [36, 157].

Nous avons également pu remarquer que, contrairement au recalage géométrique, le calibrage de points de vues pouvait présenter un certain manque de robustesse : certaines des photographies calibrées présentent des défauts d'alignement avec le modèle 3D lorsque celui-ci est projeté dans l'espace image de la caméra d'acquisition. Cependant, le résultat est rarement très éloigné de la solution théorique, aussi pensons-nous qu'il serait bon de mettre au point un algorithme de raffinement de la localisation de ces points de vue, au même titre que cela a été fait dans le cas du recalage géométrique. Ce point est d'autant plus important que, lors de la reconstruction de la texture utilisée par l'algorithme de visualisation, un décalage de quelques pixels seulement peut engendrer une altération flagrante de l'information (effet de flou au niveau des détails picturaux). Des techniques d'ajustement de textures sont alors à envisager. Nous pensons qu'une approche telle que celle de Bernardini *et al.* [10] est intéressante dans notre cas : elle utilise des «images télémétriques colorées», c'est-à-dire que chaque point se voit assigner la couleur perçue par la caméra du scanner au moment où celle-ci acquiert la géométrie. Cette information peut alors permettre d'ajuster correctement les photographies prises indépendamment de la géométrie avec cette information de couleur dont le recalage est implicitement donné. Dans la mesure où l'environnement d'acquisition ne peut pas forcément être contrôlé dans notre cas, cette approche a l'avantage d'être plus robuste que les techniques plus traditionnelles basées sur la mise en correspondance de silhouettes.

Au niveau de la visualisation des champs lumineux surfaciques acquis, nous avons vu que le modèle de textures polynomiales révélait bien vite ses limites du fait du lissage important qu'il engendre sur l'information encodée. De plus, même lorsque le résultat semble

plausible, celui-ci peut être souvent éloigné des données réelles (lobes spéculaires déplacés ou déformés). Les techniques comme celle proposée par Chen *et al.* [26], basées sur des analyses en composantes principales dans l'espace 4D du champ lumineux surfacique, semblent être, actuellement, ce qui aboutit aux meilleurs résultats, autant en termes d'interactivité que de qualité visuelle, tout en restant relativement simples à mettre en œuvre. Nous devrions certainement nous pencher sur ces travaux, d'autant plus que la fidélité est importante lorsqu'il s'agit de restituer des œuvres d'art numérisées. Nous avons initialement choisi d'utiliser les textures polynomiales du fait de leur simplicité, mais ces premiers essais nous aurons finalement permis d'affirmer que ce modèle trop simpliste ne sied pas à la représentation de champs lumineux surfaciques.

Le second point qui a été révélé par la visualisation des copies numériques est la lourdeur induite par les maillages de résolutions trop élevées que fournissent les dispositifs actuels de mesure de formes. Le nombre trop important de primitives géométriques constitue effectivement un véritable goulet d'étranglement lors de la visualisation, lorsque celles-ci doivent être transmises à la carte graphique. Ce constat met en évidence la nécessité d'ajouter une étape supplémentaire de simplification de maillages à la chaîne de traitements géométriques. Cette étape peut éventuellement être incorporée à la phase d'intégration en utilisant, pour ce faire, un algorithme adaptatif de fusion [146, 161]. Les pour et les contre d'avoir la simplification de maillages décorrélée de l'intégration sont donc à étudier.

■ 2.2 Visualisation temps réel d'objets réfringents

La principale limitation quant au fait d'utiliser une représentation en harmoniques sphériques pour la simulation du phénomène de réfraction est que le processus de rendu est assez lourd, et ce autant en termes de consommation mémoire que de temps de calcul. Cela peut malheureusement rendre la méthode peu praticable dans le cas d'applications telles que des jeux vidéo, où de nombreux autres éléments doivent également être calculés et affichés en même temps. Des moyens doivent donc être mis en œuvre pour réduire les besoins en ressources.

L'autre grande limitation de cette méthode est le fait que la géométrie des objets représentés soit obligatoirement statique. Cette limitation est, en réalité, imposée par de nombreuses méthodes temps réel, du fait que des précalculs sont souvent nécessaires pour atteindre la qualité visuelle escomptée. Or, une méthode représentant une information directionnelle sur la surface d'un objet à l'aide d'harmoniques sphériques, comme dans notre cas, a été proposée par Sloan *et al.* [171] pour des objets légèrement déformables. Vérifier si une telle approche peut être mise en œuvre dans notre cas pourrait rendre la méthode plus attractive, en ce sens qu'elle serait, du coup, beaucoup moins restrictive.

Enfin, comme dans le cas précédent de la visualisation d'œuvres d'art numérisées, nous souhaiterions comparer notre méthode par harmoniques sphériques à d'autres modèles de représentation de champs lumineux surfaciques. Sachant que ces modèles ont initialement été conçus pour encoder une luminance, nous aimerions vérifier dans quelle mesure ils sont également capables d'encoder une information dont le comportement est plus chaotique,

telle que la direction de sortie d'un rayon d'observation, comme c'est le cas pour notre modèle.

■ 2.3 Numérisation et visualisation de tableaux

À propos des travaux portant sur la numérisation de tableaux d'art, nous avons pu constater que notre algorithme de visualisation, tel qu'il est défini actuellement, ne permet pas d'étendre facilement la classe des objets représentables. La principale limitation vient du fait que la forme globale ne soit approximée que par un simple plan. Si le support du tableau est déformé (ce qui arrive couramment avec des œuvres anciennes), ce modèle d'approximation devient en effet trop simpliste : le champ de hauteurs utilisé pour représenter la toile doit alors encoder un relief qui n'est pas simplement dû aux couches de peintures et qui risque de présenter une amplitude beaucoup trop importante. Ainsi, notre mécanisme adaptatif risque d'être mis en défaut. Nous envisageons donc de résoudre le problème posé par de tels tableaux en proposant une représentation de la toile par des morceaux de plans, chacun contenant son propre champ de hauteurs. Ainsi, même si la forme globale présente de fortes variations, nous pourrions garantir que l'information représentée à l'aide des champs de hauteurs ne correspondrait qu'à un relief de faible amplitude. De plus, une telle représentation nous permettrait de prendre également en compte des éléments artistiques tels que des bas-reliefs.

À nos yeux, l'un des intérêts de la méthode proposée est de ne pas avoir besoin de conserver de maillages 3D. Nous avons cependant pu constater qu'un rendu basé sur un tel maillage était plus performant que notre méthode sous certaines conditions d'observation, à savoir lorsque l'observateur se situe vraiment très près du tableau. Sur ce point, nous voyons deux éléments importants à étudier dans un proche avenir :

- tout d'abord, nous nous interrogeons sur la nécessité d'un troisième niveau d'échelle dans notre rendu hybride pour traiter le cas des points de vue les plus proches. Nous aimerions, pour cela, éviter d'avoir à utiliser le maillage, afin de conserver le gain de mémoire que cela occasionne.
- ensuite, nous envisageons d'étudier précisément les causes des pertes de performances de notre méthode dans de tels cas. Les études que nous avons présentées nous ont d'ailleurs déjà donné des pistes à ce sujet.

Enfin, concernant la capture de l'apparence des tableaux, nous pensons que l'acquisition précise, effectuée en laboratoire, des matériaux les plus utilisés en peinture dans le domaine de l'art (pigments, vernis) pourrait permettre de constituer une base d'étude pour l'élaboration de nouveaux modèles de représentation de l'information bidirectionnelle dans le cas spécifique de cette catégorie d'objets. Ces acquisitions pourraient également permettre d'établir une base de connaissances pour faciliter l'estimation de tels modèles à partir de données réelles lors de la mesure des tableaux en eux-mêmes. Il faut d'ailleurs noter que tous les problèmes liés au calibrage photométrique des différents dispositifs utilisés lors des mesures n'ont absolument pas été abordés au cours de ces travaux de thèse. Ce point reste donc à traiter pour le moment.

D'une manière générale, on peut dire qu'il y a encore du chemin à faire pour passer du prototype laboratoire actuel à un système réellement exploitable sur site. Notamment, la structure rigide permettant la localisation de la source dans l'espace objet doit être revue. On peut d'ailleurs noter à ce sujet que des structures beaucoup plus mobiles ont été développées dans ce but [39]. Il y a aussi le problème des trop nombreuses variables fixées manuellement dont nous ne savons pas, aujourd'hui, comment en déterminer automatiquement les valeurs. Enfin, il faudra également se poser la question de la numérisation du cadre du tableau, problème qui n'a pas du tout été considéré ici.

3 Perspectives à longs termes

Au vu des travaux de numérisation sur site réalisés à l'heure actuelle, on peut dire sans trop se tromper que l'un des enjeux majeurs est l'automatisation des traitements. La chaîne qui part de l'acquisition d'objets réels et qui aboutit à leur visualisation est extrêmement complexe en soi, et les différents maillons qui la composent nécessitent encore trop souvent des interventions manuelles. Nous avons tenté, durant cette thèse, d'apporter quelques solutions pour l'automatisation de certains de ces traitements, plus particulièrement sur les aspects liés au recalage.

Nous aimerions cependant aller plus loin dans cette voie. Nous avons évoqué la première ébauche d'une approche visant à effectuer automatiquement le filtrage des maillages 3D numérisés⁴. Bien entendu, cette première idée doit encore être étoffée. Cependant, nous voudrions aller plus loin en proposant un mécanisme de filtrage basé sur les données photométriques acquises. Pour ce faire, nous envisageons un pipeline pour l'acquisition de la géométrie et de la texture bidirectionnelle d'un objet qui soit capable d'estimer conjointement, et uniquement à partir d'images :

- la position d'un illuminant, c'est-à-dire de la source lumineuse utilisée pour acquérir la dépendance des matériaux à l'illumination. Cette estimation permettrait alors d'utiliser une source mobile, simplement placée à la main, sans contrainte matérielle pour effectuer la localisation.
- la BRDF du matériau, ou plus spécifiquement l'information bidirectionnelle perçue, c'est-à-dire incluant également les effets d'ombrage et d'interréflexion.
- les normales à la surface, étroitement liées à l'illumination résultant de l'information bidirectionnelle estimée. Ces normales, calculées de manière purement photométrique, permettraient alors de corriger celles obtenues à partir de l'acquisition géométrique de façon à éliminer le bruit de numérisation.

Nous pensons en effet que pour les régions où le matériau reste à peu près uniforme, les nouvelles normales estimées à partir de ce matériau évolueront également de manière relativement lisse, rendant ainsi la possibilité d'un tel débruitage effective. De plus, un tel pipeline permettrait également d'étendre notre protocole au cas de l'acquisition de textures

⁴ cf. chapitre 7.

bidirectionnelles, ce qui n'a pour le moment été abordé que dans le cas du prototype de laboratoire pour la numérisation de tableaux.

À plus longs termes, nous aimerions également élaborer un système d'acquisition entièrement interactif, qui soit capable d'effectuer la quasi totalité des tâches en des temps permettant de fournir un retour rapide à l'opérateur chargé de la mesure. Au vu de l'état actuel de la recherche, ce point semble, lui aussi, être crucial. En effet, le manque de moyens de contrôle devient de plus en plus flagrant, certainement du fait que la demande envers les technologies de numérisation est sans cesse grandissante. Pour éviter que la maîtrise de ces technologies ne continue de se cantonner aux seuls laboratoires de recherche, des outils de contrôle robustes, conviviaux et le plus automatiques possibles sont absolument nécessaires.

Quatrième partie

— Annexes —

- Glossaire photométrique -

Un certain nombre de notions utiles de photométrie vont être définies ici. Il est à noter que les quantités photométriques sont dérivées des quantités radiométriques : elles représentent le même type d'informations, mais la photométrie s'intéresse spécifiquement à l'énergie lumineuse et exprime donc toutes ces quantités en tenant compte des courbes de sensibilité de l'œil humain aux différentes longueurs d'onde.

Ainsi, si deux sources émettent avec une même puissance énergétique dans des longueurs d'onde différentes, l'interprétation de ces puissances d'un point de vue photométrique est différente : si l'une des deux sources n'émet pas dans le spectre visible de la lumière, sa puissance lumineuse est considérée comme nulle.

Nous allons ici définir les quantités radiométriques. Un tableau de correspondances entre radiométrie et photométrie est proposé en fin de cette annexe, avec un récapitulatif des unités employées dans chacun des deux cas. Les courbes de sensibilité de l'œil en fonction de la longueur d'onde sont également fournies, à titre indicatif, ainsi que les formules permettant d'effectuer les conversions entre puissance énergétique et puissance lumineuse.

1 Définitions radiométriques

Angle solide ω : l'angle solide n'est pas, en soi, une notion photométrique ou radiométrique. Cependant, comme il est utilisé pour définir les quantités qui vont suivre, nous allons commencer par expliquer ce dont il s'agit. L'angle solide peut être interprété comme la visibilité d'une surface d'aire dS' depuis un point donné. Il est déterminé en calculant l'aire de S projetée sur la sphère unité centrée en ce point (voir image A.1). L'unité d'angle solide est le stéradian (sr). L'angle solide maximum est de $4\pi sr$.

Si l'on considère un élément de surface dS au lieu d'un simple point, il est possible de définir l'*angle solide projeté*, qui correspond à la projection de l'angle solide sur le plan

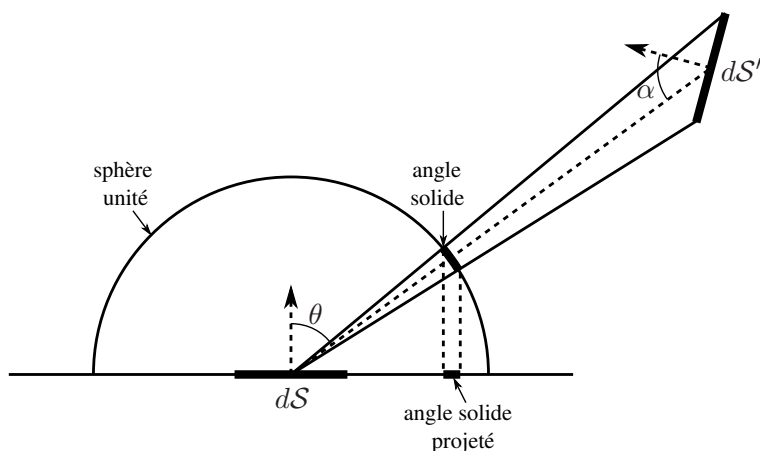


Figure A.1 – Réflexion et réfraction d'un rayon lumineux à l'interface de deux milieux d'indices de réfraction différents.

tangent de cet élément de surface. Ainsi, si l'angle solide est $d\omega$ et l'angle entre la normale en dS et la direction associée à cet angle solide est θ , la valeur de l'angle solide projeté est donnée par $d\omega \cos \theta$.

Flux énergétique Φ : puissance énergétique d'une source, c'est-à-dire à la quantité d'énergie émise dans tout l'espace durant une seconde. En termes d'énergie lumineuse, le flux est corrélé au nombre de photons émis durant cette même période de temps.

Exprimé en Watt (W).

Intensité énergétique I : flux émis par une source dans une direction donnée par unité d'angle solide.

$$I = \frac{d\Phi}{d\omega} \quad (\text{A.1})$$

Exprimée en Watt par stéradian ($W \cdot sr^{-1}$).

Éclairement énergétique E : flux reçu par unité de surface.

$$E = \frac{d\Phi}{dS} \quad (\text{A.2})$$

Exprimé en Watt par mètre carré ($W \cdot m^{-2}$).

Luminance énergétique L : flux émis ou reçu par un élément de surface dans une direction donnée. Dépend à la fois de l'aire de l'élément de surface et de l'angle solide projeté associé à la direction d'observation.

$$L = \frac{d^2\Phi}{dS d\omega \cos \theta} = \frac{dE}{d\omega \cos \theta} \quad (\text{A.3})$$

Le terme en $\cos \theta$, où θ est l'angle entre la normale en S et la direction d'émission/réception, apparaît du fait que c'est un angle solide projeté qui est considéré. Cet angle solide projeté

Photométrie		Radiométrie	
Quantité	Unité	Quantité	Unité
flux énergétique (ou puissance énergétique)	Watt (W)	flux lumineux (ou puissance lumineuse)	lumen (lm)
intensité énergétique	$W.sr^{-1}$	intensité lumineuse	candela (cd) $\Leftrightarrow lm.sr^{-1}$
luminance énergétique	$W.m^{-2}.sr^{-1}$	luminance lumineuse	$lm.m^{-2}.sr^{-1}$
éclairage énergétique	$W.m^{-2}$	éclairage lumineux	lux (lx) $\Leftrightarrow lm.m^{-2}$

Tableau A.1 – Parallèle entre les quantités radiométriques et photométriques (unités SI).

permet de pondérer l'intensité émise/reçue en fonction de l'orientation de l'élément de surface (surface de front : intensité maximale, surface rasante : intensité faible).

Exprimée en Watt par stéradian par mètre carré ($W.sr^{-1}.m^{-2}$).

2 Correspondance radiométrie / photométrie

Le tableau A.1 illustre le parallèle existant entre les quantités radiométriques et photométriques.

Comme nous l'avons expliqué au début de cette annexe, si la luminance, l'intensité et l'éclairage se définissent de la même manière par rapport au flux, ce dernier ne représente cependant pas les mêmes grandeurs de part et d'autre. En radiométrie, le flux énergétique décrit une puissance en tant que telle alors que, en photométrie, le flux lumineux tient compte de la sensibilité de l'œil aux différentes longueurs d'onde de l'énergie lumineuse.

Cette sensibilité est donnée par les *fonctions de sensibilité spectrale normalisée* ν_{ph} et ν_{sc} , dont les courbes sont représentées sur la figure A.2, représentant les sensibilités respectives des visions photopique (diurne) et scotopique (crépusculaire).

La correspondance entre flux énergétique Φ_e et flux lumineux Φ_l pour une longueur d'onde λ donnée est alors la suivante :

$$\Phi_l(\lambda) = 683 \nu(\lambda) \Phi_e(\lambda), \quad \text{avec } \nu \in \{\nu_{ph}, \nu_{sc}\} \quad (\text{A.4})$$

Ainsi, lorsque la sensibilité atteint son maximum théorique de 1 (à $555nm$ pour ν_{ph} et à $507nm$ pour ν_{sc}), une puissance énergétique de $1W$ équivaut à $683lm$.

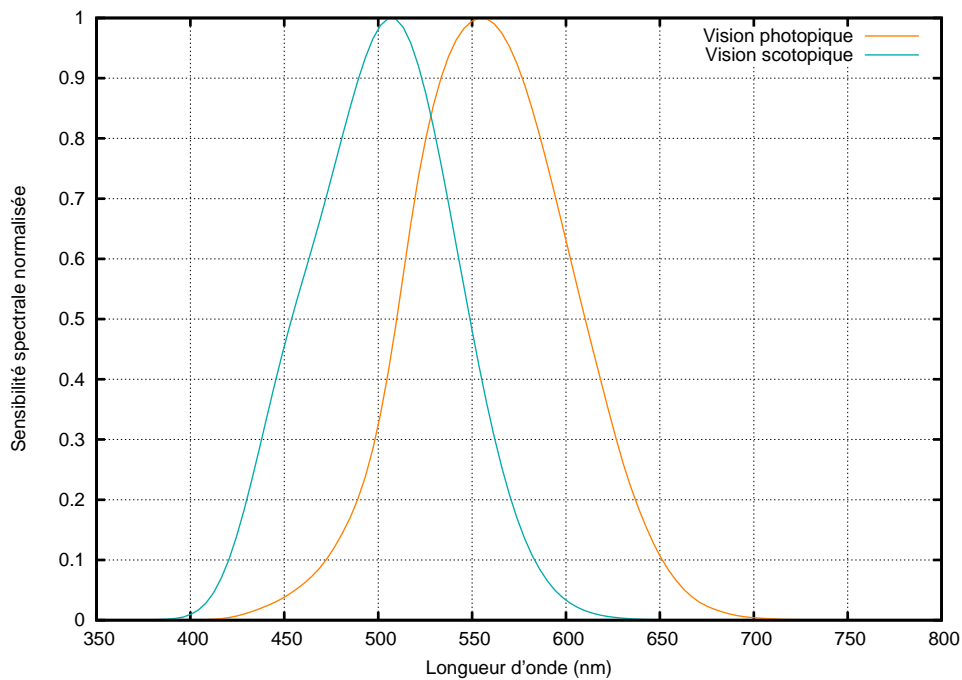


Figure A.2 – Courbes de sensibilité spectrale normalisée pour les visions photopique (diurne) et scotopique (crépusculaire).

3 Acquisition de quantités photométriques

Il faut noter que lors de l'acquisition par une caméra numérique, l'information perçue en chaque pixel ne correspond pas directement à la luminance quittant la surface observée. Si L_p est la luminance en un pixel de l'image et L_r la luminance réelle renvoyée vers la caméra par l'élément de surface observé en ce pixel, le lien entre les deux peut être défini par la relation suivante :

$$L_p = f(t \cdot L_r + N^c) \quad (\text{A.5})$$

où t représente le temps d'exposition sous lequel l'image a été acquise, N^c représente un bruit additif provenant du processus de formation de l'image numérique et dont les sources peuvent être diverses, et f est la *fonction de réponse* de la caméra. Cette fonction peut dépendre de différents paramètres, comme par exemple de la sensibilité des cellules CCD, ou de différents traitements introduits par le constructeur.

La principale difficulté pour mettre en relation L_p et L_r consiste donc à déterminer la fonction de réponse f de la caméra. Pour plus d'information sur ce sujet, l'article de Robertson *et al.* [149] peut être consulté. Une méthode pour déterminer cette fonction à l'aide de plusieurs images acquises avec temps d'exposition différents y est notamment présentée.

Il faut noter que cette discussion ne considère absolument pas le problème du calibrage des couleurs d'une caméra.

- Réflexion et réfraction -

Lois de Snell-Descarte et équations de Fresnel

Lorsqu'un rayon lumineux passe d'un milieu d'indice de réfraction η_1 à un second milieu d'indice de réfraction η_2 , ce rayon est à la fois réfléchi et réfracté. Les lois de Snell-Descartes permettent alors de définir les directions des rayons réfléchis et transmis en fonction des conditions d'incidence.

Soit \vec{n} la normale au point d'incidence entre le rayon lumineux et l'interface, considérée comme parfaite (sans rugosité), entre les deux milieux. Soient θ_i , θ_r et θ_t les angles que forment avec cette normale les rayons incident, réfléchi et transmis, respectivement (voir figure B.1). Ces lois sont alors définies par les deux égalités suivantes :

$$\theta_r = \theta_i \quad (\text{B.1})$$

$$\eta_1 \sin \theta_i = \eta_2 \sin \theta_t \quad (\text{B.2})$$

De plus, l'intensité lumineuse portée par le rayon incident est répartie entre les rayons réfléchis et transmis. La fraction de cette intensité qui est réfléchi est appelée *coefficient de réflexion*, noté R , et la fraction qui est transmise est appelée *coefficient de transmission*, noté T . Ces coefficients sont définis grâce aux équations de Fresnel, présentées ci-après dans le cas de matériaux *diélectriques*.

Pour une lumière polarisée dans le plan parallèle au plan d'incidence (plan passant par les rayons incident, réfléchi et transmis), les coefficients de Fresnel sont définis par les équations suivantes :

$$R_{\parallel} = \left(\frac{\eta_1 \cos \theta_t - \eta_2 \cos \theta_i}{\eta_1 \cos \theta_t + \eta_2 \cos \theta_i} \right)^2 \quad (\text{B.3})$$

$$T_{\parallel} = 1 - R_{\parallel} \quad (\text{B.4})$$

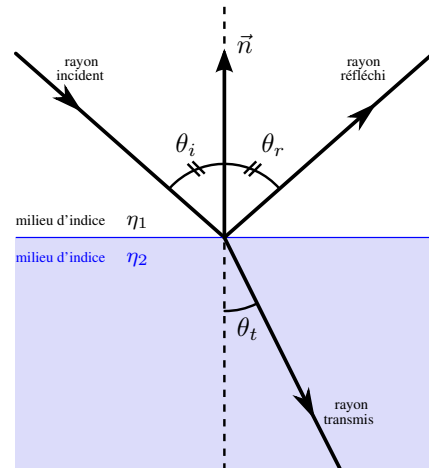


Figure B.1 – Réflexion et réfraction d'un rayon lumineux à l'interface de deux milieux d'indices de réfraction différents.

Dans ce cas d'une lumière polarisée dans le plan perpendiculaire au plan d'incidence, les coefficients de Fresnel sont définis par :

$$R_{\perp} = \left(\frac{\eta_1 \cos \theta_i - \eta_2 \cos \theta_t}{\eta_1 \cos \theta_i + \eta_2 \cos \theta_t} \right)^2 \quad (\text{B.5})$$

$$T_{\perp} = 1 - R_{\perp} \quad (\text{B.6})$$

Dans le cas général d'une lumière non polarisée, les coefficients de réflexion et de transmission sont obtenus à partir des définitions précédentes grâce aux équations suivantes :

$$R = \frac{R_{\perp} + R_{\parallel}}{2} \quad (\text{B.7})$$

$$T = \frac{T_{\perp} + T_{\parallel}}{2} = 1 - R \quad (\text{B.8})$$

On peut d'ailleurs remarquer, à partir de ces équations, que la polarisation n'intervient plus dans la définition des coefficients de Fresnel lorsque la direction d'incidence est orthogonale à l'interface entre les deux milieux :

$$R = R_{\perp} = R_{\parallel} = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2} \right)^2 \quad (\text{B.9})$$

$$T = T_{\perp} = T_{\parallel} = 1 - R = \frac{4\eta_1\eta_2}{(\eta_1 + \eta_2)^2} \quad (\text{B.10})$$

L'égalité $T + R = 1$ que l'on peut observer quelque soit le cas considéré est bien entendu liée au principe de conservation de l'énergie, selon lequel la somme des quantités d'énergie respectivement transmises et réfléchies est égale à la quantité d'énergie lumineuse incidente.

- Harmoniques sphériques -

Mécanisme de compression accéléré

Pour simplifier les écritures dans cette annexe, nous posons $i = l(l + 1) + m$ de telle sorte que $Y_i = Y_l^m$ et $C_i = C_l^m$.

La projection d'une fonction sphérique f dans la base des harmoniques sphériques se fait par convolution de f avec les fonctions harmoniques Y_i , c'est-à-dire en évaluant l'intégrale de l'équation C.1 pour chacune de ces Y_i .

$$C_i = \int_{S^2} f(\theta, \phi) \cdot Y_i(\theta, \phi) d\theta d\phi \quad (\text{C.1})$$

Les coefficients C_i obtenus, appelés *coefficients de Fourier généralisés*, représentent alors les coordonnées de la fonction f dans cette base.

D'un point de vue numérique, cette intégrale peut être approximée à l'aide d'une intégration de Monte-Carlo. Étant données \mathcal{K} variables aléatoires $\{s_k = (\theta_k, \phi_k)\}_{1 \leq k \leq \mathcal{K}}$ définies sur la sphère S^2 ainsi que la *fonction de densité de probabilité* p associée à ces variables, l'intégrale de l'équation C.1 peut être approximée de la manière suivante :

$$C_i \approx \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} \frac{f(s_k) \cdot Y_i(s_k)}{p(s_k)} \quad (\text{C.2})$$

Si p représente une distribution uniforme des variables aléatoires sur la sphère, la probabilité associée à chaque direction s_k est alors de $\frac{1}{4\pi}$ (soit 1 sur l'aire du domaine considéré). L'approximation précédente devient alors :

$$C_i \approx \frac{4\pi}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} f(s_k) \cdot Y_i(s_k) \quad (\text{C.3})$$

Pour que cette approximation soit précise par rapport à l'intégrale d'origine, il est nécessaire que le nombre \mathcal{K} d'échantillons soit très grand (l'approximation devient d'ailleurs une égalité lorsque $\mathcal{K} = \infty$). Cependant, si cette intégrale doit elle-même être évaluée un grand nombre de fois (ce qui est souvent le cas pour les applications en informatique graphique), cette solution devient vite très coûteuse en termes de temps de calcul.

Considérons maintenant que la fonction f soit représentée par un ensemble d'échantillons $\{\bar{f}_j\}_{1 \leq j \leq N}$ associés à des directions connues sur la sphère, et interpolés de manière continue à l'aide d'un ensemble de fonctions $\{b_j\}_{1 \leq j \leq N}$ telles que :

$$f(\theta, \phi) = \sum_{j=1}^N \bar{f}_j \cdot b_j(\theta, \phi) \quad (\text{C.4})$$

Dans ce cas, l'intégrale à évaluer devient :

$$C_i = \int_{S^2} f(\theta, \phi) \cdot Y_i(\theta, \phi) d\theta d\phi \quad (\text{C.5})$$

$$= \sum_{j=1}^N \bar{f}_j \int_{S^2} b_j(\theta, \phi) \cdot Y_i^m(\theta, \phi) d\theta d\phi \quad (\text{C.6})$$

$$= \sum_{j=1}^N \bar{f}_j \cdot b_j^i \quad (\text{C.7})$$

où les valeurs des b_j^i ne dépendent alors plus que de la distribution de l'échantillonnage sur la sphère.

Ainsi, pour toutes les fonctions étant définies à l'aide d'une même distribution d'échantillonnage sur la sphère, les mêmes coefficients b_j^i peuvent être utilisés pour effectuer la projection dans la base des harmoniques sphériques. Cette projection peut alors se résumer à un simple produit matrice vecteur :

$$\underbrace{\begin{bmatrix} C_0 \\ \vdots \\ C_M \end{bmatrix}}_{\mathbf{c}} = \underbrace{\begin{bmatrix} b_0^0 & \dots & b_N^0 \\ \vdots & \ddots & \vdots \\ b_0^M & \dots & b_N^M \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \bar{f}_1 \\ \vdots \\ \bar{f}_N \end{bmatrix}}_{\mathbf{f}} \quad (\text{C.8})$$

où les éléments de la matrice \mathbf{B} peuvent être précalculés une fois pour toute à l'aide d'une méthode classique d'intégration de Monte-Carlo.

Cette méthode a l'avantage de permettre l'utilisation d'échantillonnages sur la sphère dont la distribution n'est pas nécessairement uniforme. Par exemple, il est possible de choisir une subdivision régulière de la sphère selon les coordonnées (θ, ϕ) . Si le mécanisme d'interpolation utilisé pour étendre cet échantillonnage discret à toute la sphère est l'interpolation bilinéaire, chaque fonction d'interpolation b_j correspond alors :

- à la fonction *chapeau* à l'intérieur du quadrangle défini par les huit échantillons voisins de l'échantillon concerné (voir figure C.1),
- à la fonction de valeur nulle partout ailleurs.

Pour une telle subdivision de la sphère, une matrice \mathbf{B} peut être précalculée. Tant que cette même subdivision est considérée, la même matrice \mathbf{B} peut continuer à être utilisée. En revanche, si la subdivision change (résolution en θ ou ϕ modifiée), les fonctions d'interpolation b_j doivent être adaptées (car la dimension et la forme des quadrangles changent) et la matrice \mathbf{B} doit alors être recalculée.

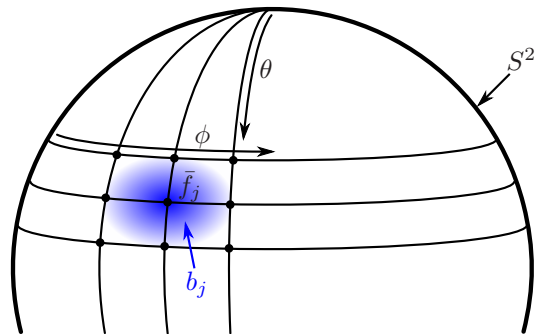


Figure C.1 – Exemple d'échantillonnage (subdivision régulière de la sphère) et de fonction d'interpolation (fonction *chapeau* correspondant à l'interpolation bilinéaire) pouvant être employés par le mécanisme proposé pour la projection dans la base des harmoniques sphériques.

Cet exemple illustre la flexibilité de ce mécanisme : n'importe quelle distribution peut être utilisée, à condition d'être capable de construire un ensemble de fonctions d'interpolation adéquat.

- Liste de publications -

Dans cette annexe sont présentées toutes les publications et les communications scientifiques auxquelles les travaux réalisés durant cette thèse ont donné lieu.

Conférences internationales avec comité de lecture et publication des actes

- Frédéric Larue, Lucas Ammann et Jean-Michel Dischler. "A pipeline for the digitization and the realistic rendering of paintings". *Proceedings of VAST 2007 : the 8th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 71–78, 2007.
- Frédéric Larue et Jean-Michel Dischler. "Automatic registration and calibration for efficient surface light field acquisition". *Proceedings of VAST 2006 : the 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 171–178, 2006.
- Olivier Génévaux, Frédéric Larue et Jean-Michel Dischler. "Interactive refraction on complex static geometry using spherical harmonics". *I3D'06 : Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 145–152, 2006.

Conférences internationales sans comité de lecture

- Jean-Pierre Chambard, Vincent Chalvidan, Mohammed Tazeroualti, Frédéric Larue, Jean-Michel Dischler, Virginie Vurpillot et Anne-Claire Legrand. "Digitization of art pieces based on 3d, colour and texture parameters". *Proceedings of SPIE Symposium on Optical Metrology*, volume 6618 - O3A : Optics for Arts, Architecture, and Archaeology, 2007.

Conférences nationales sans comité de lecture

- Frédéric Larue, Lucas Ammann et Jean-Michel Dischler. "Chaîne de traitement pour la numérisation et le rendu réaliste de peintures d'art". *Journées de l'Association Française d'Informatique Graphique*, pages 73–81, 2007.

- Frédéric Larue et Jean-Michel Dischler. "Recalages et calibrages automatiques pour l'acquisition de champs lumineux surfaciques". *Journées de l'Association Française d'Informatique Graphique*, pages 171–178, 2006.

- Bibliographie -

- [1] D. Akca. Full automatic registration of laser scanner point clouds. In *Proceedings of the 6th Conference on Optical 3-D Measurement Techniques*, volume 1, pages 330–337, 2003.
- [2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987.
- [3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6) :891–923, 1998.
- [4] M. Ashikhmin and P. Shirley. An anisotropic phong brdf model. *Journal of Graphics Tools*, 5(2) :25–32, 2000.
- [5] L. Baboud and X. Décoret. Realistic water volumes in real-time. In *Proceedings of the Eurographics Workshop on Natural Phenomena*, 2006.
- [6] L. Baboud and X. Décoret. Rendering geometry with relief textures. In *Proceedings of the Conference on Graphics Interface*, pages 195–201, 2006.
- [7] B. G. Becker and N. L. Max. Smooth transitions between bump rendering algorithms. In *Proceedings of SIGGRAPH '93*, pages 183–190, 1993.
- [8] R. Benjemaa and F. Schmitt. Fast global registration of 3d sampled surface using a multi-z-buffer technique. In *Image and Vision Computing*, number 17, pages 113–123, 1999.
- [9] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9) :509–517, 1975.
- [10] F. Bernardini, I. M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4) :318–332, 2001.
- [11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4) :349–359, 1999.
- [12] F. Bernardini, H. Rushmeier, I. M. Martin, J. Mittleman, and G. Taubin. Building a digital model of michelangelo's florentine pieta. *IEEE Computer Graphics and Applications*, 22(1) :59–67, 2002.
- [13] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :239–256, 1992.
- [14] G. Blais and M. D. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8) :820–824, 1995.

-
- [15] J. F. Blinn. Simulation of wrinkled surfaces. In *Proceedings of SIGGRAPH '78*, pages 286–292, 1978.
- [16] C.-S. Bouganis and M. Brookes. Multiple light source detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 :509–514, 2004.
- [17] B. J. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics*, 26(3) :21, 2007.
- [18] K. Brunnstrom and A. Stoddart. Genetic algorithms for free-form surface matching. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 4, pages 689–693, 1996.
- [19] M. Callieri, P. Cignoni, F. Ganovelli, G. Impoco, C. Montani, P. Pinci, F. Ponchio, and R. Scopigno. Visualization and 3d data processing in the david restoration. *IEEE Computer Graphics and Applications*, 24(2) :16–21, 2004.
- [20] M. Callieri, P. Cignoni, and R. Scopigno. Reconstructing textured meshes from multiple range rgb maps. In *Proceedings of the Vision, Modeling, and Visualization Conference*, pages 419–426, 2002.
- [21] O. T. Carmichael, D. F. Huber, and M. Hebert. Large data sets and confusing scenes in 3-d surface matching and recognition. In *Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling*, pages 358–367, 1999.
- [22] C. Chao and I. Stamos. Semi-automatic range to range registration : A feature-based method. In *Proceedings of the 5th International Conference on 3-D Digital Imaging and Modeling*, pages 254–261, 2005.
- [23] C. C. Chen and I. Stamos. Range image registration based on circular features. In *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 543–550, 2006.
- [24] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng. A fast automatic method for registration of partially-overlapping range images. In *Proceedings of the IEEE International Conference on Computer Vision*, page 242, 1998.
- [25] S. E. Chen and L. Williams. View interpolation for image synthesis. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 279–288, 1993.
- [26] W.-C. Chen, J.-Y. Bouguet, M. Chu, and R. Grzeszczuk. Light field mapping : efficient representation and hardware rendering of surface light fields. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 447–456, 2002.
- [27] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Proceedings of IEEE International Conference on Robotics and Automation*, 3 :2724–2729, 1991.
- [28] P. Choudhury and J. Tumblin. The trilateral filter for high contrast images and meshes. In *Proceedings of the 14th Eurographics Symposium on Rendering*, pages 186–196, 2003.
- [29] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski. Environment matting extensions : towards higher accuracy and real-time capture. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 121–130, 2000.
- [30] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *Photogrammetric Record*, 16(91) :51–66, 1998.
- [31] J. G. Cleary. Analysis of an algorithm for finding nearest neighbors in euclidean space. *ACM Transactions on Mathematical Software*, 5(2) :183–192, 1979.
- [32] R. L. Cook. Shade trees. In *Proceedings of SIGGRAPH '84*, pages 223–231, 1984.

- [33] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1) :7–24, 1982.
- [34] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.
- [35] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1) :1–34, 1999.
- [36] F. De Piero. Deterministic surface registration at 10hz based on landmark graphs with prediction. In *Proceedings of the British Machine Vision Conference*, 2003.
- [37] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH '00 : Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 145–156, 2000.
- [38] P. E. Debevec, G. Borshukov, and Y. Yu. Efficient view-dependent image-based rendering with projective texture-mapping. In *Proceedings of the 9th Eurographics Rendering Workshop*, 1998.
- [39] M. Dellepiane, M. Corsini, M. Callieri, and R. Scopigno. High quality ptm acquisition : Reflection transformation imaging for large objects. In *The 7th International Symposium on VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 179–186, 2006.
- [40] U. Dhond and J. Aggarwal. Structure from stereo : A review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6) :1489–1510, November 1989.
- [41] C. Dorai, J. Weng, and A. K. Jain. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10) :1131–1138, 1997.
- [42] J.-F. Dufort. Acquisition de brdfs à partir d'images. Technical report, Université de Montréal, 2004.
- [43] M. Erdem, I. Erdem, U. Yilmaz, and V. Atalay. Image-based extraction of material reflectance properties of a 3d rigid object. In *Proceedings of the 12th IEEE Conference on Signal Processing and Communications Applications Conference*, pages 245–248, 2004.
- [44] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2) :99–119, 1996.
- [45] J. Filip and M. Haindl. Non-linear reflectance model for bidirectional texture function synthesis. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 80–83, 2004.
- [46] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics*, 24(3) :544–552, 2005.
- [47] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22(3) :950–953, 2003.
- [48] V. Forest and M. Paulin. Rendu temps réel de fonction de textures bidirectionnelles. *Technique et Science Informatiques*, 26(8) :975–997, 2007.
- [49] T. Franken, M. Dellepiane, F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. Minimizing user intervention in registering 2d images to 3d models. *The Visual Computer*, 21(8-10) :619–628, 2005.
- [50] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3) :209–226, 1977.
- [51] S. F. Frisken and R. N. Perry. Efficient estimation of 3d euclidean distance fields from 2d range images. In *Proceedings of the IEEE Symposium on Volume Visualization and Graphics*, pages 81–88, 2002.

- [52] M. Fuchs, V. Blanz, and H.-P. Seidel. Bayesian relighting. In *Proceedings of the Eurographics Symposium on Rendering Techniques*, pages 157–164, 2005.
- [53] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-d model building. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 581–586, 1994.
- [54] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proceedings of the 3rd Eurographics Symposium on Geometry Processing*, page 197, 2005.
- [55] N. Gelfand, S. Rusinkiewicz, L. Ikemoto, and M. Levoy. Geometrically stable sampling for the icp algorithm. *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling*, page 260, 2003.
- [56] M. Giegl and M. Wimmer. Unpopping : Solving the image-space blend problem for smooth discrete lod transitions. *Computer Graphics Forum*, 26 :46–49, 2007.
- [57] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*, pages 179–186, 2001.
- [58] G. Godin, M. Rioux, and R. Baribeau. Three-dimensional registration using range and intensity information. In *Proceedings of SPIE : Videometrics III*, volume 2350, 1994.
- [59] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [60] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996.
- [61] R. Green. Spherical harmonic lighting : The gritty details. In *Game Developers Conference*, 2003.
- [62] M. Greenspan and G. Godin. A nearest neighbor method for efficient icp. In *Proceedings of the 3rd International Conference on 3-D Digital Imaging and Modeling*, pages 161–168, 2001.
- [63] M. Greenspan and M. Yurick. Approximate k-d tree search for efficient icp. *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling*, pages 442–448, 2003.
- [64] G. Guidi, C. Atzeni, and S. Lazzari. 3d optical scanning diagnostics for Leonardo Da Vinci's "Adorazione dei Magi" conservation. In *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling*, pages 110–115, 2003.
- [65] S. Guy and C. Soler. Graphics gems revisited. *ACM Transactions on Graphics*, 23(3) :231–238, 2004.
- [66] Z. S. Hakura and J. M. Snyder. Realistic reflections and refractions on graphics hardware with hybrid rendering and layered environment maps. In *12th Eurographics Workshop on Rendering*, pages 289–300. Eurographics, 2001.
- [67] K. Hara, K. Nishino, and K. Ikeuchi. Light source position and reflectance estimation from a single view without the distant illumination assumption. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4) :493–505, 2005.
- [68] G. Häusler and S. Karbacher. Reconstruction of smoothed polyhedral surfaces from multiple range images. In *Proceedings of 3D Image Analysis and Synthesis*, pages 191–198, 1997.
- [69] T. Hawkins, J. Cohen, and P. Debevec. A photometric approach to digitizing cultural artifacts. In *Proceedings of the Conference on Virtual reality, Archeology, and Cultural Heritage*, pages 333–342, 2001.

- [70] P. S. Heckbert. Ray tracing jell-o brand gelatin. *Seminal graphics : pioneering efforts that shaped the field*, pages 165–166, 1998.
- [71] P. S. Heckbert and P. Hanrahan. Beam tracing polygonal objects. *Computer Graphics*, 18(3) :119–127, 1984.
- [72] W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. Illuminating micro geometry based on precomputed visibility. In *Proc. of SIGGRAPH '00*, pages 455–464, 2000.
- [73] W. Heidrich, H. Lensch, M. F. Cohen, and H.-P. Seidel. Light field techniques for reflexions and refractions. In *Rendering Techniques '99*, pages 187–196. Eurographics, 1999.
- [74] J. S. Heuberger and W. Pasman. Data fitting report. Technical report, Delft University of Technology, 2005.
- [75] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-d models from unregistered range images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2248–2253, 1994.
- [76] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, 1992.
- [77] B. Horn. Extended gaussian images. In *MIT Artificial Intelligence Laboratory, A.I. Memo No.740*, 1983.
- [78] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4) :629, 1987.
- [79] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7) :1127, 1988.
- [80] P. S. Huang and S. Zhang. 3-d optical measurement using phase shifting based methods. In *Proceedings of the SPIE Two- and Three-Dimensional Methods for Inspection and Metrology III*, volume 6000, pages 15–24, 2005.
- [81] P. S. Huang and S. Zhang. A fast three-step phase-shifting algorithm. In *Proceedings of the SPIE Two- and Three-Dimensional Methods for Inspection and Metrology III*, volume 6000, pages 143–150, 2005.
- [82] D. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7) :637–650, 2003.
- [83] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel. Eikonal rendering : efficient light transport in refractive objects. *ACM Transactions on Graphics*, 26(3) :59, 2007.
- [84] L. Ikemoto, N. Gelfand, and M. Levoy. A hierarchical method for aligning warped meshes. In *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling*, page 434, 2003.
- [85] A. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, page 121, 1997.
- [86] A. E. Johnson and S. B. Kang. Registration and integration of textured 3-d data. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 234–241, May 1997.
- [87] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics*, 22(3) :943–949, 2003.
- [88] J. T. Kajiya. The rendering equation. *ACM SIGGRAPH Computer Graphics*, 20(4) :143–150, 1986.

- [89] S. B. Kang and K. Ikeuchi. The complex egi : A new representation for 3-d pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7) :707–721, 1993.
- [90] J. Kautz and H.-P. Seidel. Towards interactive bump mapping with anisotropic shift-variant brdfs. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, pages 51–58, 2000.
- [91] B. J. King, T. Malisiewicz, C. V. Stewart, and R. J. Radke. Registration of multiple range scans as a location recognition problem : Hypothesis generation, refinement and verification. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 180–187, 2005.
- [92] M. Koudelka, S. Magda, P. Belhumeur, and D. Kriegman. Acquisition, compression and synthesis of bidirectional texture functions. In *Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis*, pages 59–64, 2003.
- [93] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97 : Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 117–126, 1997.
- [94] C. Lahanier. *La Joconde : Essai scientifique*. Codex Images International.
- [95] H. P. A. Lensch, W. Heidrich, and H.-P. Seidel. Automated texture registration and stitching for real world models. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, page 317, 2000.
- [96] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2) :234–257, 2003.
- [97] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.
- [98] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project : 3d scanning of large statues. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144, 2000.
- [99] R. R. Lewis. Making shaders more physically plausible. In *Fourth Eurographics Workshop on Rendering*, pages 47–62, 1993.
- [100] S. Li and K. Mueller. Accelerated, high-quality refraction computations for volume graphics. In *International Workshop on Volume Graphics 2005*, pages 73–81, 2005.
- [101] X. Li and I. Guskov. Multi-scale features for approximate alignment of point-based surfaces. In *Proceedings of the 3rd Eurographics Symposium on Geometry processing*, page 217, 2005.
- [102] X. Li and W. G. Wee. Range image fusion for object reconstruction and modeling. In *Proceedings of the 1st Canadian Conference on Computer and Robot Vision*, pages 306–313, 2004.
- [103] E. Lomonosov, D. Chetverikov, and A. Ekárt. Pre-registration of arbitrarily oriented 3d surfaces using a genetic algorithm. *Pattern Recognition Letters*, 27(11) :1201–1208, 2006.
- [104] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. *SIGGRAPH Computer Graphics*, 21(4) :163–169, 1987.
- [105] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-d rigid transformations. In *Proceedings of the British Conference on Machine Vision*, volume 1, pages 237–246, 1995.

- [106] T. M. MacRobert. *Spherical harmonics : an elementary treatise on harmonic functions with applications*. Pergamon Press, 1967.
- [107] M. Magnor and B. Girod. Adaptive block-based light field coding. In *Proceedings of the International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging*, pages 140–143, 1999.
- [108] M. Magnor and B. Girod. Hierarchical coding of light fields with disparity maps. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3, pages 334–338, 1999.
- [109] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *SIGGRAPH '01 : Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 519–528, 2001.
- [110] Z. Mao, L. Ma, M. Zhao, and X. Xiao. Susan structure preserving filtering for mesh denoising. *The Visual Computer*, 22(4) :276–284, 2006.
- [111] S. Marschner, S. Westin, E. Lafortune, , and K. Torrance. Image-based measurement of the bidirectional reflectance distribution function. *Applied Optics*, 39(16) :2592–2600, 2000.
- [112] T. Mashiko, H. Yagou, D. Wei, Y. Ding, and G. Wu. 3d triangle mesh smoothing via adaptive mmse filtering. In *Proceedings of the 4th International Conference on Computer and Information Technology*, pages 734–740, 2004.
- [113] V. Masselus, P. Dutré, and F. Anrys. The free-form light stage. In *SIGGRAPH '02 : ACM SIGGRAPH 2002 conference abstracts and applications*, page 262, 2002.
- [114] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3d photography using opacity hulls. *ACM Transactions on Graphics*, 21(3) :427–437, 2002.
- [115] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and rendering of transparent and refractive objects. In *Proceedings of the 13th Eurographics Workshop on Rendering*, pages 267–278, 2002.
- [116] D. K. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, pages 79–88, 2002.
- [117] R. Mencl and H. Müller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *DAGSTUHL '97 : Proceedings of the Conference on Scientific Visualization*, page 223, 1997.
- [118] B. Mercier. *Reconstruction et analyse automatiques pour le rééclairage d'objets basés-image*. PhD thesis, Université de Poitiers, Laboratoire SIC, 2006.
- [119] J. Meseth, G. Müller, M. Sattler, and R. Klein. Btf rendering for virtual environments. In *Proceedings of Virtual Concepts*, pages 356–363, 2003.
- [120] J. Meseth, G. Müller, and R. Klein. Preserving realism in real-time rendering of bidirectional texture functions. In *OpenSG Symposium*, pages 89–96, 2003.
- [121] G. S. P. Miller, S. M. Rubin, and D. B. Ponceleon. Lazy decompression of surface light fields for precomputed global illumination. In *Proceedings of the 9th Eurographics Workshop on Rendering Techniques*, pages 281–292, 1998.
- [122] D. Miyazaki, T. Oishi, T. Nishikawa, R. Sagawa, K. Nishino, T. Tomomatsu, Y. Takase, and K. Ikeuchi. The great buddha project : modeling cultural heritage through observation. *Modelling from Reality*, pages 181–193, 2001.

- [123] M. Mudge, T. Malzbender, C. Schroer, and M. Lum. New reflection transformation imaging methods for rock art and multiple-viewpoint display. In *The 7th International Symposium on VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 195–202, 2006.
- [124] G. Müller, J. Meseth, and R. Klein. Compression and real-time rendering of measured btfs using local pca. In *Proceedings of Vision, Modeling and Visualization*, pages 271–280, 2003.
- [125] G. Müller, J. Meseth, and R. Klein. Fast environmental lighting for local-pca encoded btfs. In *Proceedings of the Computer Graphics International*, pages 198–205, 2004.
- [126] S. A. Nene and S. K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9) :989–1003, 1997.
- [127] P. J. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proceedings of the International Conference on Shape Modeling and Applications*, page 130, 1997.
- [128] P. J. Neugebauer. Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *International Journal of Shape Modeling*, 3(1&2) :71–90, 1997.
- [129] P. J. Neugebauer and K. Klein. Texturing 3d models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum*, 18(3) :245–256, 1999.
- [130] K. Nishino and K. Ikeuchi. Robust simultaneous registration of multiple range images. In *Proceedings of the 5th Asian Conference on Computer Vision*, pages 454–461, 2002.
- [131] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-texture method : appearance compression and synthesis based on a 3d model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11) :1257–1265, 2001.
- [132] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached k-d tree search for icp algorithms. In *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling*, pages 419–426, 2007.
- [133] E. Ohbuchi. A real-time refraction renderer for volume objects using a polygon-rendering scheme. In *Computer Graphics International*, pages 190–195, 2003.
- [134] Y. Ohtake, A. Belyaev, and M. Alexa. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In *Proceedings of the 3rd Eurographics Symposium on Geometry Processing*, page 149, 2005.
- [135] Y. Ohtake, A. Belyaev, and H. Seidel. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normal. In *Proceedings of Vision, Modeling, and Visualization*, pages 203–210, 2002.
- [136] M. M. Oliveira, G. Bishop, and D. McAllister. Relief texture mapping. In *Proceedings of SIGGRAPH '00*, pages 359–368, 2000.
- [137] M. Oren and S. K. Nayar. Generalization of lambert's reflectance model. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246, 1994.
- [138] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6) :311–317, 1975.
- [139] P. Pingi, A. Fasano, P. Cignoni, C. Montani, and R. Scopigno. Exploiting the scanning sequence for automatic registration of large sets of range maps. *Computer Graphics Forum*, 24(3) :517–526, 2005.
- [140] R. Pito. Mesh integration based on co-measurements. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 397–400, 1996.

- [141] F. Policarpo and M. M. Oliveira. Relief mapping of non-height-field surface details. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pages 55–62, 2006.
- [142] F. Policarpo and M. M. Oliveira. Relaxed cone stepping for relief mapping. In H. Nguyen, editor, *GPU Gems 3*, pages 409–428. Addison-Wesley Professional, 2007.
- [143] F. Policarpo, M. M. Oliveira, and J. L. D. Comba. Real-time relief mapping on arbitrary polygonal surfaces. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pages 155–162, 2005.
- [144] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C (2nd ed.) : the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
- [145] K. Pulli. Multiview registration for large data sets. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.
- [146] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, and W. Stuetzle. Robust meshes from multiple range maps. In *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 205–211, 1997.
- [147] M. K. Reed, P. K. Allen, and I. Stamos. 3-d modeling from range imagery : an incremental method with a planning component. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, page 76, 1997.
- [148] C. Robertson and R. B. Fisher. Parallel evolutionary registration of range data. *Computer Vision and Image Understanding*, 87(1-3) :39–50, 2002.
- [149] M. A. Robertson, S. Borman, and R. L. Stevenson. Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging*, 12(2) :219–285, 2003.
- [150] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Multiple textures stitching and blending on 3d objects. In *Proceedings of the Eurographics Workshop on Rendering*, pages 127–138, 1999.
- [151] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Acquiring, stitching and blending diffuse appearance attributes on 3d models. *The Visual Computer*, 18(3) :186–204, 2002.
- [152] D. Rodgman and M. Chen. Refraction in discrete ray tracing. In *Proceedings of Volume Graphics*, pages 3–17, 2001.
- [153] G. Roth. Registering two overlapping range images. *Proceedings of the 2nd International Conference on 3-D Digital Imaging and Modeling*, pages 191–200, 1999.
- [154] H. Rushmeier and F. Bernardini. Computing consistent normals and colors from photometric data. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 99–108, 1999.
- [155] H. Rushmeier, F. Bernardini, J. Mittleman, and G. Taubin. Acquiring input for rendering at appropriate levels of detail : Digitizing a Pietà. In *Proceedings of the 9th Eurographics Rendering Workshop*, pages 81–92, 1998.
- [156] H. E. Rushmeier, G. Taubin, and A. Guézic. Applying shape from lighting variation to bump map capture. In *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 35–44, 1997.
- [157] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics*, 21(3) :438–446, 2002.
- [158] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*, 2001.

-
- [159] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 573–580, 1994.
- [160] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross. A practical structured light acquisition system for point-based geometry and texture. In *Proceedings of the Eurographics Symposium on Point-Based Graphics*, pages 89–98, 2005.
- [161] R. Sagawa, K. Nishino, and K. Ikeuchi. Robust and adaptive integration of multiple range images with photometric attributes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 172–179, 2001.
- [162] A. D. Sappa and M. A. Garcia. Incremental multiview integration of range images. In *Proceedings of the International Conference on Pattern Recognition*, volume 01, page 1546, 2000.
- [163] A. D. Sappa, A. Restrepo-Specht, and M. Devy. Range image registration by using an edge-based representation. In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems*, pages 167–176, 2001.
- [164] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. *Computer Graphics*, 31(Annual Conference Series) :379–387, 1997.
- [165] S. Seipel and A. Nivfors. Efficient rendering of multiple refractions and reflections in natural objects. In *Proceedings of the Annual SIGRAD Conference*, pages 1–6, 2006.
- [166] G. C. Sharp, S. W. Lee, and D. K. Wehe. Icp registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :90–102, 2002.
- [167] L. Silva, O. R. Bellon, and K. L. Boyer. Enhanced, robust genetic algorithms for multiview range image registration. In *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling*, page 268, 2003.
- [168] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-d pose estimation using a high-speed range sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2235–2241, 1994.
- [169] P.-P. Sloan. Stupid spherical harmonics (sh) tricks. In *Game Developers Conference*, 2008.
- [170] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3) :527–536, 2002.
- [171] P.-P. Sloan, B. Luna, and J. Snyder. Local, deformable precomputed radiance transfer. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers*, pages 1216–1224, 2005.
- [172] M. Soucy and D. Laurendeau. Multi-resolution surface modeling from multiple range views. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 348–353, 1992.
- [173] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4) :344–358, 1995.
- [174] I. Stamos and P. K. Allen. 3-d model construction using range and image data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 531–536, 2000.
- [175] I. Stamos and P. K. Allen. Integration of range and image sensing for photorealistic 3d modeling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1435–1440, 2000.
- [176] I. Stamos and M. Leordeanu. Automated feature-based range registration of urban scenes of large scale. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 555–561, 2003.

- [177] Y. Sun and M. A. Abidi. Surface matching by 3d point's fingerprint. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 263–269, 2001.
- [178] Y. Sun, C. Dumont, and M. Abidi. Mesh-based integration of range and color images. In *Sensor Fusion : Architectures, Algorithms, and Applications*, pages 24–28, 2000.
- [179] Y. Sun, J. Paik, A. Koschan, and M. Abidi. Surface modeling using multi-view range and color images. *Integrated Computer-Aided Engineering*, 10(1) :37–50, 2003.
- [180] J.-P. Tarel and N. Boujemaa. A coarse to fine 3d registration method based on robust fuzzy clustering. *Computer Vision and Image Understanding*, 73(1) :14–28, 1999.
- [181] N. Tatarchuk. Dynamic parallax occlusion mapping with approximate soft shadows. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pages 63–69, 2006.
- [182] G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of the 5th International Conference on Computer Vision*, page 852, 1995.
- [183] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995.
- [184] G. Taubin. Linear anisotropic mesh filtering. Research Report RC2213, IBM, 2001.
- [185] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98 : Proceedings of the Sixth International Conference on Computer Vision*, page 839, 1998.
- [186] S. Tominaga, N. Tanaka, and T. Komada. Imaging and rendering of oil paintings using a multi-band camera. In *Southwest Symposium on Image Analysis and Interpretation*, pages 6–10, 2004.
- [187] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. pages 221–244, 1992.
- [188] P. Y. Ts'o and B. A. Barsky. Modeling and rendering waves : Wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics*, 6(3) :191–214, 1987.
- [189] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH '94 : Proceedings of the 21st annual Conference on Computer Graphics and Interactive Techniques*, pages 311–318, 1994.
- [190] J. Vollmer, R. Mencl, and H. Müller. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, 18(3) :131–138, 1999.
- [191] M. W. Walker, L. Shao, and R. A. Volz. Estimating 3-d location parameters using dual number quaternions. *CVGIP : Image Understanding*, 54(3) :358–367, 1991.
- [192] G. J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Computer Graphics*, 26(2) :265–272, 1992.
- [193] S. Weik. Registration of 3-d partial surface models using luminance and depth information. In *Proceedings of the International Conference in 3-D Digital Imaging and Modeling*, page 93, 1997.
- [194] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report TR95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.
- [195] T. Werner, R. D. Hersch, and V. Hlavac. Rendering real-world objects using view interpolation. In *Proceedings of the 5th International Conference on Computer Vision*, page 957, 1995.
- [196] S. H. Westin, J. R. Arvo, and K. E. Torrance. Predicting reflectance functions from complex surfaces. *Computer Graphics*, 26(2) :255–264, 1992.

- [197] M. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. In *Proceedings of the International Conference on Computer Vision*, pages 917–924, 1998.
- [198] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6) :343–349, June 1980.
- [199] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296, 2000.
- [200] C. Wyman. An approximate image-space approach for interactive refraction. *ACM Transactions on Graphics*, 24(3) :1050–1053, 2005.
- [201] J. Wyngaerd and L. Van Gool. Combining texture and shape for automatic crude patch registration. In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pages 179–186, 2003.
- [202] J. V. Wyngaerd, L. J. V. Gool, R. Koch, and M. Proesmans. Invariant-based registration of surface patches. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 301–306, 1999.
- [203] H. Yagou, Y. Ohtake, and A. Belyaev. Mesh smoothing via mean and median filtering applied to face normals. In *Proceedings of the Geometric Modeling and Processing — Theory and Applications*, page 124, 2002.
- [204] S. M. Yamany, M. T. Ahmed, E. E. Hemayed, and A. A. Farag. Novel surface registration using the grid closest point (gcp) transform. In *Proceedings of the IEEE International Conference on Image Processing*, pages 809–813, 1998.
- [205] S. M. Yamany and A. A. Farag. Free-form surface registration using surface signatures. In *Proceedings of the International Conference on Computer Vision*, volume 2, page 1098, 1999.
- [206] R. Yang and P. K. Allen. Registering, integrating and building cad models from range data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3115–3120, 1998.
- [207] L. Zagorchev and A. Goshtasby. A mechanism for range image integration without image registration. In *Proceedings of the 5th International Conference on 3-D Digital Imaging and Modeling*, pages 126–133, 2005.
- [208] A. Zhang, W. Sun, S. Hu, and C. Qian. Automatic global registration of multiple 3d data sets from outdoor urban environments based on feature units. In *Proceedings of the 20th Spring Conference on Computer Graphics*, pages 193–199, 2004.
- [209] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 1999.
- [210] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Proceedings of the 7th IEEE International Conference on Computer Vision*, 1 :666–673, 1999.
- [211] H. Zhou, Y. Liu, and L. Li. Incremental mesh-based integration of registered range images : Robust to registration error and scanning noise. In *Proceedings of the 7th Asian Conference on Computer Vision*, volume 3851, pages 958–968, 2006.
- [212] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proceedings of ACM SIGGRAPH 99*, pages 205–214. ACM, 1999.