Université de Strasbourg

# UNIVERSITÉ DE STRASBOURG

msii ÉCOLE DOCTORALE

## *ÉCOLE DOCTORALE MSII 269*

*Mathématiques, Sciences de l'Information et de l'Ingénieur*

Laboratoire des sciences de l'Ingénieur, de l'Informatique et de l'Imagerie

ICube UMR 7357

**Thèse** présentée par :

**Boyang Yu**

soutenue le : 10/12 **en 2024**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : **Informatique**

# High-quality recovery of garment models and sewing patterns from 3D clothed human data

THÈSE dirigée par :

| | |
|---|---|
| **Hyewon Seo** | Directrice de recherche, Université de Strasbourg |
| **Frédéric Cordier** | Co-encadrant, Maître de conférences, Université de Haute Alsace |

MEMBRES DU JURY :

Rapporteurs :

| | |
|---|---|
| **Damien Rohmer** | Professeur des Universités, Ecole Polytechnique |
| **Jean-Sebastien Franco** | Directeur de Recherche, Inria Grenoble |

Examinateurs :

| | |
|---|---|
| **Liming Chen** | Professeur des Universités, Ecole Centrale de Lyon |
| **Miguel Otaduy** | Professeur des Universités, Université Rey Juan Carlos |

Invités:

| | |
|---|---|
| **Christiane Luible-Bär** | Professeur des Universités, Université d'Art et de Design Linz |

Université de Strasbourg

Boyang Yu

iCUBE

# High-quality recovery of garment models and sewing patterns from 3D clothed human data

## Résumé

La capacité de reconstruire des modèles de vêtements de haute qualité à partir des formes 3D de personnes vêtues peut améliorer l'interprétabilité de la géométrie capturée des vêtements réels, ainsi que leur reproduction fidèle dans le monde numérique. Cela aura notamment un impact sur des domaines tels que la capture de formes en réalité virtuelle sociale et les essayages virtuels dans l'industrie de la mode.

Cette thèse aborde le problème de la reconstruction de la géométrie d'un vêtement cible arbitraire. Nous proposons une approche pour estimer une réplique animable ainsi que son patron 2D correspondant. Basée sur un simulateur de vêtements différentiable, notre méthode exécute un processus d'optimisation visant à minimiser l'écart entre la forme simulée du vêtement et la géométrie cible, tout en conservant des propriétés souhaitables telles que la symétrie gauche-droite. En particulier, nous implémentons une chaîne de conception inverse de vêtement pour s'aligner avec le processus de modélisation de vêtements standardisé par l'industrie de la mode et les logiciels de simulation. Ce système est conçu pour reconstruire la géométrie du vêtement en modifiant les patrons 2D et en ajustant les propriétés des matériaux, ce qui permet ensuite de placer le vêtement autour du modèle corporel du porteur et de le réanimer par simulation.

Parallèlement à cette proposition, nous présentons également une nouvelle méthode basée sur une optimisation de déformation, qui ajuste la géométrie du maillage source pour correspondre aux détails fins du modèle 3D cible. Nous montrons que cette méthode est particulièrement efficace pour raffiner la géométrie du modèle de vêtement afin de reproduire des détails fins et de haute fréquence, améliorant ainsi la qualité de l'ajustement et soutenant des applications telles que l'enregistrement non rigide.

Les résultats expérimentaux sur diverses données réelles et synthétiques montrent que nos méthodes surpassent les méthodes de pointe en produisant des modèles de vêtements de haute qualité et des patrons 2D précis.

**Mots clés :** Méthode des éléments finis, Modélisation inverse des vêtements, Simulation différentiable, Déformation du maillage.

# Abstract

The capability to recover high-quality garment models from 3D shapes of clothed people can significantly enhance the interpretability of captured geometry of real garments, as well as their faithful reproduction in the digital world. This will notably impact fields like shape capture in social VR, and virtual try-ons in the fashion industry.

This thesis addresses the problem of arbitrary target cloth geometry recovery. We propose an approach to estimate its animatable replica along with its corresponding 2D pattern. Built upon a differentiable cloth simulator, it runs an optimization process that is directed towards minimizing the deviation of the simulated garment shape from the target geometry, while maintaining desirable properties such as left-to-right symmetry. In particular, we are implementing an inverse garment design pipeline to align with the garment modeling process standardized by the fashion industry and cloth simulation software, it is designed to recover the garment geometry by altering the 2D patterns and adjusting the material properties, which can be further placed around the wearer's body model and reanimated via simulation.

Along with this proposition, we also present a new method based on a deformation optimization that deforms the source mesh geometry to match the fine-grain details of the 3D target. We show that it is adept at updating garment geometry to produce fine, high-frequency details, improving fit quality and supporting applications like non-rigid registration.

Experimental results on various real-world and synthetic data show that our methods outperform state-of-the-art methods in producing both high-quality garment models and accurate 2D patterns.



**Key words :** Finite element method, Inverse garment modeling, Differentiable simulation, Physics-aware mesh deformation.

# Acknowledgments

I would like to express my thanks to my thesis supervisor, Hyewon Seo, who has given me the chance to undertake this thesis project. My heartfelt gratitude to her and my thesis co-advisor Frédéric Cordier for their guidance, time, and mentorship throughout the journey.

I would also like to extend my gratitude to my thesis committee members: Damien Rohmer, Jean-Sebastien Franco, Liming Chen, Miguel Otaduy and Christiane Luible-Bär. Special thanks to the reporters Damien Rohmer and Jean-Sebastien Franco, having the manuscript read and reported by two renowned experts is an honor.

Big thanks to all the mates in the lab. Especially, I'd like to thank Pierre Galmiche and Ziqiu Zeng for integrating me so well into the team on my arrival, and for their availability during the rest of my time here. Thanks to Kaifeng Zou, my thesis fellow and friend, for all the good times, Pupu, and collaboration on the facial expression synthesis projects.

I'd like to thank my wife Qianyu Guo, for her immeasurable support over the last few years. Thanks to my lovely parents and sister, my wonderful parents-in-law, and my gracious grandparents for their unconditional love all the time.

Finally, I'd like to thank the teachers who enlightened me and motivated me to pursue a PhD: Valérie Burdin, Dominique Pastor, Jiujiang Liu, Lifeng Zhou.

# CONTENTS

# NOTATIONS

## Acronyms

| | |
|---|---|
| CAD | Computer Aided Design |
| PBS | Physics-Based Simulation |
| FEM | Finite Element Method |
| MVC | Mean Value Coordinates |
| LBS | linear Blend Skinning |
| CF | Chamfer Distance |
| DOF | Degree Of Freedom |

# Notations

| | |
|---|---|
| $t$ | time variable |
| $M$ | mass matrix |
| $f$ | force variable |
| $J = \frac{\partial f}{\partial x}$ | Jacobian of force |
| $U$ | sewing pattern |
| $C$ | control points |
| $K$ | effective points |
| $B$ | body model |
| $X = \{x_i\}$ | garment mesh vertices |
| $T = \{t_i\}$ | target mesh vertices |
| $F$ | garment mesh faces |
| $\boldsymbol{H}$ | stretching stiffness |
| $k$ | bending stiffness |
| $\phi$ | deformation mapping |
| $\mathbf{F}$ | deformation gradient |
| $\Psi$ | strain energy density |
| $\mathbf{G}$ | green strain |
| $A$ | area of triangle |
| $e$ | edge |
| $\alpha$ | dihedral angle |
| $\mathbf{n}$ | normal vector |
| $\mathbf{J}$ | Jacobian at triangle |
| $sdf(\cdot)$ | signed distance function |
| $W(\cdot)$ | skinning function |

CHAPTER 1

# INTRODUCTION

## 1.1 General context

Clothing has been an integral part of human life throughout history. Initially serving as a protective barrier for the skin and a means to retain warmth, it has evolved from the modern textile and apparel industry to contemporary virtual clothing for digital avatars. The concept of clothing continues to expand with advances in technology. In recent years, rapid developments in virtual reality (VR), augmented reality (AR), and telepresence have spurred a growing interest in the development of virtual clothing that ensures plausibility, interactivity, and efficient rendering. Additionally, the increasing availability of human clothing scans could facilitate the creation of digital replicas from the real world, which underscores the pressing need of recovering realistic virtual outfits for personalized avatars.

While automation and digitally integrated processes widely exist in the development, computer-aided design (CAD), and digital simulation of apparel to assist in production, they form a workflow which we refer to as *virtual* to *real*. There is a renewed interest in developing virtual alternatives inversely from real-world inputs, *real* to *virtual*. However, this requires the recovery of high-quality clothing assets which presents a substantial challenge. In essence, a *garment recovery system* needs to replicate the shape and appearance of real garments with digital 3D models. For such a system to be successful, it is imperative that it makes accurate reconstruction, provides reusable results, and generalizes well to a large scale of garments. To satisfy these requirements, the goal of this thesis is to develop new automated solutions for recovering digital garments up to the need of virtual avatar creation.

In the following sections, we introduce the steps involved in virtual clothing recovery, address the associated technical challenges, and outline the contributions of this thesis in mitigating these challenges.

### 1.1.1 Virtual clothing and simulation

Cloth modeling has been studied in the textile mechanics community for a long time. Later in the mid-80s, it gradually came into the view of computer graphics researchers. From the modeling of basic shapes to the modeling of complex physical behaviors, the visual realism of virtual clothing continues to improve, enabling animation and visualization in the generated images and films.

The most critical aspect of this process is faithfully reflecting the nature of the real fabrics.

Fabrics are the main material used in the manufacture of garments and textiles. A multitude of fibers are combined to form yarn, and thousands of interlaced yarn strands are woven together to create the fabric. This intricate assembly process results in complex mechanical properties observed in garments. Generally, fabric deformations can be categorized into four types: tensile, shearing, bending, and twisting. The complex mechanical properties of fabrics have motivated researchers to develop many different models aimed at approximating the behavior of clothing. In the clothing industry, in addition to the material of fabrics, the size and style of a garment are also the primary factors. Proper size ensures comfort by conforming to the body's dimensions and enhances the wearer's appearance by complementing their natural shape. Style, which is subjective, allows individuals to express their personality, tastes, and preferences, thereby making a personal statement on various occasions. The interplay between size and style significantly contributes to the diversity of garments, providing a wide range of options to meet various needs. The digitalization of all these factors promotes related research and downstream applications in the fashion industry, including virtual try-ons and software for garment design (e.g., CLO Virtual Fashion, Marvelous Designer, Style3D).

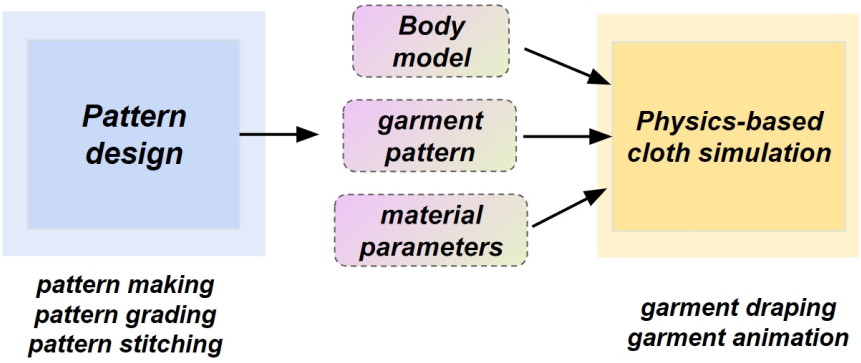


**Figure 1.1:** The pattern design and cloth simulation pipeline.

Since the early 1990s, a conventional approach to constructing virtual garments has been to replicate the real-life process of traditional dressmaking. A two-dimensional sewing pattern is created, which is then virtually stitched together to form a three-dimensional garment. This process mirrors how physical clothing is made, offering a familiar and effective framework for generating digital clothing models.

In general, such a system can be divided into two components: 2D garment pattern design and 3D garment simulation, as depicted in Figure 1.1. The first component involves the creation of garment patterns, which is followed by a 3D garment simulator used for deforming. Within the 3D simulation, essential steps include mechanics modeling, dynamics solving, and collision handling. User specifications are required to bridge the gap

between 2D and 3D, involving tasks such as pattern placement, seaming, and garment assembly.

In practice, a pattern consists of several pieces of fabric, i.e. panels. Each panel's geometric representation is discretized into a triangular thin-shell mesh. The planar panels should be placed around the body, then attached and seamed along designated lines as shown in Figure 1.2. A mechanical simulation is conducted to achieve the final 3D shape, which is influenced by the underlying body shape, material properties, and the collision and contact with the body surface. The final visual appearance of the garment is rendered based on parameters such as color, shininess, and texture.


**Figure 1.2:** Sewing pattern samples are placed and draped over a body to get the corresponding 3D garment model, drawn from Korosteleva et Lee (2021).

## 1.2 Problematics

The challenge of achieving high-quality garment recovery is twofold. The first involves 3D cloth reconstruction, which aims to generate virtual garments that accurately replicate the target geometry. The second pertains to 2D garment pattern estimation, which plays a crucial role, as the pattern directly affects the garment's fit and the formation of wrinkles on the body. Both are essential to the visual realism and physical accuracy of virtual garments.

### 1.2.1 3D cloth reconstruction

With the increasing popularity of commercial 3D body scan solutions, such as vitronic, botspot, eisko; the acquisition of 3D clothed human data is becoming a common alternative to traditional 2D images. Given the challenging variations in clothing, the detailed and robust reconstruction of the human shape as well as the dressed garments, is still widely studied in the computer graphics and computer vision communities. In particular, reconstructed garments can potentially be used to enable further cloth retargeting or animation applications.

Given 3D scans, methods like Ma et al. (2020); Saito et al. (2021) output registered single-layer clothing models for high-quality 3D scans obtained with high-resolution body scanners. The main limitation of these approaches is that a single representation is output to model both the human body and clothing, which hinders the digital manipulation of the captured garment part. To reconstruct garments separately as a layer on top of the human body, Chen et al. (2021); Pons-Moll et al. (2017) segment clothing from the registered clothed human meshes by solving the per-vertex clothing label for each vertex. Tiwari et al. (2020) maps a registered single mesh (to the scan) to multi-layer meshes, and predicts the separated clothing draping on the body as a function of size. Lahner et al. (2018) enables posing a template garment for reconstructing global shape deformations, while also learning to regress pose-dependent high-frequency wrinkles.

On the other hand, inferring detailed geometries and body shapes from images/videos is still a popular research topic, as they are the most common and accessible. Learning-based techniques have achieved significant progress. Zheng et al. (2019) refines a coarse-scale volumetric reconstruction with a normal refinement network to add high-frequency details. Saito et al. (2019) proposes a pixel-aligned implicit surface representation to regress detailed clothed humans from images. Habermann et al. (2020) disentangles the capture into an articulated skeleton estimation and a non-rigid surface deformation; its following work Li et al. (2021) separates the cloth part and further incorporates an on-the-fly simulation supervision during training. However, these image-based approaches still have scale ambiguity and inferior human shape accuracy due to the lack of one dimension. To alleviate the lack of dimension, multi-view camera setups are used for learning to reconstruct textured garment meshes as in Xiang et al. (2022, 2021). Especially, a physical simulator is used in Xiang et al. (2022) to generate clothing geometry with natural and rich dynamics.

The aforementioned advancements in computer vision and graphics have significantly

accelerated the automation of clothing asset creation from images or scans. However, enforcing physical constraints in 3D geometry reconstruction remains a less explored problem. Although studies such as Li et al. (2021); Xiang et al. (2022) integrate the cloth simulation into their pipeline, the simulator operates as a black box, of which the internal workings, including how physical parameters are applied and adjusted, are not exposed or easily accessible for modification or optimization.

### 1.2.2 2D garment sewing pattern estimation

Garment sewing pattern represents a garment's intrinsic rest shape, independent of extrinsic factors such as forces, collisions, and fabric properties. Additionally, it is parameterizable, enabling direct and interpretable control over the garment design. The recovery of garment sewing patterns remains a challenging problem, as the clothing may conform to any body shape in arbitrary poses, introducing significant complexities in the inverse problem.

Existing works on sewing pattern reconstruction mostly rely on 3D inputs. Bang et al. (2021); Goto et Umetani (2021); Pietroni et al. (2022) cut the input mesh into developable surface patches and unfold them onto 2D planes. Korosteleva et Lee (2022); Nag et al. (2023) train deep learning architecture to predict sewing patterns based on point cloud inputs. Image-based sewing pattern estimation is also a commonly studied technique. Yang et al. (2016) infers sewing patterns by estimating panel parameters with iterative optimization. Chen et al. (2024); Liu et al. (2023) train deep neural networks to extract the panels of the sewing pattern and predict the stitching information. A short survey is provided in section 2.5.

State-of-the-art pattern estimation methods such as Korosteleva et Lee (2022); Liu et al. (2023) leverage extensive literature in deep learning and computer vision to generate patterns, but they often neglect material properties.

## 1.3 Contributions

The combination of visual aesthetics and physical accuracy is essential in achieving convincing virtual clothing. While physics-based simulation and editing tools can generate high-quality, realistic garment models, creating a personalized virtual outfit using them for each user remains labor-intensive. This process involves manual design and calibration by skilled artists, requiring specialized software and expertise. The primary aim of

this manuscript is to propose novel automated solutions to streamline this process, facilitating the creation of customized avatars in clothing for both novices and experts.

Rather than solely manipulating the 3D clothing mesh geometry or predicting the sewing pattern, we first aim to bridge this gap by employing differentiable simulation. By integrating physical simulation within the optimization loop, we ensure that the dynamics of the clothing are accounted for in the optimization process. This approach allows for the adjustment of 2D sewing patterns in conjunction with the fitting of draped 3D models, facilitating the accurate physical replication of these garments. By extending novel breakthroughs in differentiable simulation, we optimize assets in a physically plausible manner. This enables the recovery of 3D garment geometry and 2D sewing patterns, as well as the retrieval of material properties, using only a minimal template garment library. Consequently, our optimization process outputs comprehensive assets suitable for further simulation and various other downstream applications.

In addition to utilizing differentiable simulation, and inspired by the trend of self-supervised neural simulation, we subsequently explored garment reconstruction via mesh deformation, incorporating physical constraints into the cost functions. This approach focuses on synthesizing 3D garments from a base mesh template, driven by a single 3D target shape. It effectively updates the garment's geometry to capture fine, high-frequency details, such as dynamic folds and wrinkles. The resulting geometries can benefit many applications such as telepresence, and virtual try-ons.

In summary, our contributions are as follows:

- A new formulation of inverse garment draping based on the differentiable physics-based simulation (PBS). At the heart of our method, three contributions enable us to recover 3D draped garments geometry with their corresponding 2D sewing patterns and material parameters from clothed human data. First, an upgraded differentiable simulator aimed at improving computation speed, operating seamlessly within an end-to-end optimization process. Second, a compact parameterization of the sewing pattern to streamline the optimization, ensuring the preservation of desirable pattern properties. Third, a set of carefully designed loss terms which enable high quality, design-preserving results on a variety of clothes ranging from tight-fitting to loose styles.

- A novel two-phase approach for deforming a template mesh to achieve high-quality fitting. This method operates in a coarse-to-fine manner, initially capturing broader dimensions (e.g., length, size). Subsequently, a mesh deformation optimization

process refines the result to align with the fine-grain details of the 3D target. Geometry supervision is introduced directly in 3D space by updating Jacobians to promote smooth, large-scale deformations while minimizing localized noise artifacts during gradient-based optimization. Additionally, a carefully constructed combination of loss functions enables the base mesh to deform freely toward the target. Notably, physics constraints, previously handled on a frame-by-frame basis via implicit integrators, are now incorporated as regularization loss terms in the optimization.

The works during the thesis:

- "PhyDeformer: High-quality Non-Rigid Garment Registration with Physics-awareness", Preprint.
- "Inverse Garment and Pattern Modeling with a Differentiable Simulator", B. Yu, F. Cordier and H. Seo, published at Computer Graphics Forum. Volume 43, Issue 7 (in Pacific Graphics 2024 conference proceedings).
- "4D Facial Expression Diffusion Model", K. Zou, S. Faisan, B. Yu, S. Valette, H. Seo, accepted for publication at ACM Transactions on Multimedia Computing, Communications, and Applications, published online: https://dl.acm.org/doi/10.1145/3653455.
- "3D Facial Expression Generator Based on Transformer VAE", K. Zou, B. Yu and H. Seo, accepted at 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 2023, pp. 2550-2554.

CHAPTER 2

# BACKGROUND

## 2.1 Introduction

Virtual clothing has garnered significant interest from both computer graphics and computer vision communities and empowered a diverse range of notable studies that form the basis of this thesis. This chapter provides a structured overview of the related works, together with the preliminaries, organized as follows:

- Section 2.2, Section 2.3 review respectively the literature on cloth simulation and learning base cloth deformation. We also discuss the existing methods of garment reconstruction in Section 2.4.
- Section 2.5 reviews the literature of pattern representation and estimation. We discuss the existing methods as well as their advantages and limitations.
- Section 2.6 briefly reviews the literature on SMPL human body modeling and estimation of accurate bodies of real images or scans.
- Section 2.7 outlines the concepts and notations for the continuum mechanics and finite element discretizations, the concise mathematical descriptions will serve for the coming chapters.

## 2.2 Physics-based cloth simulation

Cloth simulation has been and remains one focus of computer graphics and animation. Since the practice pioneered by Baraff and Witkin Baraff et Witkin (1998), modern cloth animation workflows commonly adopt an implicit time integration scheme. At the heart of a cloth simulation system, it's a mechanical model that deforms the cloth surfaces so that they behave like real fabrics. Considerable research has focused on refining the material modeling to accurately emulate cloth's mechanical behaviors. Mass-spring systems Provot et al. (1995) and more general particle systems Eberhardt et al. (1996); Volino et al. (1995) used to be very popular as simulation techniques in computer graphics, because of the ease of implementation and low computational cost. However, the accuracy offered by a mass-spring method is rather limited. For more authentic material behavior, approaches based on continuum mechanics have been used. Finite element method (FEM) is one powerful method Narain et al. (2012); Volino et al. (2009), which computes mechanical energies within a predefined spatial discretization. As a transition between mass-spring systems and finite elements, Volino et Magnenat-Thalmann (2005) proposed an

accurate particle system that draws on notions from continuum mechanics but replaces the numerical discretization with a direct geometric formulation. Simulating garments at the yarn level instead of per elementary continuum, although computationally intensive, was also explored in Cirio et al. (2014, 2015), by condensing the dynamics of individual yarn strands.

To further increase efficiency while providing favorable trade-offs between stability, speed, and accuracy, many improvements have been proposed. Position Based Dynamics (PBD) Müller et al. (2007) updates positions directly using projection functions resulting in high-performance simulations, but the stiffness behavior is subject to non-physical factors such as the number of iterations and the mesh resolution. eXtended Position Based Dynamics (XPBD) Macklin et al. (2016) overcome the limitation of iteration dependent behavior of PBD. Unlike PBD updates positions directly to project constraints, projective dynamics Sofien et al. (2014) interconnects the FEM methods and PBD methods, it uses projection to define quadratic energy, leading to an efficient and accurate solver with constant hessian.

Given the reduced stretchability of many fabrics, strain limiting is extensively used to prevent instability and artifacts due to large deformation, such as Thomaszewski et al. (2009); Wang et al. (2010). Cloth bending is closely tied to the parameterization of the dihedral angle Volino et al. (1995). Bridson et al. (2005) developed the bending mode of a hinge-based element orthogonal to in-plane deformations. Later, a quadratic bending model was proposed in Bergou et al. (2006) with assumptions of the planar case and little stretching(inextensible), with discrete mean curvature approximating bending.

With an established material model, efforts have been made in Miguel et al. (2012); Wang et al. (2011) to incorporate captured real-world data into cloth models, rather than relying on heuristic parameters that only ensure visual plausibility. They measure the deformation of small pieces of fabrics under controlled setups, and solve for simulation material parameters such as stretching, shearing, and bending, to replicate the real samples in simulators.

### 2.2.1 Differentiable cloth simulation

In the last few years, the rapid growth in deep learning frameworks has stimulated the development of differentiable cloth simulation methods. A fully functional differentiable simulation enables the calculation of gradients for parameters that affect the outcome of simulation results. Subsequently, gradient-based optimization algorithms can be applied

for inverse design and system identification purposes. Liang et al. (2019) presents the first differentiable cloth simulation within the FEM framework(Section 2.7). Several other simulation frameworks are made differentiable such as DiffPD Du et al. (2021) and DiffXPBD Stuyck et Chen (2023). While DiffPD is extended by Li et al. (2022b) with dry frictional contact for clothing, DiffXPBD has paved the way for numerous applications such as material parameters estimation Larionov et al. (2022), avatar recovery Li et al. (2024c).

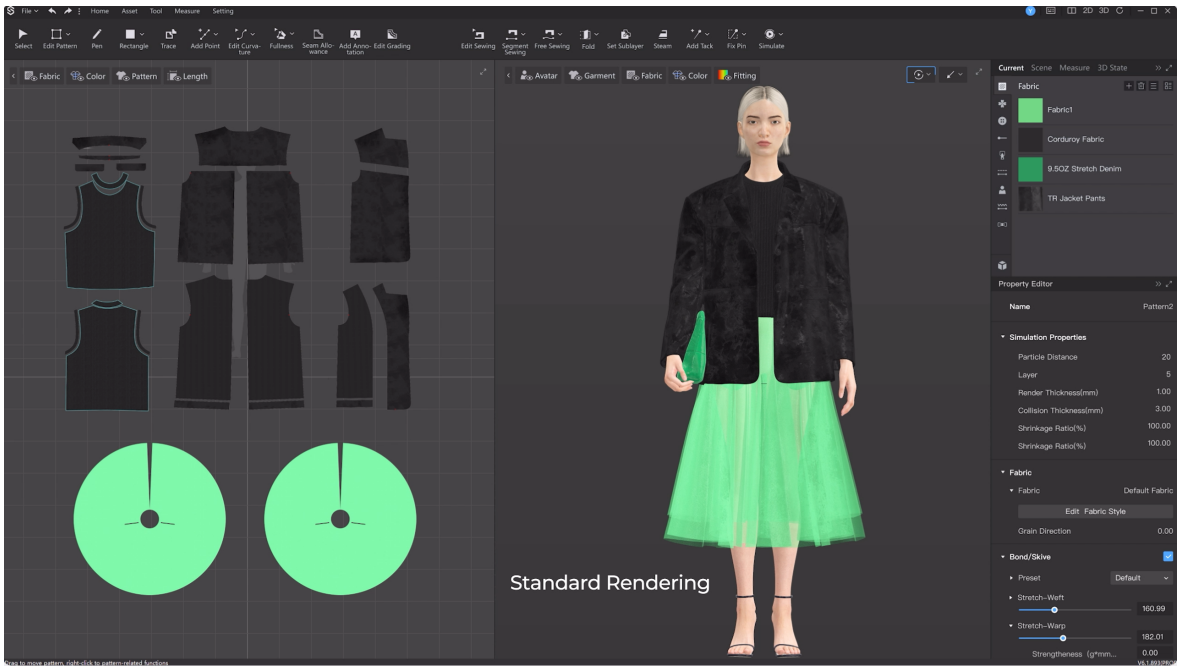


**Figure 2.1:** Style3D Studio linctex (2024). 3D garment modeling and simulation software.

## 2.3 Learning-based cloth deformation

In contrast to physics-based simulation, which typically requires solving large systems of nonlinear equations at each time step, learning-based methods aim at regressing the desired output via a single function learned by neural networks.

Many methods use *supervised* strategies, for which datasets are of great importance to enable training. Most of them Bertiche et al. (2020); Santesteban et al. (2019) opt for synthetic data generated with physics-based simulators such as Narain et al. (2012). A common practice of learning the 3D garment deformations is in the form of additive displacements added to a template mesh Gundogdu et al. (2019); Jiang et al. (2020); Patel et al. (2020); Santesteban et al. (2019); Vidaurre et al. (2020), as a function of parameters such as pose, shape and cloth design style.

Such displacements-based representation assumes one-to-one mapping between the body and the garment surface, primarily suited for tight-fit clothes that closely conform to the body in terms of both topology and shape. However, it shows reduced expressiveness when it comes to loose garments like skirts/dresses, given its reliance on body skinning weights. To cope with loose garments, the blend shapes and skinning weights Santesteban et al. (2021) are diffused for 3D points around the body surface and have been used to drive the shape of garments according to the body pose. Alternative representations have been investigated, including implicit representations Buffet et al. (2019); Corona et al. (2021); Santesteban et al. (2022b), patch-based surface representations Ma et al. (2021a), articulated dense point clouds Ma et al. (2021b), and sketches Wang et al. (2018). Despite these advancements, parametric patterns remain a prevalent representation due to the explicit control over style and topology Chen et al. (2022); Shen et al. (2020); Vidaurre et al. (2020).

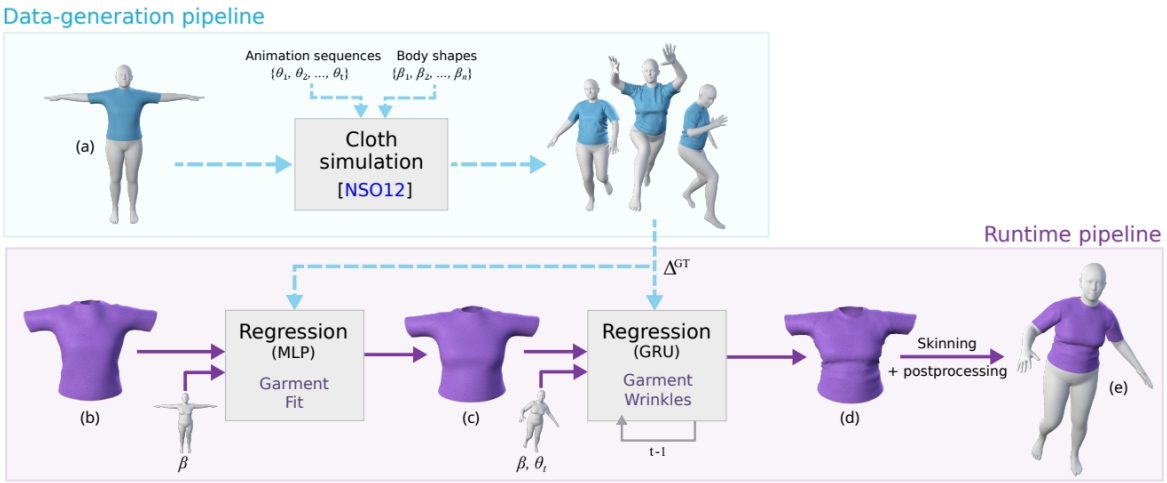


**Figure 2.2:** An example of learning-based methods for garment deformations based on displacements, drawn from Santesteban et al. (2019). As a large dataset is required by the supervised training schemes, the physics-based simulations are computed for multiple animated bodies wearing the garments. Once trained, the data-driven cloth deformation model infers by predicting the fit corrective displacements dependent on the body's shape and the dynamic wrinkle displacements dependent on the body's shape and pose. Then, the deformed unposed garment is skinned on the body to produce the final result.

Although supervised learning methods have proven to be highly effective, their dependence on ground-truth data for training presents a notable limitation, primarily due to the high costs and logistical difficulties involved in data acquisition. To circumvent the preparation of the ground-truth dataset, *self-supervised* strategies were proposed Bertiche et al. (2021); Santesteban et al. (2022a). Instead of relying on losses that evaluate the prediction error based on the difference with precomputed ground-truth samples, they adopt

physics-based losses as the supervision signal. The idea of self-supervised learning has propelled advances in more recent neural garment models. Grigorev et al. (2023) combines the graph neural network training with the self-supervision loss terms; De Luigi et al. (2023) conditions on a latent code for more generic garments draping using Unsigned Distance Function to represent garment surfaces; Li et al. (2024b) consists of flat 2D panels defined by 2D Signed Distance Function, and each panel is associated a 3D surface parameterized by the 2D panel coordinates.

## 2.4 Garment reconstruction

Reconstructing 3D representations of garments from real-world data is a well-established and extensively studied task. The input data varies widely, ranging from scans to monocular and multi-view images/videos.

In Section 1.2.1, we have introduced several related works. In addition, the techniques mentioned in the previous sections can also be applied to this task. The parametric models of 3D garments (Section 2.3) can be applied to reconstruct clothed people in images or scans, as the neural networks are inherently differentiable. The cloth parameters are optimized through trained neural networks to generate the template-mesh displacements Jiang et al. (2020); Patel et al. (2020) or unsigned neural fields Corona et al. (2021) for fitting monocular images or scans. In particular, as typical follow-up work of leaning-based methods, Li et al. (2024a) proposed a fitting method that leverages shape and deformation priors derived from synthetic data Li et al. (2024b) to achieve 3D garment reconstruction from static images. Su et al. (2023) learns a neural garment deformation model of clothing conditioned on the human poses by leveraging 3D supervised training in combination with unsupervised physical energies, they also extend to the recovery of fabric's physical parameters from human scans.

Guo et al. (2021) consider dynamics of the cloth geometry as secondary movement controlled by the body states, and they reconstruct only by optimizing body shape and pose through a simulation process. Li et al. (2024c); Yu et al. (2024) employ differentiable simulation (Section 2.2.1) to optimize 2D garment patterns and material properties for recovering clothing assets from static clothed individuals. PhysAvatar Zheng et al. (2024b) combines inverse rendering with a differentiable simulator to recover not only the garment shape but also the appearance of humans and the fabric of their clothes from multi-view video.

| Dataset | #Garment | #Subject | #Type | #Syn | #Pattern |
|---|---|---|---|---|---|
| DeepGarment Daněřek et al. (2017) | 2 | - | 2 | S | × |
| Wang et al. Wang et al. (2018) | 8 k | - | 3 | S | ✓ |
| Santesteban et al. Santesteban et al. (2019) | 7k | 17 | 1 | S | × |
| GarNet Gundogdu et al. (2019) | 1.5 k | - | 3 | S | × |
| TailorNet Patel et al. (2020) | 207 | 9 | 4 | S | × |
| Cloth3D Bertiche et al. (2020) | 11.3 k | - | 7 | S | × |
| Shen et al. Shen et al. (2020) | 104 | - | 5 | S | × |
| SMPLicit Corona et al. (2021) | 2 M | - | 11 | S | × |
| DIG Li et al. (2022a) | 200 | - | 2 | S | × |
| DrapeNet De Luigi et al. (2023) | 900 | 1 | 5 | S | × |
| NeuralTailor Korosteleva et Lee (2022) | 20k | 1 | 19 | S | ✓ |
| SewFactory Liu et al. (2023) | 19.1k | - | 8 | S | ✓ |
| Zhao et al Zhao et al. (2023) | 2 | - | 2 | S | × |
| CLOTH4D Zou et al. (2023) | 1 k | - | 6 | S | ✓ |
| ClothesNet Zhou et al. (2023) | 4.4 k | - | 11 | S | × |
| BUFF Zhang et al. (2017) | 24 | 6 | 4 | R | × |
| DeepWrinkles Lahner et al. (2018) | 4 | 2 | 2 | R | × |
| MGN Bhatnagar et al. (2019) | 712 | 356 | 5 | R | × |
| CAPE Ma et al. (2020) | 8 | 11 | 5 | R | × |
| DeepFashion3D Zhu et al. (2020) | 563 | - | 10 | R | × |
| GarmCap Lin et al. (2023) | 4 | - | 3 | R | × |
| Design2Cloth Zheng et al. (2024a) | 2k | **2k** | 31 | R | × |

**Table 2.1:** A summary of existing 3D garment datasets. #Garment: the number of different individual garment instances, #Subject: the number of individual identities, #Type: the number of distinct cloth types, #Syn: S for synthetic datasets generated with graphics simulation or R for real-world data, #Pattern: whether the sewing pattern is provided.

Another line of research relies heavily on mesh deformation. Earlier attempts include Xu et al. (2019) which applies Free-Form Deformation to deform a garment template, but only for conforming to the rough sizing information extracted from images. Pons-Moll et al. (2017) fits a clothing template shape to target clothing region in a non-rigid iterative closest point (ICP) manner. To provide good initialization for the nonrigid ICP optimization, Xiang et al. (2022, 2021) build a coarse correspondence between the target clothing

boundary with the template mesh boundary, and then apply the Biharmonic Deformation Fields to solve for per-vertex deformations that satisfy the boundary alignment constraints. Instead of operating within the Euclidean space, leveraging intrinsic manifold properties has proven effective for mesh alignment, such as utilizing the eigenfunctions of the Laplace-Beltrami operator on manifolds Ovsjanikov et al. (2012), which facilitates the establishment of correspondences between source and target meshes. Building on this approach, Lin et al. (2023) aligns garment geometry with real-world captures using a coarse-to-fine strategy that incorporates intrinsic encodings with neural deformation fields on garment meshes.

## 2.5 Sewing pattern estimation

Most garments are created from the well established foundational structure: sewing pattern. Typically, a base sewing pattern is composed of a set of regularly flat pieces. Garment sewing patterns provide a strong prior for garment shapes within the design space, allowing garments with the same topology to be uniformly represented. In practice, a parametric sewing pattern template is developed first, where the structure is designed to reflect the general characteristics of a garment category, enabling the description of a variety of garments. The parameterization of sewing patterns can vary significantly between different approaches, as demonstrated in Figure 2.3. Such parameterization may consist of numerical values, such as waist girth or sleeve length. Or it can be represented as an oriented loop of sequentially labeled edges. The position of the two endpoints of an edge, along with the location of a quadratic Bezier control point, defines the curvature of each edge.

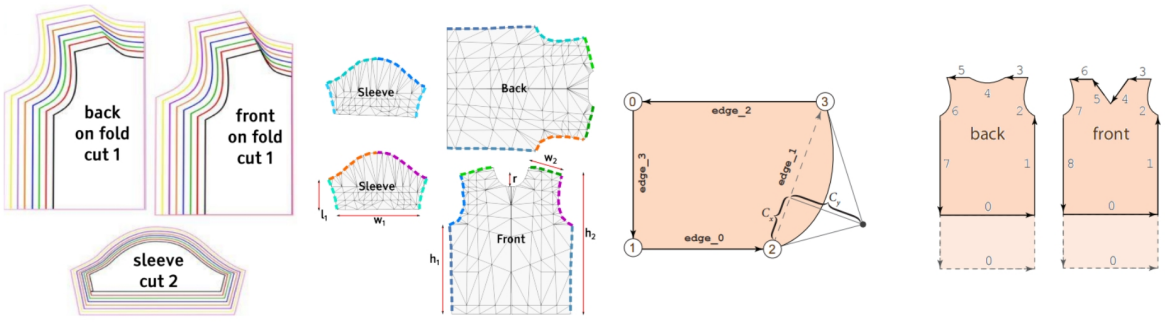


**Figure 2.3:** Parametric sewing pattern, taken from Yang et al. (2016) and Korosteleva et Lee (2022)

Closely related to our work, several recent studies have tackled the challenge of estimating 2D sewing patterns from 3D garment meshes.

### 2.5.1 3D-to-2D surface flattening

Several works consider the garment mesh as a developable surface, obtain 2D pattern panels by cutting the garment surface into 3D patches and then flatten each patch onto a plane. The cutting lines are found either by projecting the predefined seam lines from the body mesh onto the garment mesh Bang et al. (2021), along curvature directions Pietroni et al. (2022); Vaxman et al. (2016) on the surface, or through variational surface cutting Sharp et Crane (2018) to minimize the distortion induced by cutting and flattening Wolff et al. (2021). While intuitive and versatile, such geometric strategy is prone to generating patterns that deviate from traditional panel semantics, or lack of symmetry, making them unsuitable for garment production. Moreover, purely geometric methods Bang et al. (2021) do not account for the fabric's elasticity in the physical body-cloth interaction during the draping process, often leading to a sewing pattern that cannot accurately replicate the originally designed garment.

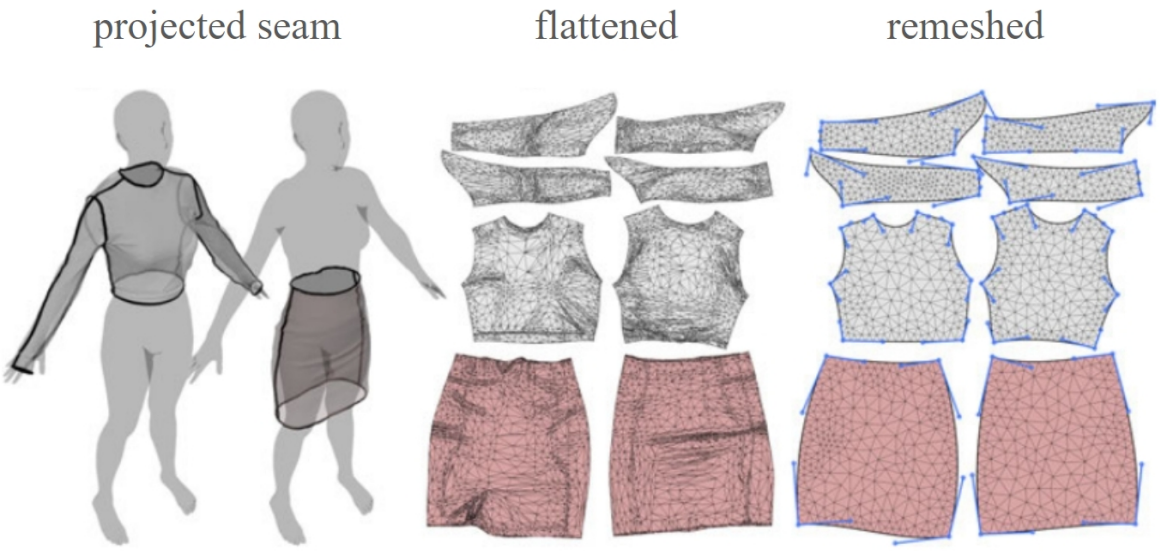


**Figure 2.4:** An example of flattening based pattern recovery, adapted from Bang et al. (2021). Projected seam curve on scan data (left), flattened pieces from scan data (middle), and re-meshed pattern with Bezier curve fitting (right).

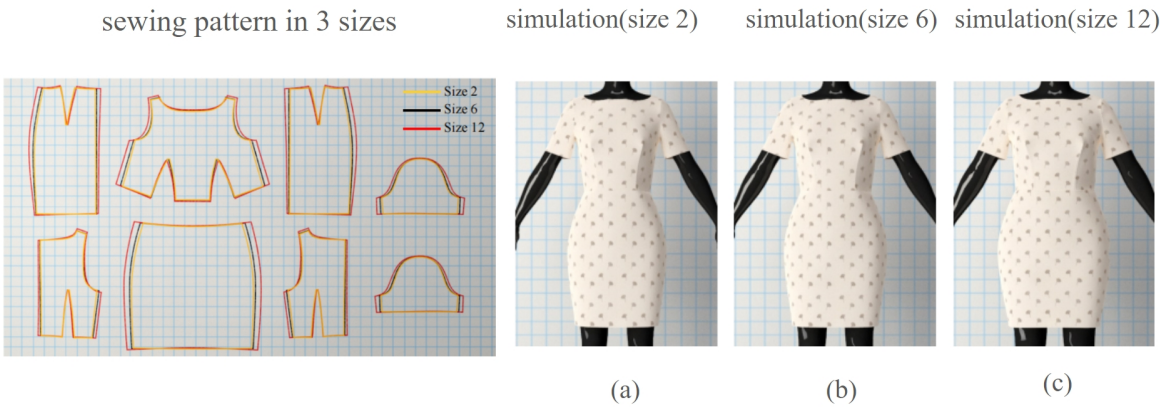


**Figure 2.5:** The grading results of a secretary dress example, adapted from Wang (2018). Given a base sewing pattern (black) and its simulated garment in (b), a pattern optimization automatically alters the base pattern into new patterns for fitting new bodies, as shown in (a) and (c).

### 2.5.2 Pattern geometry alteration

Several works regard 3D synthetic garments as the simulation result of their 2D pattern input. Patterns can be optimized using the physics simulator in an iterative manner. This optimization problem is complex, as the objective function depends not only on the parameters of the pattern but also on the quasistatic simulation process, typically formulated as:

$$U = \arg\min f\left(\mathrm{Sim}_B(U), U, B, \mathrm{Sim}_B(\bar{U}), \bar{U}, \bar{B}\right), \tag{2.1}$$

where $U$ is a set of parameters controlling the shape of a sewing pattern, and $\mathrm{Sim}_B(U)$ is a simulation function generating the vertex positions. Pattern adjustment can be defined as the minimization of an objective function $f$ (which varies according to the task of reconstruction or refitting), in which $\bar{U}$ controls the base pattern and $\bar{B}$ represents the base body.

To simplify this problem, existing techniques such as Bartle et al. (2016) introduce an intermediate variable and solve the problem in a two-phase approach comprising direct 3D garment editing and pattern alternation for fitting: a 3D garment shape optimization phase and an inverse 2D pattern design phase. While sharing the main idea of computing the sewing pattern inversely from the garment, they use a quasi-static simulation as a black box. Moreover, the solution obtained in the second phase generally does not meet the result elaborated in the first phase. Wang (2018) tries to solve only one constrained optimization for adjusting a standard sewing pattern, with a focus on garment grading tasks (e.g., resizing from S to XL) to achieve a good fit on a different body shape (Figure 2.5). The cost function is designed to accommodate garment grading for various body sizes, ensur-

ing that the resulting garments replicate the same degree of looseness, surface smoothness, and stretching ratio as observed in the base garment on the base body. Wolff et al. (2021) solves another problem of garment customization, by optimizing the garment rest shape, thereby improving the comfort of the cloth to a specific individual. The 2D pattern is indirectly altered by computing patch lines followed by geometric flattening.

### 2.5.3 Learning based estimation

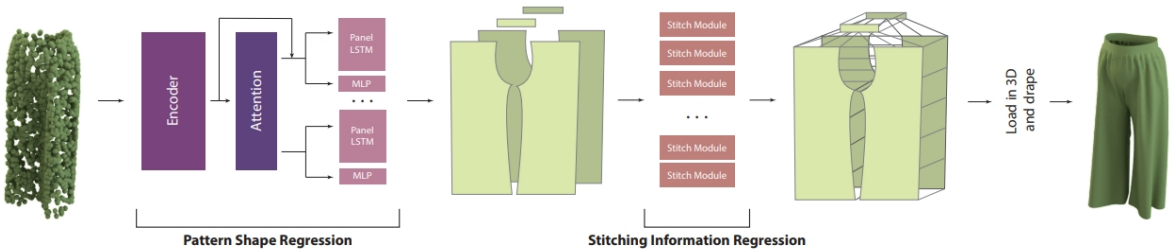


**Figure 2.6:** An example of learning-based pattern estimation method, taken from Korosteleva et Lee (2022). 2D garment sewing pattern is predicted from the 3D garment point cloud by recovering the structured panels and regressing stitching information via deep learning.

The work of Korosteleva et Lee (2022) explores a learning-based approach for estimating the sewing pattern of a given 3D garment shape. Leveraging a dataset of 3D garments with known sewing patterns across a variety of garment designs, their LSTM-based model is capable of regressing the sewing patterns representing the garment, as well as the stitching information among them. While intriguing, their model is limited to the settings presented in the training dataset: Specifically, as we will show later in Section 3.3 the model struggles when handling garments with different material properties than those used to generate the dataset. Likewise, it tends to show limited performance on garments draped on bodies other than the average SMPL female body in a T-pose(for which the garments have been draped and used for training). A plethora of works have followed this approach such as Chen et al. (2024); He et al. (2024); Nag et al. (2023). In particular, SewFormer Liu et al. (2023) trains a two-level Transformer network to regress sewing patterns from a single 2D human image, by using a dataset of images and sewing-pattern pairs covering a wide range of body poses and shapes, as well as garment types. All these learning methods, however, require a substantial dataset, which is expensive to build.

## 2.6 SMPL body modeling

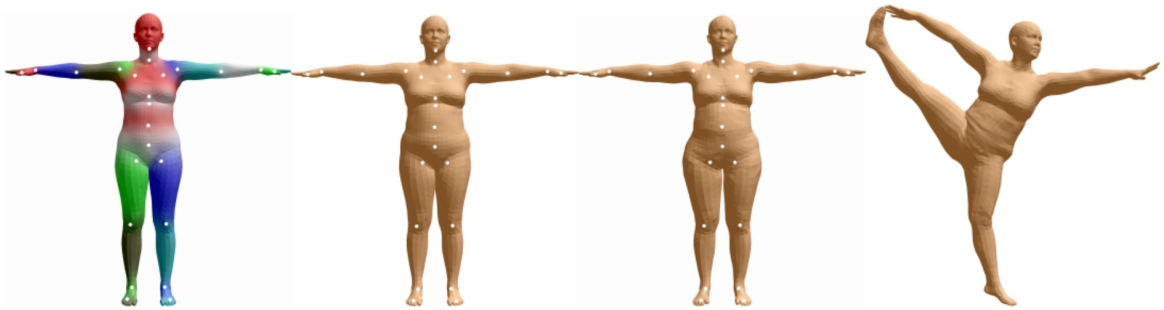


**Figure 2.7: SMPL model**, image taken from Loper et al. (2015). (a) The average template body mesh is represented as **T**, with blend weights indicated by color and joints shown in white. $B_S(\beta)$ and $B_P(\theta)$ are two blendshapes that output the per-vertex 3D displacements, which model respectively the deformations to change the body shape and the correctives to skinning artifacts. (b) and (c) are the meshes with the addition of shape and pose blend shapes in preparation for the blend skinning, according to the shape and pose parameters $\beta$ and $\theta$. (d) The vertices are reposed using dual quaternion skinning, resulting in the final deformed mesh.

SMPL Loper et al. (2015) is a widely used statistical human body model, providing a standard skeleton as well as a rigged body surface with shape ($\beta$) and pose ($\theta$) parameters. SMPL representation starts with a rigged template mesh in the rest pose. Given a pair of shape and pose parameters, the unposed body mesh is defined as:

$$T(\beta,\theta)=\mathbf{T}+B_S(\beta)+B_P(\theta), \tag{2.2}$$

where $\mathbf{T}\in\mathbb{R}^{n\times 3}$ is the base mesh with $n$ vertices, deformed by adding two blendshapes, pose-dependent deformations $B_P(\theta)\in\mathbb{R}^{n\times 3}$, and shape dependent deformations $B_S(\beta)\in\mathbb{R}^{n\times 3}$. After the offsets are added to the template, the resulting mesh $T(\beta,\theta)$ is further posed using the learned skinning function $W(\cdot)$:

$$B(\beta,\theta)=W\left(T(\beta,\theta),J(\beta),\theta,\mathcal{W}\right), \tag{2.3}$$

$\mathcal{W}$ are the blend weights of the skeleton $J(\cdot)$.

Since then, this approach has been extended to the modeling of fully articulated hands Romero et al. (2017), expressive faces Ranjan et al. (2018), and bodies incorporating both hands and faces Pavlakos et al. (2019b). The parameterization of SMPL provides a concise representation of body shapes and poses, facilitating the unification of extensive datasets

such as motion capture data Mahmood et al. (2019), densely annotated action dataset Punnakkal et al. (2021) and datasets of humans interacting with full realistic 3D scenes or in wild Hassan et al. (2019); von Marcard et al. (2018). Furthermore, substantial research focuses on fitting model parameters to images of real individuals Feng et al. (2021); Pavlakos et al. (2019a), which is particularly pertinent in applications like creating virtual avatars. Leveraging a parametric model significantly reduces the complexity of the solution space, which is crucial for accurately estimating avatars from ambiguous inputs such as monocular images.

## 2.7 Preliminaries of FEM

The simulation of deformable elastic objects has its roots in well-established disciplines such as mechanical engineering and computational physics. Among the various techniques, the Finite Element Method (FEM) has gained significant popularity in graphics and visual computing. This is primarily due to FEM's versatility in representing elastic objects with complex geometric features and diverse material properties. To characterize an elastic object deforming in space, it is necessary to define a quantitative formulation for the deformed shape and deduce the resulting forces thereafter. In earlier mass-spring systems, analogous concepts were applied to one-dimensional elastic strands. However, in the context of the Finite Element Method (FEM), these concepts are extended to accommodate the continuous nature of elastic deformation.

### 2.7.1 Deformation mapping and deformation gradient

Given an undeformed elastic object in space, it occupies a volume denoted by $\Omega$, which refers to the reference (or undeformed) configuration. When the object undergoes deformation, every point $P \in \Omega$ in the reference shape is displaced to its respective deformed location $x$. This displacement is described by the mapping function $x = \phi(P)$. The Jacobian matrix of the deformation map $\mathbf{F} \in \mathbb{R}^{3\times3}$ can be derived directly from $\frac{\partial x}{\partial P}$, referred to as the deformation gradient tensor. If we write $P = (P_x, P_y, P_z)$ and $x = (\phi_x, \phi_y, \phi_z)$. The

deformation gradient is written as:

$$\mathbf{F} = \frac{\partial\left(\phi_x, \phi_y, \phi_z\right)}{\partial\left(P_x, P_y, P_z\right)} = \begin{pmatrix} \partial\phi_x/\partial P_x & \partial\phi_x/\partial P_y & \partial\phi_x/\partial P_z \\ \partial\phi_y/\partial P_x & \partial\phi_y/\partial P_y & \partial\phi_y/\partial P_z \\ \partial\phi_z/\partial P_x & \partial\phi_z/\partial P_y & \partial\phi_z/\partial P_z \end{pmatrix}. \tag{2.4}$$

In general, $\mathbf{F}$ is spatially varying across $\Omega$, such dependence can be put explicitly with $\mathbf{F}(P)$.

### 2.7.2 Strain and stress

Elastic deformation results in the accumulation of potential energy within the deformed object, referred to as strain energy in the context of deformable solids. By assuming that the potential energy associated with a deformed configuration depends only on the initial and final configurations, and not on the deformation history taken over time to reach the current state (i.e. hyperelastic materials), we can denote the strain energy as $E(\phi)$. This notation indicates that the energy is solely determined by the deformation map of the given configuration.

Different parts of an elastic object may deform at different severity, the relationship between deformation and strain energy is more accurately defined on a local scale. This leads to the strain energy density function $\Psi(P)$, which measures strain energy per unit undeformed volume on the infinitesimal space $d\Omega$ around the $P$. The energy density is independent of the rigid translation, i.e. $\Psi$ is a function of the local deformation gradient alone, thereby $\Psi(\mathbf{F})$. The exact mathematical expression for $\Psi(\mathbf{F})$ will define the material properties being modeled.

In the context of continuum mechanics, a mathematical measure is used to describe the state of stress at a point within a material. It encapsulates the internal forces that neighboring particles of a material exert on each other. For hyperelastic materials, the 1st and 2nd Piola-Kirchhoff stress tensors are commonly used, which can be written as an explicit formula function of $\mathbf{F}$, and related to the strain energy density $\Psi$ via a simple formula. The formula for strain energy as a function of $\mathbf{F}$ and the formula for stress tensor as a function of $\mathbf{F}$ are equivalent.

However, the elements in the deformation gradient matrix $\mathbf{F}$ do not inherently provide an intuitive reflection of deformation severity. This complicates the design of constitutive

models for different behaviors and applications. Consequently, it is common to use intermediate quantitative descriptors derived from **F**, referred to as the strain tensor in the development of constitutive models. A strain tensor focuses on assessing deformation magnitude while disregarding irrelevant information unrelated to shape change, yielding a set of specific traits of deformation that influence energy or stress values more directly than the deformation gradient itself. Ideally, such a tensor is expected to only describe the shape deformation.

The Green strain tensor is a role model as strain measure. After applying a singular value decomposition to deformation gradient, $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$, only $\mathbf{V}^{\mathrm{T}}$ and **D** are relevant to deformation. The rotation part **U** can be removed to get:

$$\mathbf{G} = \frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\right) = \frac{1}{2}\left(\mathbf{V}\mathbf{D}^{2}\mathbf{V}^{\mathrm{T}} - \mathbf{I}\right), \tag{2.5}$$

defined as the Green strain. We note that **G** is rotation invariant. If there is no deformation, $\mathbf{G} = \mathbf{0}$. If deformation increases, $\|\mathbf{G}\|$ increases.

### 2.7.3 Constitutive models

The governing formula reveals material properties of a hyperelastic material could be an explicit formula for $Psi$ as a function of **F**, in constitutive models. A set of models has been proposed, they differ in the constitutive law and computing complexity, but share the same fashion to use specific intermediate quantities.

**Linear elastic model** is the simplest constitutive model, employing the small strain tensor $\xi = \frac{1}{2}\left(\mathbf{F} + \mathbf{F}^{\mathrm{T}}\right) - \mathbf{I}$ as an intermediate variable. the strain energy density is defined as

$$\Psi(\mathbf{F}) = \mu \operatorname{tr}\left(\xi^{2}\right) + \frac{\lambda}{2}\operatorname{tr}^{2}(\xi), \tag{2.6}$$

where tr is the trace, $\mu$ and $\lambda$ represent the Lamé coefficients, which can be derived from Young's modulus (indicating stretch resistance) and Poisson's ratio (indicating incompressibility).

The linear elasticity model exhibits several important characteristics. It is relatively straightforward to implement and computationally efficient since the nodal elastic forces have a linear relationship with nodal positions. However, it is accurate only for small deformations. Furthermore, the model lacks rotational invariance, as a rigid rotation results in a nonzero strain tensor $\xi$.

**St.Venant-Kirchhoff model** uses the Green strain tensor. It is worthy to note that the linear elasticity, is an approximation of the Green strain tensor for small deformations. In a similar fashion, the strain energy density for St.Venant-Kirchhoff elasticity can be defined as:

$$\Psi(\mathbf{F}) = \mu \operatorname{tr}\left(\mathbf{G}^2\right) + \frac{\lambda}{2}\operatorname{tr}^2(\mathbf{G}). \tag{2.7}$$

The St.Venant-Kirchhoff model exhibits rotational invariance and demonstrates plausible material response in various scenarios involving large deformations, where linear elasticity would not be suitable. However, there are trade-offs associated with this model. The relationship between nodal elastic forces and nodal positions is no longer linear, necessitating advanced solvers for numerical resolution. Additionally, the model has limitations in terms of its resistance to extreme compression.

**Corotational model** is introduced as a solution to the problem of rotational invariance in the linear model. By utilizing the Polar Decomposition $\mathbf{F} = \mathbf{RS}$, a new strain measure is constructed as $\mathbf{S} - \mathbf{I}$, which is linear on the symmetric tensor $\mathbf{S}$ obtained by factoring away the rotational component of $\mathbf{F}$. Replacing the small strain tensor with the modified tensor measure, the strain energy density for corotated elasticity is defined as follows:

$$\Psi(\mathbf{F}) = \mu \operatorname{tr}\left((\mathbf{S}-\mathbf{I})^2\right) + \frac{\lambda}{2}\operatorname{tr}^2(\mathbf{S}-\mathbf{I}). \tag{2.8}$$

The corotational model ensures rotational invariance in linear elasticity, albeit at the additional cost of performing the Polar decomposition. However, it still encounters the limitation that the reliable measure is only applicable to small-scale deformations.

### 2.7.4 Discretization with triangles

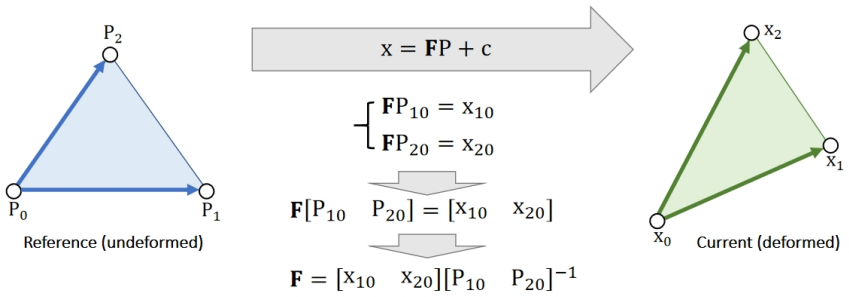


**Figure 2.8:** Deformation gradient is calculated by edge vectors for linear FEM.

For numerical simulation, the physical laws expressed relative to a continuous deformation in space need to be discretized: quantities such as the deformation map, the elastic

strain energy, stress tensors, and elastic forces all have to be reformulated as functions of discrete state variables. In a nutshell, linear thin shell FEM assumes that the deformation inside an elementary triangle is uniform, for any point $P$ in the reference triangle, its deformed correspondence follows $x = \mathbf{F}P + c$. For any vector between two points, we can use $\mathbf{F}$ to convert it from reference to deformed: $x_{ba} = x_b - x_a = \mathbf{F}P_b + c - \mathbf{F}P_a - c = \mathbf{F}P_{ba}$. Therefore, we can calculate the deformation gradient by edge vectors as shown in Figure 2.8. Assuming that the reference triangle is flat on 2D, $\mathbf{F}$ is of size $\mathbb{R}^{3\times 2}$, which is related to deformation but also contains rotation:

$$\mathbf{F} = \begin{bmatrix} x_{10} & x_{20} \end{bmatrix} \begin{bmatrix} P_{10} & P_{20} \end{bmatrix}^{-1} = \begin{bmatrix} x_{10} & x_{20} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ax_{10} + cx_{20} & bx_{10} + dx_{20} \end{bmatrix} \tag{2.9}$$

Accordingly, the green stain is a symmetric matrix, which has thus only 3 independent entries:

$$\mathbf{G} = \frac{1}{2}\left(\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{I}\right) = \begin{bmatrix} \varepsilon_{uu} & \varepsilon_{uv} \\ \varepsilon_{uv} & \varepsilon_{vv} \end{bmatrix}, \tag{2.10}$$

which leads to an energy density defined in Equation (2.11) if the St.Venant-Kirchhoff model is being employed:

$$\Psi\left(\varepsilon_{uu}, \varepsilon_{vv}, \varepsilon_{uv}\right) = \frac{\lambda}{2}\left(\varepsilon_{uu} + \varepsilon_{vv}\right)^2 + \mu\left(\varepsilon_{uu}^2 + \varepsilon_{vv}^2 + 2\varepsilon_{uv}^2\right). \tag{2.11}$$

The force, defined as the gradient of energy to position can be thus deduced, here we take the elastic force applied on $x_1$ as an example:

$$\begin{aligned}
f_1 &= -A^{\mathrm{ref}}\left(\frac{\partial\Psi}{\partial\varepsilon_{uu}}\frac{\partial\varepsilon_{uu}}{\partial x_1} + \frac{\partial\Psi}{\partial\varepsilon_{vv}}\frac{\partial\varepsilon_{vv}}{\partial x_1} + \frac{\partial\Psi}{\partial\varepsilon_{uv}}\frac{\partial\varepsilon_{uv}}{\partial x_1}\right)^{\mathrm{T}} \\
&= -A^{\mathrm{ref}}\left(\frac{\partial\Psi}{\partial\varepsilon_{uu}}a\left(ax_{10} + cx_{20}\right)^{\mathrm{T}} + \frac{\partial\Psi}{\partial\varepsilon_{vv}}b\left(bx_{10} + dx_{20}\right)^{\mathrm{T}} + \frac{\partial\Psi}{\partial\varepsilon_{uv}}\frac{1}{2}a\left(bx_{10} + dx_{20}\right)^{\mathrm{T}} + \frac{\partial\Psi}{\partial\varepsilon_{uv}}\frac{1}{2}b\left(ax_{10} + cx_{20}\right)^{\mathrm{T}}\right) \\
&= -A^{\mathrm{ref}}\left(\begin{bmatrix} ax_{10} + cx_{20} & bx_{10} + dx_{20} \end{bmatrix}\begin{bmatrix} \frac{\partial\Psi}{\partial\varepsilon_{uu}}a + \frac{\partial\Psi}{\partial\varepsilon_{uv}}\frac{1}{2}b \\ \frac{\partial\Psi}{\partial\varepsilon_{uv}}\frac{1}{2}a + \frac{\partial\Psi}{\partial\varepsilon_{vv}}b \end{bmatrix}\right) \\
&= -A^{\mathrm{ref}}\begin{bmatrix} ax_{10} + cx_{20} & bx_{10} + dx_{20} \end{bmatrix}\begin{bmatrix} \frac{\partial\Psi}{\partial\varepsilon_{uu}} & \frac{1}{2}\frac{\partial\Psi}{\partial\varepsilon_{uv}} \\ \frac{1}{2}\frac{\partial\Psi}{\partial\varepsilon_{uv}} & \frac{\partial\Psi}{\partial\varepsilon_{vv}} \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = -A^{\mathrm{ref}}\mathbf{F}\mathbf{S}_{\mathbf{PK}}\begin{bmatrix} a \\ b \end{bmatrix}
\end{aligned} \tag{2.12}$$

$\mathbf{S_{PK}}$ is known as the second Piola-Kirchhoff stress, given as:

$$\mathbf{S_{PK}} = \begin{bmatrix} \frac{\partial \Psi}{\partial \varepsilon_{uu}} & \frac{1}{2}\frac{\partial \Psi}{\partial \varepsilon_{uv}} \\ \frac{1}{2}\frac{\partial \Psi}{\partial \varepsilon_{uv}} & \frac{\partial \Psi}{\partial \varepsilon_{vv}} \end{bmatrix} = \begin{bmatrix} 2\mu\varepsilon_{uu} + \lambda\varepsilon_{uu} + \lambda\varepsilon_{vv} & 2\mu\varepsilon_{uv} \\ 2\mu\varepsilon_{uv} & 2\mu\varepsilon_{vv} + \lambda\varepsilon_{uu} + \lambda\varepsilon_{vv} \end{bmatrix} = 2\mu\mathbf{G} + \lambda\,\mathrm{trace}(\mathbf{G})\mathbf{I} \tag{2.13}$$

### 2.7.5 FEM-based cloth simulation

**Algorithm 1:** Cloth Simulation

```
v_0 ← 0;
for t = 1 to n do
    M, f ← compute_forces(x, v);
    a_t ← M^{-1} f;
    v_t ← v_{t-1} + a_t Δt;
    x_t ← x_{t-1} + v_t Δt;
    x_t ← x_t + collision_response(x_t, v_t, x_t^body, v_t^body);
    v_t ← (x_t − x_{t-1})/Δt;
end
```

Generally, cloth simulation includes three essential steps: force computation, dynamic solving, and collision handling. For a draping garment, the most straightforward external force is the gravity. Considering the intrinsic property of the fabrics and the interaction between the body and cloth, the internal, constraint, and frictional forces should also be modeled. In the Section 2.7.4, we discussed how to follow Finite Element Method (FEM) discretization to compute strain forces. Given a cloth mesh $X_t$ together with an underlying body mesh $B_t$ at time step $t$, a cloth simulator can compute the mesh state $X_{t+1}$ at the next step $t+1$ based on the computed internal and external forces and the collision response. A simple simulation pipeline is shown in Algorithm 1, as is common in graphics, we assume that $M$ is the diagonal lumped-mass matrix, $f$ is the force, and $a$ is the acceleration. To get the acceleration $a$ for updating the velocity and position accordingly, the simplest way is to solve the linear system $Ma = f$. However, this Forward Euler method suffers from the well-known stability issue and often limits the time step size for the simulation. In order to take larger steps for faster simulation, Backward Euler is often used. More specifically, we want our acceleration to match the force computed in the next time

step.

$$\begin{cases} v_{t+1} = v_t + \Delta t M^{-1} f(x_{t+1}) \\ x_{t+1} = x_t + \Delta t v_{t+1} \end{cases} \tag{2.14}$$

By using Taylor Expansion, we have:

$$\begin{aligned} f(x_{t+1}) &= f(x_t) + \left.\frac{\partial f}{\partial x}\right|_{x_t} \left(\frac{x_{t+1} - x_t}{\Delta t}\right) \Delta t \\ &= f(x_t) + \left.\frac{\partial f}{\partial x}\right|_{x_t} v_{t+1} \Delta t \\ &= f(x_t) + \left.\frac{\partial f}{\partial x}\right|_{x_t} (v_t + \Delta v) \Delta t \end{aligned} \tag{2.15}$$

Then, the Equation (2.14) can be reformulated as:

$$\begin{aligned} v_{t+1} - v_t &= \Delta t M^{-1} \left( f(x_t) + \left.\frac{\partial f}{\partial x}\right|_{x_t} (v_t + \Delta v) \Delta t \right) \\ \Delta v &= \Delta t M^{-1} \left( f(x_t) + \left.\frac{\partial f}{\partial x}\right|_{x_t} (v_t + \Delta v) \Delta t \right) \\ \Delta v &= \Delta t M^{-1} f(x_t) + \Delta t^2 M^{-1} \left.\frac{\partial f}{\partial x}\right|_{x_t} v_t + \Delta t^2 M^{-1} \left.\frac{\partial f}{\partial x}\right|_{x_t} \Delta v \\ \left(I - \Delta t^2 M^{-1} \left.\frac{\partial f}{\partial x}\right|_{x_t}\right) \Delta v &= \Delta t M^{-1} f(x_t) + \Delta t^2 M^{-1} \left.\frac{\partial f}{\partial x}\right|_{x_t} v_t \\ \left(M - \Delta t^2 \left.\frac{\partial f}{\partial x}\right|_{x_t}\right) \Delta v &= \Delta t f(x_t) + \Delta t^2 \left.\frac{\partial f}{\partial x}\right|_{x_t} v_t \end{aligned} \tag{2.16}$$

, which is a new linear system of $\Delta v$. As long as we have the Jacobian of the forces $J = \frac{\partial f}{\partial x}$, we can compute a more stable result of $\Delta v$ and can apply larger $\Delta t$. For simplicity, the two sides of the linear solve can be denoted by ignoring the time subscript as:

$$\hat{\mathbf{M}} = \left(M - \Delta t^2 \frac{\partial f}{\partial x}\right), \quad \hat{\mathbf{f}} = f + \Delta t \frac{\partial f}{\partial x} v. \tag{2.17}$$

The global system matrix $\hat{\mathbf{M}}$ is $3n \times 3n$ symmetric, which is positive-definite. where $n$ is the number of mechanical DOFs. Eventually, the exact solution can be provided by computing the actual inverse of the system matrix as in the fourth step of Algorithm 1, but this is not recommended for large matrices. More generally, the solution is obtained by computing a factorization such as the Cholesky factorization or QR decomposition. Such direct solvers provide robust results but the computation cost tends to be expensive with larger scale, which becomes prohibitive for real-time performance in large-scale simulations. Besides, direct solvers require the explicit assembly of global matrices to compute

the factorization or decomposition of the system. The forces applied by the element to its nodes are split and added to the global force vector based on the node indices in the mesh, as illustrated in the following Figure 2.9. The assembly step may become the bottleneck when combined with efficient solvers, as we will address in Chapter 3.

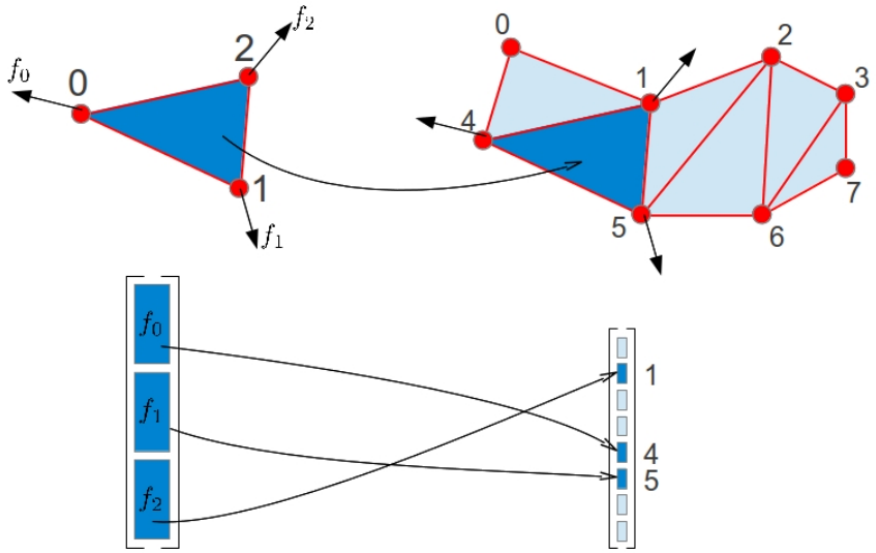


**Figure 2.9:** The force vector of an element (dark blue) is split and accumulated to the global force vector (light blue) based on the node indices. Copyright @Francois Faure.

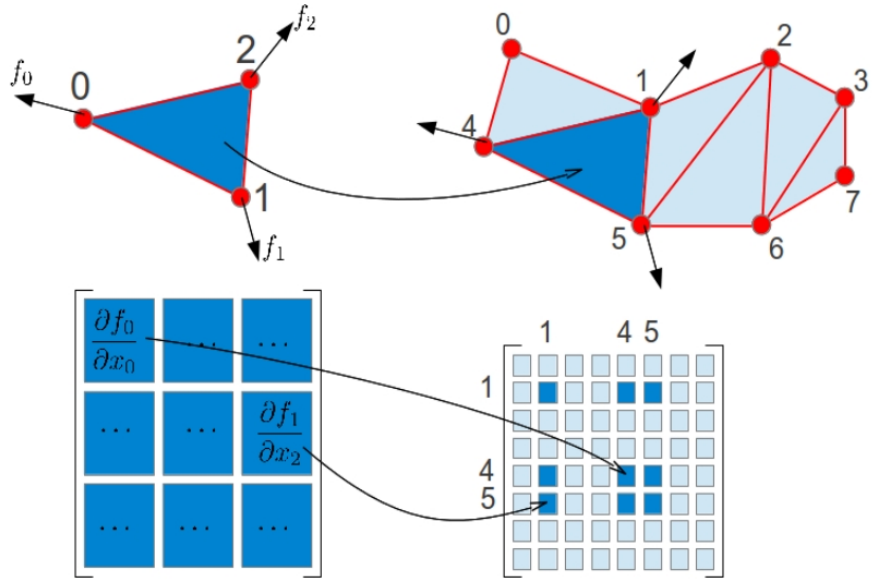


**Figure 2.10:** The Jacobian matrix of an element (dark blue) is split and accumulated to the global Jacobian matrix (light blue) based on the node indices. Copyright @Francois Faure.

On the other hand, iterative methods are usually used as the number of iterations can be set to compute an approximated solution while having more control over the time spent during the solving process. Popular methods like the Conjugate Gradient (CG) algorithm and the parallel skills are also abundant in the real-time simulation literature.

In Section 2.7.4 we illustrate how the first derivatives of the strain energy are computed to obtain the forces explicitly in FEM for dynamics updates. Alternatively, some work proposes to tackle the problem from an optimization point of view. Note that Equation (2.14) can be reformulated as:

$$M\frac{x_{t+1}-x_t-\Delta t\, v_t}{\Delta t^2} = f\left(x_{t+1}\right), \tag{2.18}$$

If a function $h$ is defined as:

$$h(x) = M\frac{x-x_t-\Delta t\, v_t}{\Delta t^2} + \frac{\partial \Phi}{\partial x}, \tag{2.19}$$

where $\Phi$ is the potential energy due to internal and external forces of the system. The solution of equations of motion is equal to $h(x) = 0$. $h(x)$ can be integrated to $E(x)$, allows recast the root-finding as a minimization problem of Gast et al. (2015). The discretization and dynamics remain unchanged, and they still address the same set of equations. Further minimization using Newton's method would require the Hessian of $E(x)$, which involves the first derivatives of forces.

$$E(x) = \frac{1}{2\Delta t^2}(x-\hat{x})^T M(x-\hat{x}) + \Phi, \tag{2.20}$$

where $\hat{x} = x_t + \Delta t\, v_t$ is a tentative position update.

$$x_{t+1} = \arg\min_x \frac{1}{2\Delta t^2}(x-\hat{x})^\top M(x-\hat{x}) + \Phi, \tag{2.21}$$

The objective function of this minimization is also the central ingredient of a self-supervised learning scheme Santesteban et al. (2022a). They proposed to leverage such optimization-based formulation to define a set of losses:

$$\mathscr{L} = \mathscr{L}_{\text{inertia}} + \mathscr{L}_{\text{static}}, \tag{2.22}$$

where $\mathscr{L}_{\text{inertia}}$ models the inertia of the garment and it is defined analogously to the first term of Equation (2.20). And $\mathscr{L}_{\text{static}}$, models the potential energy $\Phi$ of Equation (2.20) which represents the internal and external forces that affect the garment. The sum of different physics-based terms models the energies that emerge on deformable solids. We will discuss in more detail in Chapter 4.

### 2.7.6 Differentiable FEM cloth simulation

To make the physics-based simulation differentiable, it is essential to manage the computation and propagation of gradients throughout the system. In Algorithm 1, the majority of operations consist of primitive functions, many of which are directly supported by PyTorch Paszke et al. (2019) through the built-in operators. However, the linear solve (Line 4) and collision response (Line 7) must be registered as custom operations Liang et al. (2019). The derivatives of these operations need to be exposed to PyTorch to leverage its tensor-based automatic differentiation, which tracks the computational graph for gradient backpropagation.

Supposing the linear solve Equation (2.16) $\hat{\mathbf{M}}\mathbf{a} = \hat{\mathbf{f}}$ has the solution $\mathbf{z}$, and with the propagated gradient $\frac{\partial \mathscr{L}}{\partial \mathbf{a}}\Big|_{\mathbf{a}=\mathbf{z}}$, the gradients can be derived using implicit differentiation from Liang et al. (2019) and chain rules as:

$$\hat{\mathbf{M}}\partial\mathbf{a} = \partial\hat{\mathbf{f}} - \partial\hat{\mathbf{M}}\mathbf{a}. \tag{2.23}$$

For clarity of expression, we represent the matrix $\hat{\mathbf{M}}$ using the elements $\hat{\mathbf{M}}_{ij}$.

$$\begin{aligned}
\hat{\mathbf{M}}\partial\mathbf{a} = -\partial\hat{\mathbf{M}}\mathbf{z} = -&\begin{bmatrix} \partial\hat{\mathbf{M}}_{11} & \partial\hat{\mathbf{M}}_{12} & \cdots & \partial\hat{\mathbf{M}}_{1j} & \cdots & \partial\hat{\mathbf{M}}_{1n} \\ \partial\hat{\mathbf{M}}_{i1} & \partial\hat{\mathbf{M}}_{i2} & \cdots & \partial\hat{\mathbf{M}}_{ij} & \cdots & \partial\hat{\mathbf{M}}_{in} \\ \vdots & & & & & \vdots \\ \partial\hat{\mathbf{M}}_{n1} & \partial\hat{\mathbf{M}}_{n2} & \cdots & \partial\hat{\mathbf{M}}_{nj} & \cdots & \partial\hat{\mathbf{M}}_{nn} \end{bmatrix}\begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_n \end{pmatrix} \\
= -&\begin{pmatrix} \partial\hat{\mathbf{M}}_{11}\mathbf{z}_1 + \partial\hat{\mathbf{M}}_{12}\mathbf{z}_2 + \cdots \partial\hat{\mathbf{M}}_{1j}\mathbf{z}_j + \cdots + \partial\hat{\mathbf{M}}_{1n}\mathbf{z}_n \\ \vdots \\ \partial\hat{\mathbf{M}}_{i1}\mathbf{z}_1 + \partial\hat{\mathbf{M}}_{i2}\mathbf{z}_2 + \cdots \partial\hat{\mathbf{M}}_{ij}\mathbf{z}_j + \cdots + \partial\hat{\mathbf{M}}_{in}\mathbf{z}_n \\ \vdots \\ \partial\hat{\mathbf{M}}_{n1}\mathbf{z}_1 + \partial\hat{\mathbf{M}}_{n2}\mathbf{z}_2 + \cdots \partial\hat{\mathbf{M}}_{nj}\mathbf{z}_j + \cdots + \partial\hat{\mathbf{M}}_{nn}\mathbf{z}_n \end{pmatrix}
\end{aligned} \tag{2.24}$$

By splitting $\hat{\mathbf{M}}_{ij}$ from irrelevant variables, we obtain from Equation (2.24) that:

$$\hat{\mathbf{M}}\partial\mathbf{a} = -\partial\hat{\mathbf{M}}_{ij}\begin{bmatrix} 0 \\ \vdots \\ \mathbf{z}_j \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \\ all \\ the \\ rest \\ \\ \end{bmatrix} \tag{2.25}$$

Here we have the derivative of $\mathbf{a}$ with respect to $\hat{\mathbf{M}}_{ij}$:

$$\frac{\partial\mathbf{a}}{\partial\hat{\mathbf{M}}_{ij}} = -\hat{\mathbf{M}}^{\dagger}\begin{bmatrix} 0 \\ \vdots \\ \mathbf{z}_j \\ \vdots \\ 0 \end{bmatrix} \tag{2.26}$$

To eliminate the $\hat{\mathbf{M}}^{\dagger}$, an "intermediate" tensor $\mathbf{d_a}$ is introduced by solving another linear system:

$$\hat{\mathbf{M}}^{\top}\mathbf{d_a} = \frac{\partial\mathscr{L}^{\top}}{\partial\mathbf{a}}. \tag{2.27}$$

So that the derivative of $\mathscr{L}$ with respect to $\hat{\mathbf{M}}_{ij}$ can be deduced by chain rule:

$$\frac{\partial\mathscr{L}}{\partial\hat{\mathbf{M}}_{ij}} = \frac{\partial\mathscr{L}}{\partial\mathbf{a}}\cdot\frac{\partial\mathbf{a}}{\partial\hat{\mathbf{M}}_{ij}} = \mathbf{d}_{\mathbf{a}}^{\top}\hat{\mathbf{M}}\hat{\mathbf{M}}^{\dagger}\begin{bmatrix} 0 \\ \vdots \\ -\mathbf{z}_j \\ \vdots \\ 0 \end{bmatrix} \tag{2.28}$$

Combining all elements in $\hat{\mathbf{M}}$ together we have:

$$\frac{\partial\mathscr{L}}{\partial\hat{\mathbf{M}}} = -\mathbf{d_a}\mathbf{z}^{\top}. \tag{2.29}$$

The derivation of $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{f}}}$ is easier:

$$\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{f}}} = \frac{\partial \mathscr{L}}{\partial \mathbf{a}} \cdot \frac{\partial \mathbf{a}}{\partial \hat{\mathbf{f}}} = \mathbf{d}_{\mathbf{a}}^{\top} \hat{\mathbf{M}} \cdot \hat{\mathbf{M}}^{\dagger} \mathbf{I} = \mathbf{d}_{\mathbf{a}}^{\top}. \tag{2.30}$$

The analytical derivatives of the linear solver to compute $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{M}}}$ and $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{f}}}$ are used to create the new custom "linear solve" operation to PyTorch, and to have it behave like PyTorch's built-in operators with the gradients $\frac{\partial \mathscr{L}}{\partial \mathbf{a}}$ backpropagated from $\mathscr{L}$.

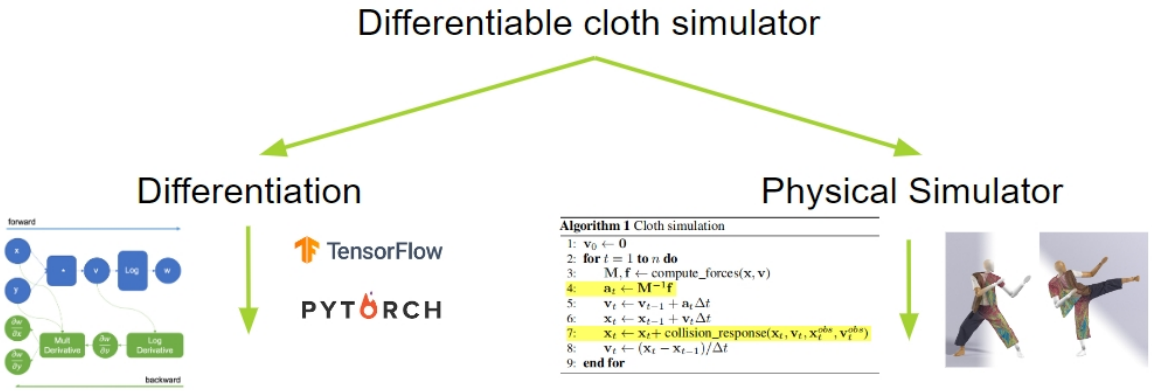


**Figure 2.11:** Auto-differentiation to backpropagate gradients through the physics-based cloth simulation.

Additionally, due to the computational demands of draping simulations, the system should be both differentiable and efficient to ensure practicality. Consequently, key computational bottlenecks must be optimized, as discussed in the Chapter 3.

CHAPTER 3

# IGPM: INVERSE GARMENT AND PATTERN MODELING WITH A DIFFERENTIABLE SIMULATOR

## 3.1 Introduction

The research of recovering realistic clothed human avatars has drawn increasing attention in the past few years as discussed in Chapter 2. The ability to generate simulation-ready garment digital twins from 3D shapes of dressed people has a wide range of applications in virtual try-on, garment reverse engineering, and social AR/VR. It will allow, from the retrieved garment models, to obtain new animation, or to better capture and interpret subsequent garment geometry undergoing deformation. This is particularly compelling given the increasing accessibility of detailed 3D scans of people with clothing. Such a garment recovery system should ideally satisfy the following: high fidelity to faithfully replicate the given 3D geometry, adaptability to obtain new garment simulations on different body shapes and poses, and the ability to recover 2D patterns, to conform to the standard garment modeling processes used in both the fashion industry and cloth simulation software.

In this chapter, we address the challenging problem of converting a given 3D shape of dressed garment to an animatable form by estimating its precise 2D pattern shape. Such pattern-based modeling closely mimics the design process for both real-world and synthetic garments, and effectively disentangles the inherent shape from deformations caused by external forces and internal fabric properties during draping. Based on a differentiable physics-based simulator, our system solves an inverse simulation problem: iteratively optimizing both the pattern shape and physical parameters to ensure that the draped garment mesh on the estimated body aligns with the target garment shape. The ability to estimate garment patterns facilitates the adaptation of the reconstructed garment to new conditions for downstream applications. New animations on different body shapes or poses can be synthesized by placing and seaming the produced pattern around the body mesh prior to the draping simulation. In addition, the approach does not require any training data and is capable of faithfully replicating intricate garment shapes. We evaluate our method across various garment types and demonstrate that it produces patterns and garment counterparts of promising quality. Compared to the state-of-the-art methods, ours achieves superior performance in terms of both reconstruction and pattern accuracy.

# 3.2 Method

## 3.2.1 Overview

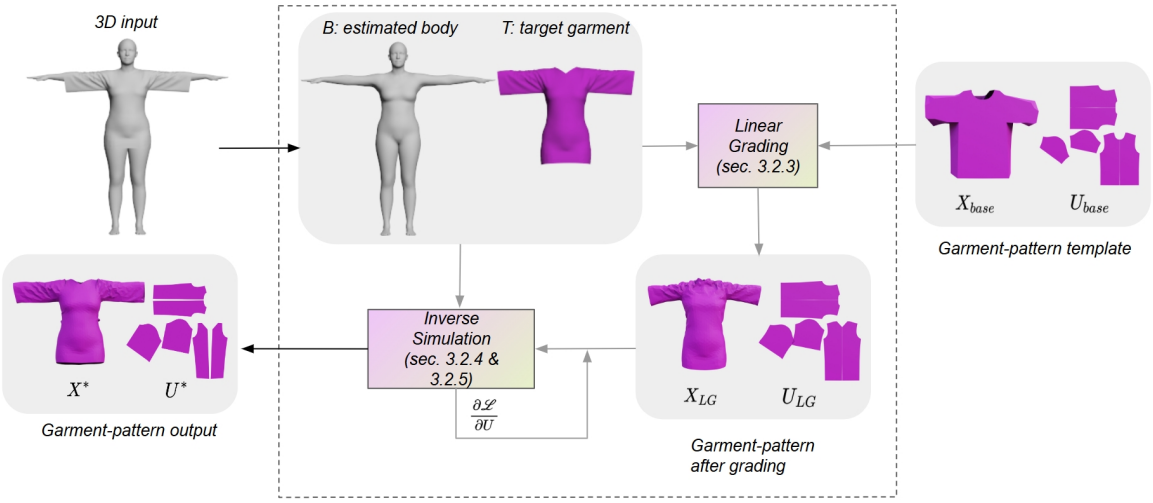


**Figure 3.1:** Given a 3D representation of a dressed garment, our method generates a simulation-ready garment by accurately estimating its corresponding 2D pattern. The garment mesh $T$ serves as the target in a two-phase fitting process, where both the sewing pattern and material parameters are jointly optimized. Initially, a linear grading step captures the overall proportions and dimensions geometrically. This is followed by an inverse simulation phase, in which optimization, coupled with a differentiable simulation, refines the fit and geometric details.

In this section, we describe our method outlined in Figure 3.1. Drawing an analogy to garment production, the garment geometry in our work is determined by the style and size of its sewing pattern, which is parameterized for efficient modification (Section 3.2.2). The first component of our system is the linear grading which accounts for capturing the coarse geometry such as size and proportion (Section 3.2.3). The second component further refines the model to capture the detailed garment shape and precise pattern. At the heart of our technique is an optimization-driven pattern refinement based on a differentiable cloth simulator (Section 3.2.4, Section 3.2.5), where the simulated garment is iteratively altered along with the physical parameters.

## 3.2.2 Representation of base pattern and body

The garment shape is determined by the shape and size of its sewing pattern, which is a collection of 2D panels that are placed around the wearer's body and stitched together at

an initial stage of the later simulation (Figure 1.2). We observe that many garments share a same pattern topology, with geometric variations. Therefore, our system provides several base models selected from the Berkeley Garment Library Narain et al. (2012). These base models, i.e. 2D patterns and their corresponding sewn 3D meshes ($U_{base}$ and $X_{base}$ in Figure 3.1, respectively), are available for representative garment categories. Although several methods exist for estimating a base model given a garment mesh, we currently let the user select a base model depending on the target garment type. Optionally, users can incorporate customized base models into the system. Figure 3.2 shows three base models (t-shirt/dress, pants, and skirt) used in our experiments.

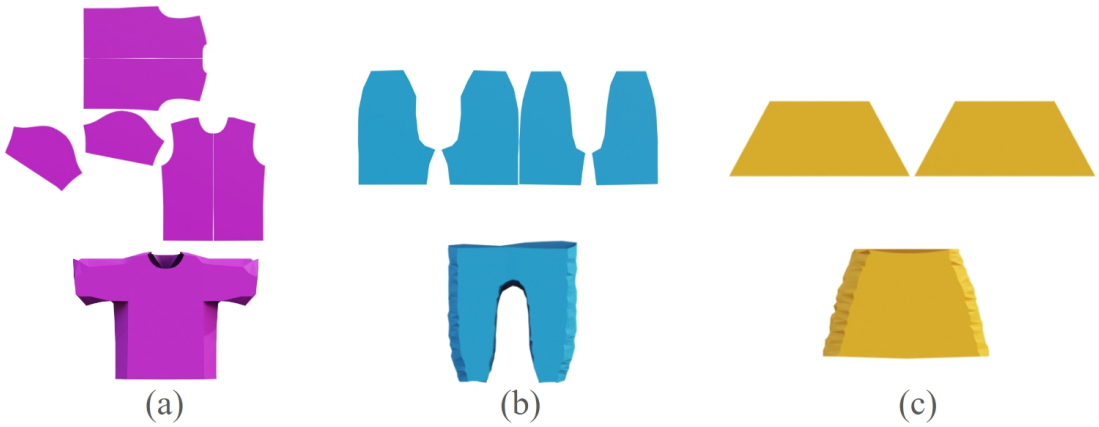


**Figure 3.2:** Base models for three garment topologies. (a) t-shirt/dress; (b) pants; and (c) skirt.

**Parameterization.** The planar pattern mesh $U$ serves as the FEM reference (prior to any deformation) for its corresponding 3D garment mesh $X$ during the later simulation. The mapping between a 2D vertex $u \in U$ to a 3D node $x \in X$ is known from the base models, in the form of a UV Map. A panel is a 2D triangular mesh bounded by a number of piece-wise curves parameterized by a set of control points. As shown in Figure 3.3, two curves join at a control point $c^i$, which is typically the vertex of $C^0$ curvature discontinuity (*corner point*), or the vertex having two or more *seam-counterparts* belonging to other panels (*join point*) which will be merged into one node at the time of sewing. Note that the same control point can be both a corner and a join point (purple points in Figure 3.3).

The control points are grouped into disjoint sets $C = \{C_p\}$, one for each panel $p$. Within a panel, the control points are ordered in a counterclockwise manner, i.e. $C_p = \{c_p^i\}$. Throughout the pattern optimization process, the control points serve as variables, while all other mesh points are repositioned to maintain the relative locations with respect to them (See Section 3.2.3 for the repositioning method). This effectively reduces the dimensionality of the solution space.

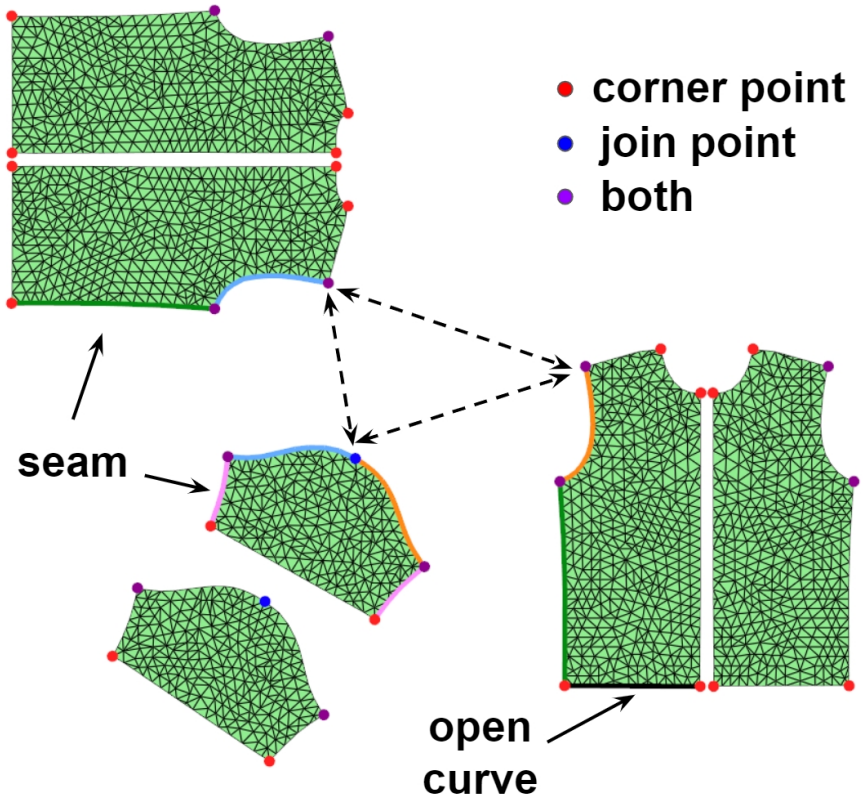


**Figure 3.3:** Control points on the T-shirt base pattern mesh. Specific seams are colored consistently across different panels for clarity.

**Symmetry detection.** Let $K$ denote the effective control point group, initially set to $\{C_p\}$. To further reduce the dimensionality of $K$ and to preserve the pattern symmetry (which is often a desirable property in garment production) during pattern adjustment, we detect the pattern symmetry in two steps: It first detects *inter-panel symmetry* by computing for each pair of panels an aligning rigid transformation Schönemann (1966) and evaluating the quality of alignment. If their alignment score is sufficiently high, we remove one of the two panels from $K$. Next, we perform *intra-panel symmetry* detection within $K$, by computing for each pair of control points $c_p^i, c_p^j \in C_p$ its axis of symmetry and evaluating the symmetry score for the remainder of control points in the panel. The control points pair with the highest score exceeding a predefined threshold is used to identify left-to-right symmetry within a panel, subsequently leading to a further reduction of control points from $K$. During the symmetry detection, we identify a transformation matrix (flipping, rotation) for each symmetry pair of control points. With the new coordinates of effective control points $K$ obtained from pattern alteration, the full coordinates of set $U$ can be restored via a series of matrix multiplication. The transformation matrices are precomputed once and reused throughout the optimization. The algorithmic description of the symmetry detection is provided in Algorithm 2 of Appendix.

**Sewn garment shape.** The initial 3D sewn garment mesh is made of 2D pattern placed in 3D, topologically stitched along seams, and geometrically deformed to have sufficiently large inter-panel distances in order to avoid any potential body-garment interpenetration. Note that the vertices along the seams will be merged with their seam counterparts on other panel(s) during stitching. Hence, the correspondence between 3D nodes $x \in X$ and 2D vertices $u \in U$ is one-to-many for those on the seams, while it remains one-to-one for the rest.

**Body model.** The draped shape of a garment is determined by not only the pattern shape $U$, but also the underlying body $B$, and their interaction during contact. We adopt the parametric SMPL model Loper et al. (2015) to represent the body, for which several efficient registration methods to 3D data exist. We used the method by Bhatnagar et al. (2020a,b) to fit SMPL parameters to the wearer's body. The resulting model is denoted as $B = SMPL(\beta^*, \theta^*)$, $\beta$ and $\theta$ are respectively the shape and pose parameters in SMPL, as described in Section 2.6.

### 3.2.3 Pattern linear grading

In this phase, we aim to perform an approximate geometric deformation at the panel level to capture the overall geometry of the target garment, such as length and proportion. The main idea is to match closely corresponding 3D open contours on both garments by deriving the relocation of their corresponding 2D open curves (See Figure 3.3 for an example). A 3D open contour is composed of edges connected to only one adjacent triangle, which often carries design information, representing features such as necklines, hem contours, cuff contours, etc. Given a base model pair $U_{base}$ and $X_{base}$, an initial draped shape $X_{init}$ is computed on the estimated body $B$, with reference to $U_{base}$. The open contours on both the simulated and the target meshes are extracted, associated with their respective counterparts, and the distances between them are measured along the skeleton of the underlying body. These longitudinal distances, together with the difference in circumference, are used to guide the relocation of control points $\{c^i\}$ on the open curves and others in the pattern. An algorithmic description is given in Algorithm 3 of the Appendix.

In Figure 3.4, we illustrate an example of how the measurements on the cuff in 3D are used in the editing of 2D panel curves. The distance between the target cuff $O_t$ and the source $O_s$ measured along the arm bones, together with the difference in their circumferences, determine the amount of the displacement $|\vec{d}|$ of control points on the corresponding open curve (in blue) on the panel. The direction of $\vec{d}$ is derived by computing and nor-

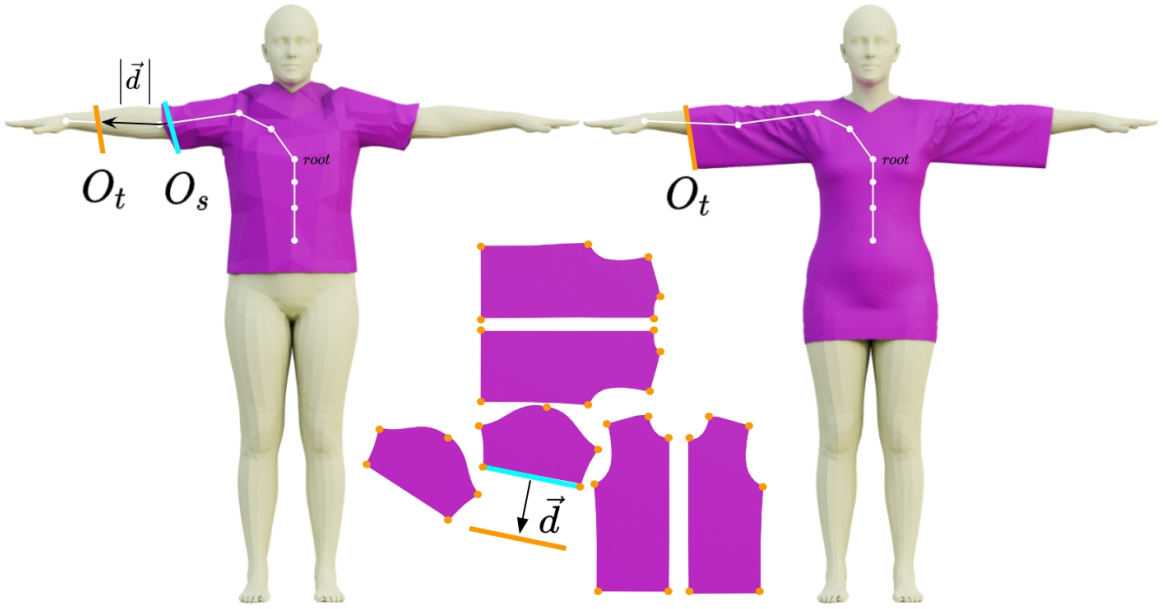


**Figure 3.4:** An example of linear grading. The axial distance $\vec{d}$ between open contours from the source mesh (left) to the target mesh (right) is used to displace the corresponding open curve (in blue) in the 2D panel.

malizing the midpoint of the two endpoints of the open curve minus the average of other control points, while the width-changing vectors are computed by subtracting each control point from the other and normalizing the resulting vectors individually.

The above process could potentially lead to substantial location changes of control points, leading to undesirable topological distortion such as fold-over. To preserve the initial topology of the pattern mesh as well as the neighboring relationship among vertices, we employ the 2D deformation method based on Mean Value Coordinates (MVC), similar to Meng et al. (2012). After the control points change their positions, the positions of other boundary vertices are updated in a per-curve manner, by repositioning each of them in the same relative location in a local coordinates system whose principal axis is defined by a vector connecting two end (control) points of the curve.

To maintain the neighboring relationship of the boundary vertices $v_j^i$, we employ a 2D editing method to first calculate its relative positions in the local coordinates system by a simple linear interpolation to its neighboring control points. The superscript $i$ and subscript $j$ used here refer to the $i$-th pair of two control points($c^i$ and $c^{i+1}$) and the $j$-th boundary vertex between them respectively.

$$\alpha_j = \frac{\left(v_j^i - c^i\right) \cdot x^i}{\|x^i\|}, \beta_j = \frac{\left(v_j^i - c^i\right) \cdot y^i}{\|y^i\|}, \tag{3.1}$$

where $x^i = c^{i+1} - c^i$ and $y^i = R_{90} * x^i$ forms the two axes of the local coordinate system, $R_{90}$ is the rotation matrix for a 90°counter-clockwise rotation . Basically, after the control points are dragged to the desired position $c^{i\prime}, c^{i+1\prime}$, they will serve as new position constraints on the pattern boundary curves. By keeping the relative position of the boundary vertices to its edited neighboring control points as before, the new coordinates of $v_j^{i\prime}$ are updated to: $v_j^{i\prime} = c^{i\prime} + \alpha_j * x^{i\prime} + \beta_j * y^{i\prime}$.

The remaining interior vertices are then updated iteratively with reference to the deformed boundary position constraints by MVC. The edit of 2D panels is followed by a remeshing and draping process to have a simulated 3D garment, to reflect the change also in 3D. The resulting pattern $U_{LG}$ and its corresponding draped garment mesh $X_{LG}$ serve as a good initial state for the subsequent optimization-driven pattern alteration, which is described in Section 3.2.5.

### 3.2.4 Inverse garment simulation

The garment-pattern result obtained from the previous phase is only an approximation of the target geometry. In the next phase, we further refine the pattern $U_{LG}$ through an optimization tightly coupled with a differentiable cloth simulation. Specifically, we extend the differentiable ARCSim Liang et al. (2019); Narain et al. (2012) by revisiting both the dynamics solve and body-cloth interaction.

#### 3.2.4.1 Differentiable cloth simulation

At each forward simulation step, the draping garment over the estimated body is computed, taking into account external and internal forces until an equilibrium is achieved. The implicit Euler integration involves solving a linear system for the cloth motion as presented in Equation (2.16), we rewrite here as:

$$(M - \Delta t^2 J)\Delta v = \Delta t(f + vJ\Delta t), \tag{3.2}$$

where $f$ is the sum of external forces (gravity, contact force) and internal forces (stretching, bending, etc). $M$ is the block diagonal mass matrix composed of the lumped mass of each node, and $J = \frac{\partial f}{\partial x}$ is the Jacobian of the forces. At each time step $\Delta t$, we solve eq. (3.2) for $\Delta v$ and update the velocity $v$ and position $x$. The equation could be written as $\hat{\mathbf{M}}\mathbf{a} = \hat{\mathbf{f}}$ for simplicity.

After the forward simulation with a predefined number of time steps (10 to 20 in our experiments), a loss $\mathscr{L}$ (Section 3.2.5) is measured between the simulated garment geometry and the target cloth mesh segmented from the 3D input. The error is used to back-propagate gradients to optimize the garment rest shape in terms of pattern parameters.

The analytical derivatives $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{M}}}$ and $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{f}}}$ of the linear solver computed in Section 2.7.6 are used to create the new custom "linear solve" operation to PyTorch, and to have it behave like PyTorch's built-in operators with the gradients $\frac{\partial \mathscr{L}}{\partial \mathbf{a}}$ backpropagated from $\mathscr{L}$.

#### 3.2.4.2 Material model

We employ an orthotropic stretching model to quantify the extent of planar internal forces in response to cloth deformation. The model defines the relation between stress $\sigma$ and strain $\epsilon$ using a constant stiffness matrix $\boldsymbol{H}$: $\sigma = \boldsymbol{H}\epsilon$, where

$$\boldsymbol{H} = \begin{bmatrix} H_{00} & H_{01} & 0 \\ H_{01} & H_{11} & 0 \\ 0 & 0 & H_{22} \end{bmatrix}. \tag{3.3}$$

The bending forces are modeled with piecewise dihedral angles Figure 3.5 which describe how much the out-of-plane forces would be when subject to cloth bending, as used in Bridson et al. (2005):

$$f_i = k \frac{\|e\|^2}{\|A_1\| + \|A_2\|} \sin\left(\frac{\pi - \alpha}{2}\right) \mathbf{u}_i, \tag{3.4}$$

where $\alpha$ is the dihedral angle, $e$ is the edge vector, $f_i$ represents the bending force applied on the $i$-th vertex ($i$=1,...,4), $A_1$ and $A_2$ denote the areas of two triangles, $\mathbf{u}_i$ is the direction vector of the $i$-th node, and $k$ is the bending stiffness coefficient.

#### 3.2.4.3 Acceleration of force vector/Jacobian matrix assembly

ARCSim Liang et al. (2019); Narain et al. (2012) uses the traditional approach of directly solving the linear system after the assembly of the extended mass matrix $\hat{\mathbf{M}}$ and the force vector $\hat{\mathbf{f}}$. The internal forces exerted by a triangle element to its nodes are split and accumulated to the global force vector, where the contributions from multiple adjacent elements are summed up for each node. Such force vector assembly process incurs a considerable overhead cost as the number of time steps grows. It is even more expensive for the

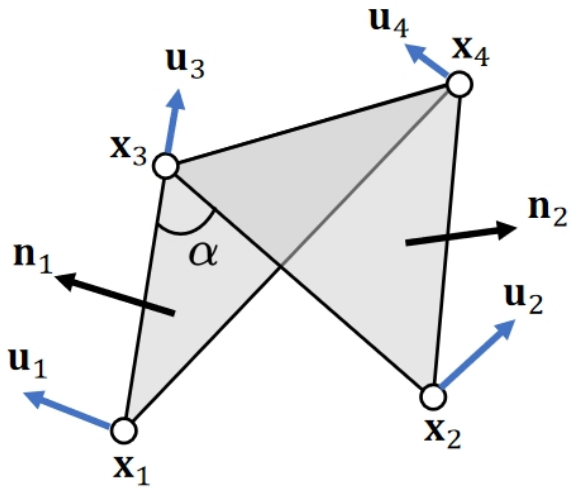


**Figure 3.5:** Dihedral Angle Model.

extended mass matrix assembly as it contains the Jacobian of forces, which is large and sparse. We propose an efficient method for accelerating the assembly. As the same assembly is executed for each time integration, we exploit the fact that the inherent topological structure remains unchanged during the simulation, with a sequence of triangle elements in a fixed filling order. We encode this information in the form of a static mapping matrix, which converts the assembly process to a matrix multiplication, which is parallelizable on a GPU. Additionally, the matrix remains very sparse regardless of the mesh resolution, for which multiple numerical tools are available.

In particular, the matrix $\hat{\mathbf{M}}$ is constructed by aggregating the local contributions from individual elements into the corresponding locations. Each triangle element (element hereafter) yields a Jacobian matrix of nine partial derivatives of the force with respect to the position of a node ($\frac{\partial f_i}{\partial x_j}$), $i, j$=1,2,3. The Jacobians for all elements are packed into a *Jacobian stack* as illustrated in Figure 3.6 (e), where we use $mn$ to denote $\frac{\partial f_m}{\partial x_n}$ ($m$, $n$: global indices) for a compact representation.

In ARCSim Liang et al. (2019); Narain et al. (2012), these Jacobians are assembled to $\hat{\mathbf{M}}$ through a total of $F \times 3 \times 3$ assignment or addition operations, where $F$ represents the number of triangles in the garment mesh. Optimizing this process becomes crucial, especially considering its higher computational cost compared to the force vector assembly. To this end, we propose to realize the Jacobian assembly using matrix multiplication. $\hat{\mathbf{M}}$ is a sparse matrix, and moreover, directly representing the mapping from the Jacobian stack to the matrix form is not feasible, although it would be ideal for leveraging GPU-accelerated matrix multiplication. We address this issue by introducing an intermediate

data structure called *compressed Jacobian vector* (Figure 3.6(b)). It is a set of Jacobians for each force-node combination, which is obtained by first reshaping the Jacobian stack into a Jacobian vector (Figure 3.6(d)), and by encoding the mapping from the per-element Jacobian to the compressed Jacobian vector as a static mapping matrix (Figure 3.6(c)). Then a GPU-based sparse matrix multiplication is performed, effectively substituting the iteration-based Jacobian assembly. It is highly backpropagation-friendly, resulting in a considerable acceleration of the assembly process of linear solve(Table 3.1). Finally, the Jacobians in the compressed vector are transferred to $\hat{\mathbf{M}}$ (Figure 3.6(a)). The number of operations reduces to $N+2\times E$ ($N$: number of nodes, $E$: number of edges), compared to the original $F\times 3\times 3$. The assembly of batched elementary forces to the global force vector follows the same principle.

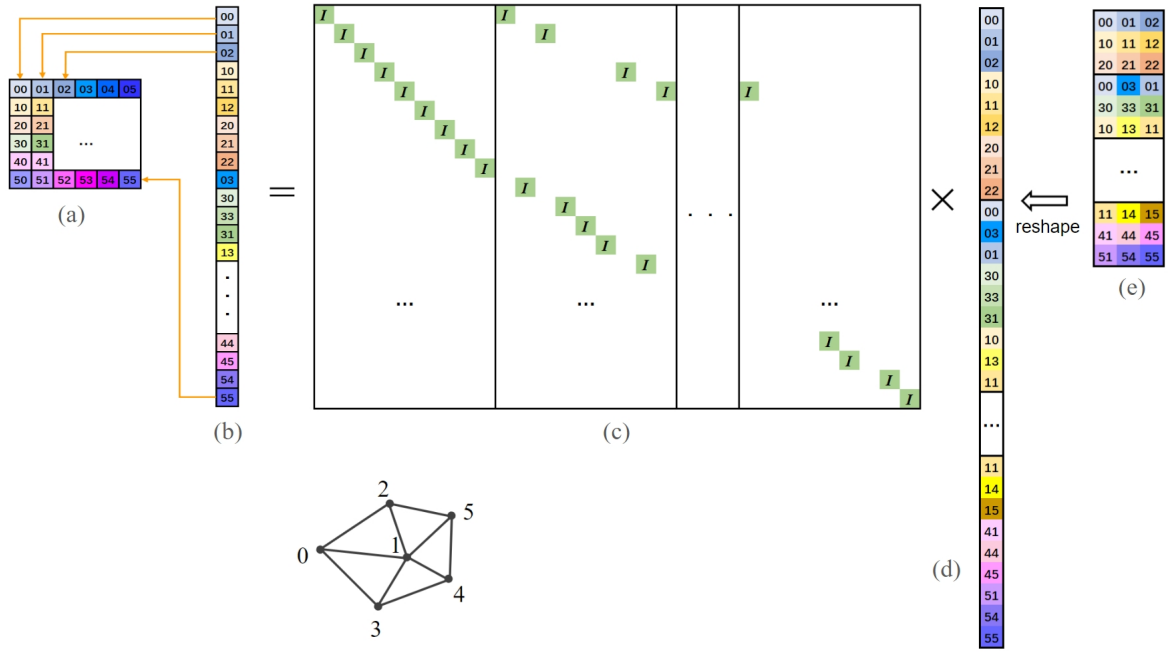


**Figure 3.6:** The assembly of a compressed Jacobian vector (b) is obtained by a static matrix multiplication, encoding the mapping from the per-element Jacobian (e) to the per force-node Jacobian (b). Compressed Jacobians are then transferred to the extended mass matrix (a) .

#### 3.2.4.4 Efficient body cloth interaction

One important component of draping simulation lies in the body-garment interaction, which involves the contact force computation and the garment-body collision handling, for which many algorithms have been proposed Bridson et al. (2005); Harmon et al. (2008); Tang et al. (2010). In particular, collision handling Harmon et al. (2008) consists of two

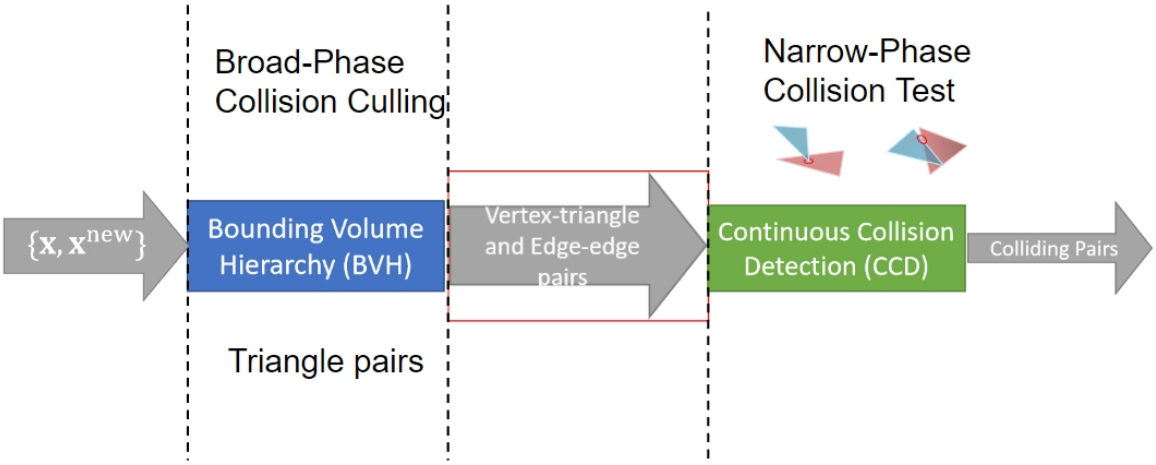


**Figure 3.7:** Collision detection contains two phases: broad-phase culling and narrow-phase test.

main steps: collision detection and collision response. Collision detection is divided into two phases: broad-phase culling and narrow-phase testing. To ensure efficiency, a bounding volume hierarchy (BVH) is employed during the broad-phase culling. This is followed by continuous collision detection to identify all potential collision instances. The BVH accelerates collision testing by organizing simple bounding volumes hierarchically. Initially, collision tests are performed against the top-level bounding boxes, and only if an overlap is detected, the test proceeds to the next level. This process is repeated recursively, down to the triangle level, where further primitive overlapping pairs (e.g., vertex-face and edge-edge) are identified (Figure 3.7 ). In the narrow-phase collision test, the precise moment of collision is determined for the potentially colliding pairs identified in the broad phase. A key characteristic of both tests is that, at the time of collision, the four involved vertices lie on the same plane. And the collision response is conducted on non-rigid impact zones, which has been made differentiable by Liang et al. (2019). However, it remains computationally expensive, leading to rapid growth of the computation graph (i.e. memory-hungry) during forward simulation, and struggles to accommodate high-resolution meshes. Hence, we chose to compromise by implementing a lightweight collision handling scheme that makes use of the signed distance function (SDF).

When the signed distance of a query garment vertex $x$ falls below a threshold (indicating proximity to the body), the repulsion force is triggered between the body and the garment, with its magnitude inversely proportional to their distance. We use the classical Coulomb's model for friction force, which is elicited when there is relative movement along the surface tangent. While the repulsion forces prevent the interpenetration, occasional collisions might still occur and need correction after the dynamic simulation. To this end, for any garment vertex $x$ with $sdf(x) < 0$ we present the collision resolving setup,

correcting the interpenetration by:

$$\tilde{x} = x + (\delta - sdf(x)) \cdot \mathbf{n}, \tag{3.5}$$

where $\mathbf{n} = \frac{\nabla sdf(x)}{|\nabla sdfx|}$ is the spatial gradient of $sdf(\cdot)$ (also the surface normal), and $\delta$ denotes the collision thickness. This scheme significantly enhances the speed of both forward and reverse simulation while maintaining the performance level.

The signed distance of the underlying body is computed either by a trained neural network, which is naturally differentiable, or analytically on the fly with Kaolin NVIDIA (2024). In the latter case, as it involves only unit operations, it can also be seamlessly integrated into the differentiable simulation. Since the SDF is also used to compute the repulsion forces for intersection prevention, the position correction (projection) is generally sparse and minor if there is any. As a result, it ensures smooth backpropagation without introducing discontinuities.

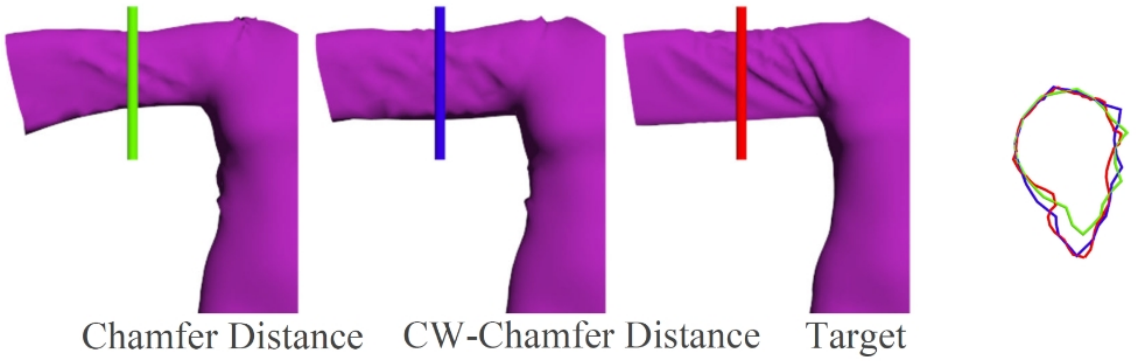


**Figure 3.8:** The cross-sectional curves on the right, generated from three meshes on the left: two simulated meshes, and the target. The colors of the cutting planes are used to draw the cross-sections. The curvature-weighted Chamfer (blue) leads to a sleeve draping silhouette closer to the ground truth (red), compared to the standard Chamfer distance (green).

### 3.2.5 Optimization-based pattern alteration

In this phase, we further refine both the pattern state $U_{LG}$ and the simulated garment $X_{LG}$ obtained from the previous stage through optimization using the differentiable draping simulator. At each iteration, the simulated garment geometry $X = \{x_i\}$ is compared with the target $T = \{t_i\}$ using a loss function, subsequently utilized by a gradient-based algorithm to refine the pattern shape. We define the following loss over the effective control

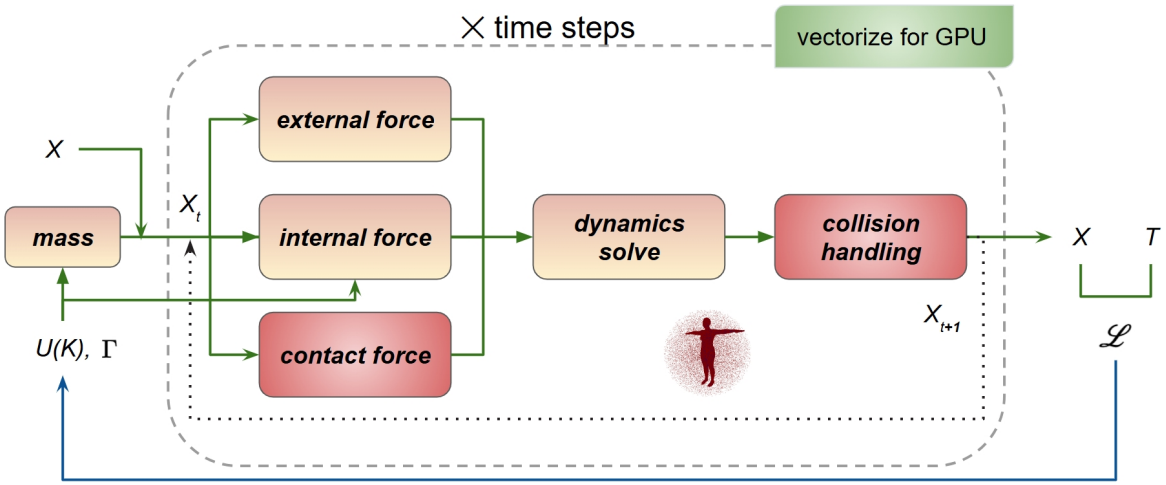


**Figure 3.9:** A schematic diagram of the optimization process with differentiable simulation.

points $K$ and the physical parameters $\Gamma$:

$$\begin{aligned}\mathscr{L} = \mathscr{L}_{rec}(X = Sim(U(K), \Gamma; B(\theta, \beta)), T) \\ + \lambda_{seam}\mathscr{L}_{seam}(U(K)),\end{aligned} \tag{3.6}$$

where $\lambda$s are weights. It combines the reconstruction loss $\mathscr{L}_{rec}$ and the seam-consistency loss $\mathscr{L}_{seam}$ penalizing the inconsistent curve lengths along the seam. The reconstruction error is composed of the Chamfer distances $\mathscr{L}_{CF}$ measured both between the surfaces ($X$ and $T$) and among open contours ($X_{open}$ and $T_{open}$),

$$\mathscr{L}_{rec} = \mathscr{L}_{CF}(X, T) + \lambda_{open}\mathscr{L}_{CF}(X_{open}, T_{open}) + \lambda_{mat}\mathscr{L}_{mat}(\Gamma). \tag{3.7}$$

The Chamfer distance, widely used for fitting deformable surfaces, has proven to work well in most cases.

$$\mathscr{L}_{CF}(X, T) = \frac{1}{|X|}\sum_{x\in X}\min_{t\in T}\|x - t\|^2 + \frac{1}{|T|}\sum_{t\in T}\min_{x\in X}\|t - x\|^2. \tag{3.8}$$

However, we observed that it is not sufficient for certain garment targets, due to its "myopia" that each point only considers its nearest neighbor on the other mesh, neglecting the surroundings. In regions with high curvatures, often present in the folded geometry of loose clothes, this can lead to lower geometric accuracy (See Figure 3.8). Hence, we use curvature-weighted Chamfer distance Bongratz et al. (2022) instead, which prioritizes high-curvature regions, subsequently improving the reconstruction of densely folded re-

gions:

$$\begin{aligned}\mathscr{L}_{CWCF}(X,T) = &\frac{1}{|X|}\sum_{x\in X}\kappa(\tilde{t})\min_{t\in T}\|x-t\|^2 \\ &+\frac{1}{|T|}\sum_{t\in T}\kappa(t)\min_{x\in X}\|t-x\|^2,\end{aligned} \tag{3.9}$$

with $\kappa$ the mean curvature and $\tilde{t} = \arg\min_{t\in T}(\|x-t\|)$. The seam loss serves as the regularization that guarantees the consistent curve lengths of two panels along the seam:

$$\mathscr{L}_{seam}(U) = \sum_{i}^{|S|}\sum_{j}^{|E_i|=|E_i^{corr}|}\left(|e_i^j| - |e_i^{corr,j}|\right)^2, \tag{3.10}$$

where $S = \{S_i\} \subset X$ denotes the set of 3D seam curves, and $E_i, E_i^{corr} \subset U$ are the sets of edges $e^j = u^j - u^{j+1}$ along the 2D panel curves comprising the seam counterparts of $S_i$.

We also optimize over the physical parameters $\boldsymbol{H}$ and $k$ by adding them into the variable set:

$$\Gamma := (H_{00}, H_{01}, H_{11}, H_{22}, k). \tag{3.11}$$

To penalize unrealistic material parameter combinations, we constrained the elements within the physically plausible ranges(above a non-negative threshold 1e-6), with $\mathscr{L}_{mat} = \text{relu}(1e^{-6} - \Gamma)$. These physical parameters are added to the variable set in later iterations, once the pattern shapes reach an approximate optimum.

## 3.3 Evaluation

### 3.3.1 Implementation details

We now describe the main implementation details and experiment setups.

**Simulation.** We set one time step $\Delta t$ to 0.05s, and the number of time steps for one forward simulation between 10 and 20. The garment resolution tested ranges from 1K to 8K vertices. We set the collision thickness to the conventional value 1e-3m. A variant of DeepSDF Park et al. (2019) with periodic activation functions Sitzmann et al. (2020) that learns the SDF of the wearer's body has been used for collision handling. By vectorizing as much as possible the force and Jacobian computation, our extension to ARCSim differentiable cloth simulator Liang et al. (2019) allows it to run all computations on a GPU. Table 3.1 summarizes the computation time of linear solve by the baseline model Liang et al. (2019) and ours, measured on a NVIDIA GeForce 3090, for T-shirt garments

with different resolutions (1K to 3K). Note that in our model the first iteration involves the construction of the deterministic mapping, which is reused in subsequent iterations. In contrast, the baseline model requires the overhead of assembly for every iteration. The acceleration of the linear solver and the SDF-based collision handling scheme result in a 14 times overall speedup compared to Liang et al. (2019).

| | Baseline Liang et al. (2019) | | Ours | | |
|---|---|---|---|---|---|
| # vertsiterations | 1 | 20 | 1 | 20 | speedup |
| 1K (forward) | 2.8 | 83.7 | 19.1 | 59.4 | 1.4x |
| 1K (reverse) | 7.0 | 155.6 | 2.7 | 60.0 | 2.59x |
| 2K (forward) | 8.4 | 178.1 | 67.6 | 133.0 | 1.34x |
| 2K (reverse) | 17.2 | 466.0 | 6.3 | 128.7 | 3.62x |
| 3K (forward) | 12.3 | 257.9 | 133.6 | 228.6 | 1.13x |
| 3K (reverse) | 24.7 | 505.4 | 12.0 | 186.3 | 2.71x |

**Table 3.1:** The computational time for the matrix assembly and linear solve (in seconds) using the baseline model and ours, measured on a T-shirt garment with varying resolutions. Note how our method improves the speed as the number of iterations increases, especially during the reverse process.

**Optimization.** The variables are optimized by using the Adam optimizer Kingma et Ba (2015), with a learning rate $10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We empirically set $\lambda_{open} = 0.1$, $\lambda_{mat} = 0.01$ and $\lambda_{seam} = 0.01$. We observed that the addition of material parameters noticeably impacts the result. Conversely, excluding these parameters from optimization might lead to distorted panels. We present the results of a related ablation study in the following section. The material parameters are initialized with a material set selected by the user from the material library, which spans a range of common fabrics. Additionally, the pattern variables are effectively initialized through linear grading.

**Pattern editor GUI.** To allow for customization of template patterns for special designs(e.g. deep V collar) and eventually support more garment categories. We developed a graphic user interface(GUI) that allows a user to input and modify a pre-existing clothing pattern. The user can mouse over the 2d pattern to see its correspondence in a 3d garment, and can move and rotate the panels of the pattern. They can select any "control points" (corner points or other points integral to the shape of the panel) and drag them or move them with the arrow keys to change the shape of a panel. They can also select two points in the pattern and press the "straighten curve" button to straighten the curve between them. The user can superimpose different layers over the pattern mesh, one to see all the possible control points and add or remove them, one to show with color coding the "inter-panel symmetry" in the panel (symmetry between panels), and another to show

the intra-panel symmetry (symmetry within a panel itself). More GUI can be found in the Figure 5.1.

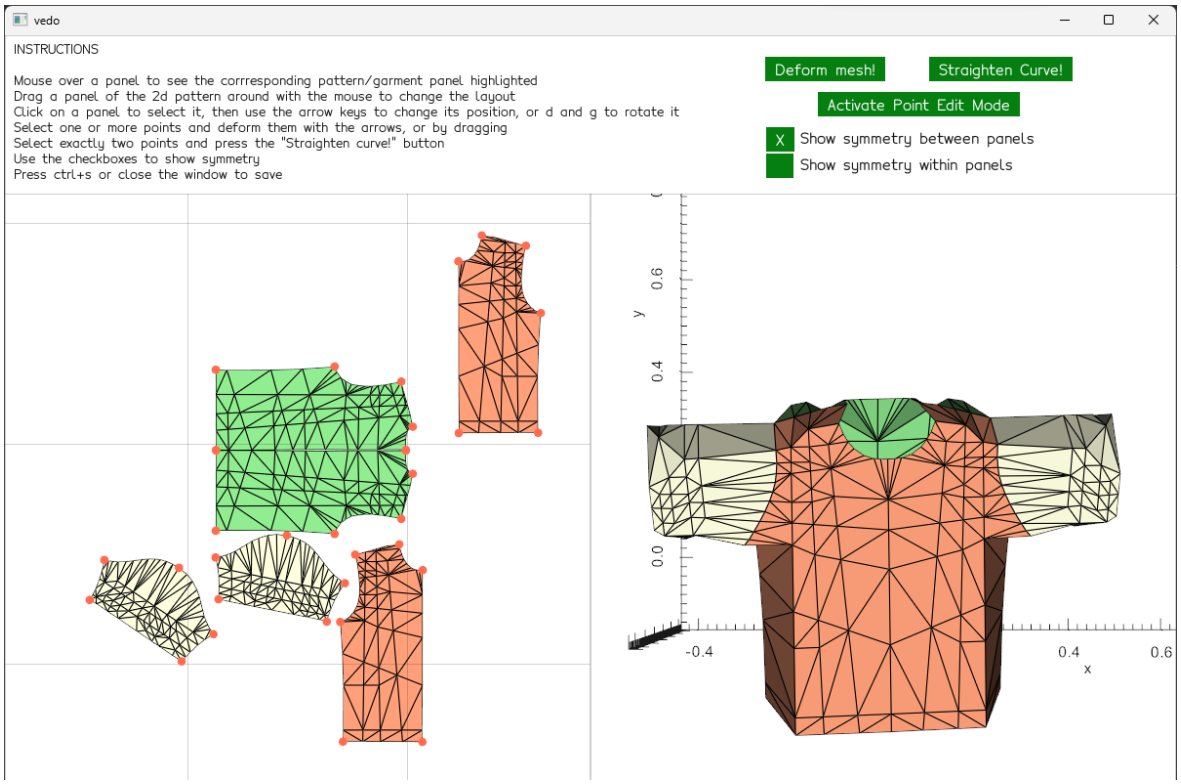


**Figure 3.10:** Graphic user interface for pattern editing.

### 3.3.2 Quantitative and qualitative comparisons

We evaluate our model on a number of representative garment types and compare it with previous works. We first evaluate the performance on 3D garment reconstruction, and then on 2D pattern estimation. To carry out a fair comparison, we use the garment meshes of a third-party LAVA lab dataset Korosteleva et Lee (2021) as targets (i.e. test data), which is an unseen dataset for all evaluated methods. It is also one of the few datasets that provide 2D sewing patterns for every 3D garment mesh, enabling the evaluation of our results in both 3D reconstruction and 2D pattern estimation. We describe the detailed results below.

**3D garment reconstruction.** We compare our approach to the related methods for garment fitting, and utilize 3D garment geometry as targets. We run the Adam optimizer Kingma et Ba (2015) for a varying number of iterations until the convergence for each method, while noting that the parameter space representing the garment geometry differs among them: The coordinates of effective control points and material parameters for

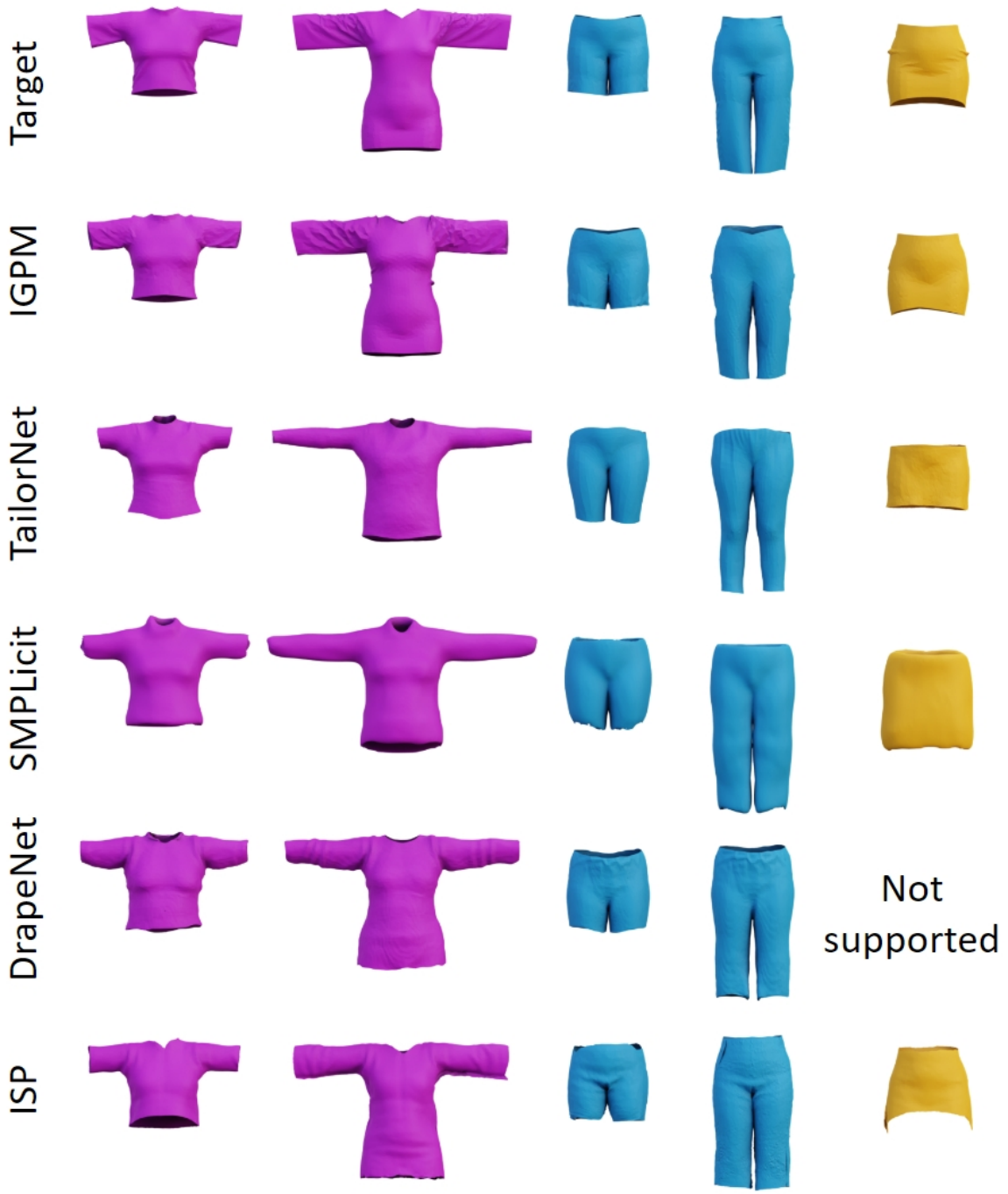


**Figure 3.11:** Comparison of 3D garment reconstruction of our method with others Corona et al. (2021); De Luigi et al. (2023); Li et al. (2024b); Patel et al. (2020). IGPM reproduces faithful garment shapes, even accounting for intricate geometry details like wrinkles on large sleeves. Best viewed on screen zoomed-in.

our method IGPM, the $\gamma$ garment style parameter for TailorNet Patel et al. (2020), the latent vector $z = [z_{cut}, z_{style}]$ describing garment cut and style for SMPLicit Corona et al. (2021), and the latent codes $z$'s encoding the garment characteristics in DrapeNet and ISP De Luigi et al. (2023); Li et al. (2024b). The results are quantitatively evaluated using two metrics: Chamfer distance(i.e., the averaged squared L2-distance from each vertex in one mesh to its closest vertex in the other mesh, computed bidirectionally.) to the ground truth mesh vertices, and the angular error to measure the similarity of the computed normal vectors, similar to Bednarik et al. (2020). As shown in Table 3.2, our method is consistently better than others, which is confirmed by qualitative results, shown in Figure 3.11. We observe that the performance of the data-driven approaches is biased by the training dataset. It is clear that TailorNet basically has learned over tight-fit datasets, so it does not generalize very well when fitting to loose styles, as seen in the Pants example. In contrast, our approach reconstructs accurate 3D geometry, for both loose and tight garments. Drapenet uses unsigned distance functions (UDFs) and requires extra computations for meshing, and it is sensitive to the initialization of latent code for optimization.

| | 3D Reconstruction | | | | |
|---|---|---|---|---|---|
| | Chamfer distance / Normal similarity | | | | |
| Garments | SMPLicit | TailorNet | Drapenet | ISP | Ours |
| T-shirt | 1.4 /- | 0.331 / 0.081 | 0.689 / 0.129 | 0.297 / 0.094 | **0.112** / **0.049** |
| Dress | 3.2 / - | 1.305 / 0.161 | 0.619 / 0.135 | 0.189 / 0.131 | **0.110** / **0.075** |
| Shorts | 1.3 / - | 1.036 / 0.050 | 0.131 / 0.048 | 0.202 / 0.095 | **0.126** / **0.043** |
| Pants | 2.9 / - | 2.587 / 0.104 | 0.485 / 0.085 | 0.185 / 0.077 | **0.142** / **0.049** |
| Skirt | 6.5 / - | 1.300 / 0.063 | - / - | 0.435 / 0.093 | **0.106** / **0.014** |

**Table 3.2:** Quantitative evaluation in 3D garment reconstruction. we use Chamfer distance and cosine normal similarity to measure geometric similarity. Our method performs notably better than others.

**2D Pattern Estimation**. The quantity of research focusing on sewing pattern recovery directly from a given 3D input data is rather limited, with the majority of them dedicated to precise but minor adjustments to existing patterns Bartle et al. (2016); Wang (2018). We compare our work with NeuralTailor Korosteleva et Lee (2022) and SewFormer Liu et al. (2023), two deep learning frameworks that predict a structural representation of a sewing pattern from a 3D point cloud and a 2D image, respectively. To facilitate comparison with the ground-truth patterns, the experiments were conducted under favorable conditions for their work – We selected five patterns from the dataset used in NeuralTailor as the ground-truth ones. Then, we generate 3D drape shapes at a T-pose by using an independent simulator, differing from both ours and theirs. To evaluate SewFormer, we render 2D images with settings similar to those used by SewFormer. Some of the results are illus-

| | 2D Estimation | | |
|---|---|---|---|
| | Turning /Surface area | | |
| Garments | NeuralTailor | SewFormer | Ours |
| T-shirt | 10.50 / 0.13 | 10.46 / 0.15 | **9.12** / **0.09** |
| Dress | 11.20 / 0.37 | 11.77 / 0.43 | **10.96** / **0.10** |
| Shorts | 7.41 / 0.05 | 7.74 / 0.06 | **7.33** / **0.04** |
| Pants | **6.77** / **0.01** | 6.79 / 0.20 | 7.00 / 0.08 |
| Skirt | 4.31 / 1.11 | 4.42 / 0.62 | **4.14** / **0.04** |

**Table 3.3:** Quantitative evaluation in 2D pattern estimation. To measure the accuracy of 2D patterns, we evaluate the turning function metric for comparing polygonal shapes Arkin et al. (1991) and the surface error (the average of normalized surface difference error computed for each panel).

trated in Figure 3.12. We observe that their method makes very good predictions on the trouser-like garments as the geometric variation of pants and shorts are limited and well covered in their training dataset. For the other garment types, however, our method produces better results. To quantitatively measure the quality of estimated 2D patterns, we have used two metrics: (1) the turning function metric for comparing polygonal shapes Arkin et al. (1991), and (2) the relative error in surface area $A(\cdot)$, as determined by averaging normalized surface difference error $\frac{1}{|p|}\sum\frac{\Delta A(p_i)}{A(p_i)}$ computed for each panel $p_i$.

**Generalization to different poses and shapes.** Target garment shapes draped on a T-posed body allow for a fair comparison to NeuralTailor, since it has been trained on garments in this setting. To demonstrate that our method also performs well in other settings, we have tested our method on some example meshes from SewFormer Liu et al. (2023). As shown in Figure 3.13, our method is able to faithfully reconstruct 3D garments worn by the individuals in challenging poses, while producing consistent patterns close to the ground truth.

### 3.3.3 Recovery of physical parameters

To demonstrate the capability of our method to faithfully recover physics, two draping skirt meshes were simulated using identical sewing patterns but varying only the physical parameters. Then we used them as targets and compared our estimated patterns with those generated from NeuralTailor Korosteleva et Lee (2022). As shown in Figure 3.14, our method can faithfully capture 3D garment geometric variations originating from different bending parameters, while producing consistent patterns close to the ground truth. On the contrary, NeuralTailor translates the geometric variation into that of panels, yielding

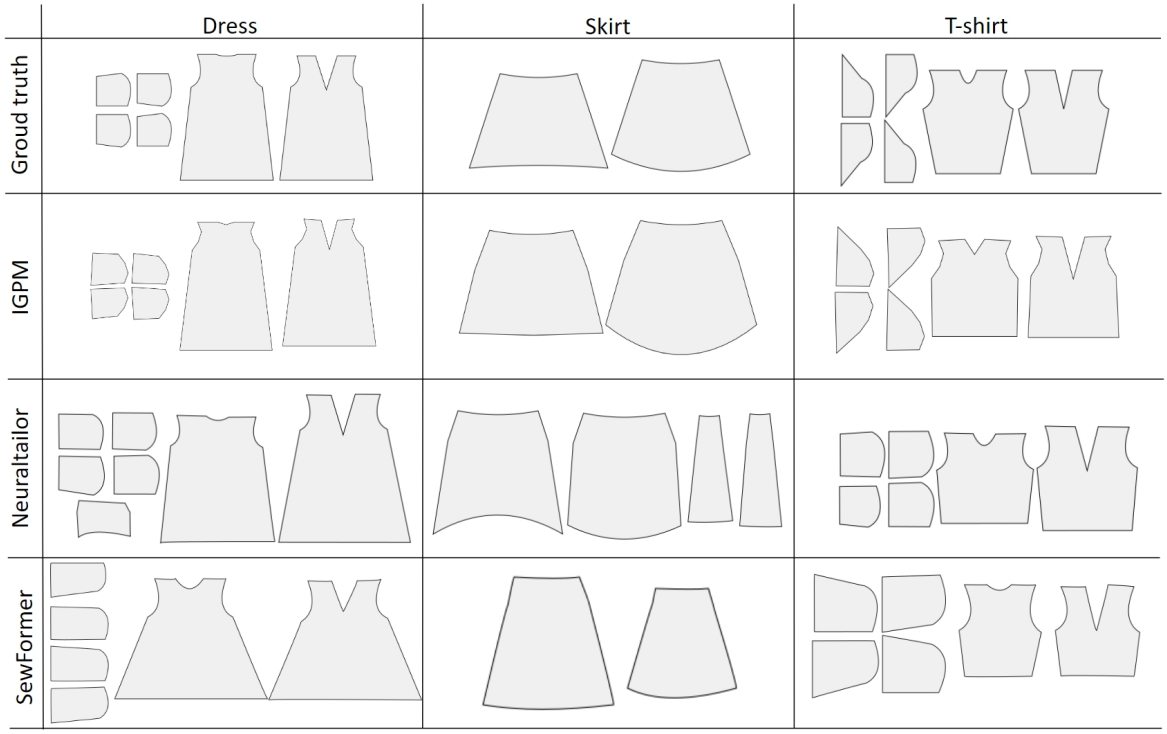


**Figure 3.12:** 2D pattern estimation from 3D garment meshes. From top to bottom: Ground truth, IGPM, NeuralTailor Korosteleva et Lee (2022) and SewFormer Liu et al. (2023).

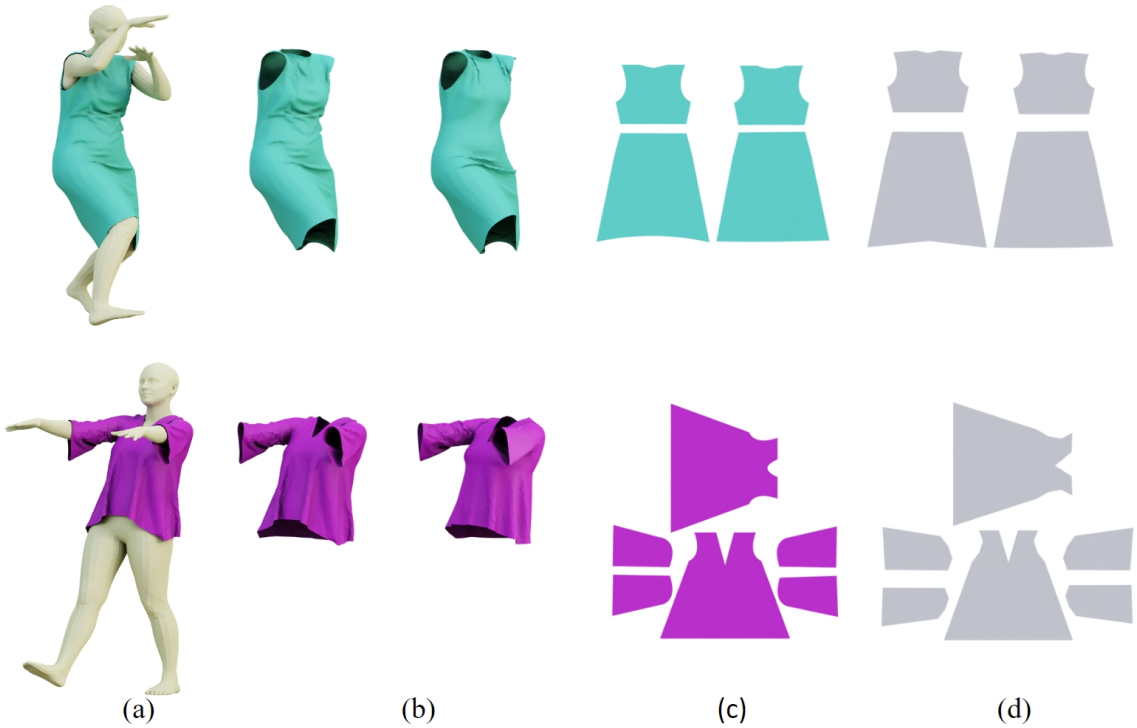


**Figure 3.13:** Results of our method evaluated using varying poses. (a) Target clothed body; (b) target garment mesh (left) and the reconstructed mesh (right); (c) ground-truth pattern; and (d) our estimated pattern.

a significantly different pattern for each target instance.

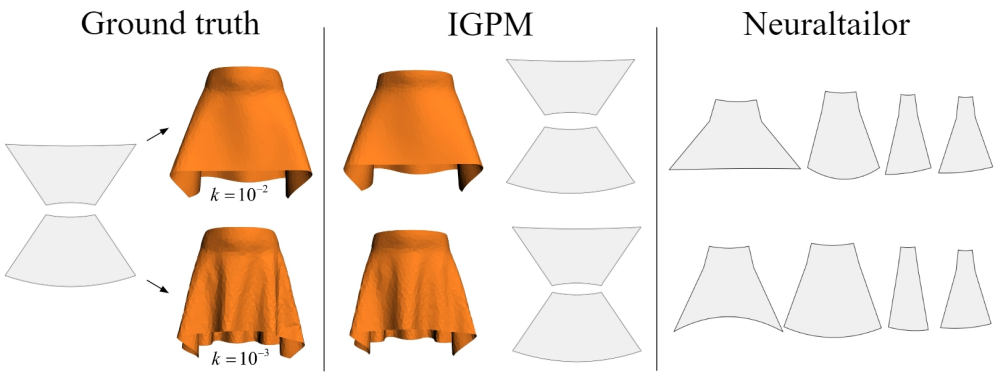


**Figure 3.14:** Sewing patterns estimated from two input meshes, both simulated from an identical ground-truth pattern but with varying bending coefficients. By optimization over bending coefficient, our method correctly finds the panel shapes, compared to the alternative method.

### 3.3.4 Evaluation on 3D scan data and retargeting

We conducted a qualitative evaluation of our method using 3D scans obtained with the Vitronic VITUS Human Solutions body scanner VITRONIC Corporation (2024), the captured meshes are of high quality and without holes. The ground-truth patterns have been obtained by placing transparent papers over the flattened garments, drawing along the seams, and then digitally cutting along the traced lines after scanning. As shown in Figure 3.15, our method outputs reasonable, quality estimations of the 3D garment and the 2D pattern. Since the recovered pattern is simulation compatible, it can be easily reused by a simulator to generate draping shapes on new conditions, as shown in Figure 3.16. More results about the reanimation are provided in Figure 3.17.

## 3.4 Ablation study

Here we report the results of our ablation study, where we examine the contributions of individual components to the overall performance (Table 3.4). Our model achieved a Chamfer distance precision (CF) of 0.1103 and a cosine distance of normals (NOR) of 0.075. It outperforms other settings where the curvature-weighted Chamfer loss is replaced with the vanilla Chamfer (1st row of Table 3.4), when the seam consistency loss term is removed (2nd row), or when the optimization of physical parameters was disabled

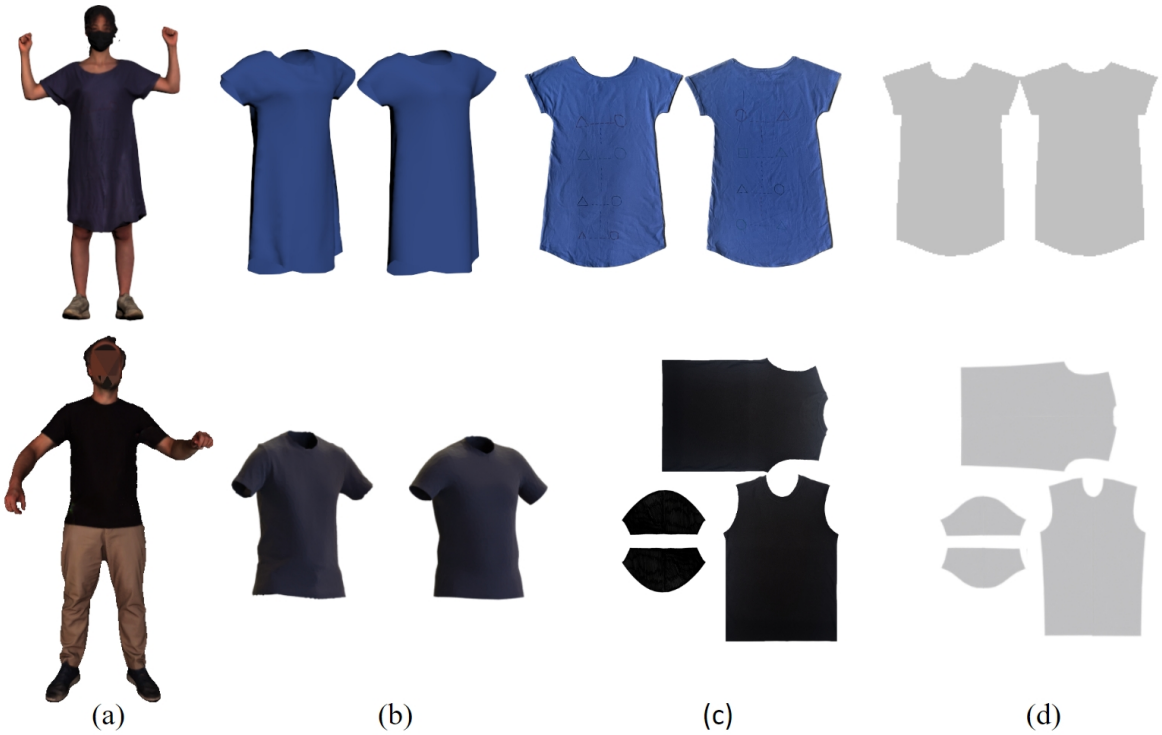


**Figure 3.15:** Results of our method evaluated using 3D scan data. (a) Input 3D scan; (b) segmented target (left) and simulated garments (right); (c) ground-truth pattern; and (d) estimated pattern.

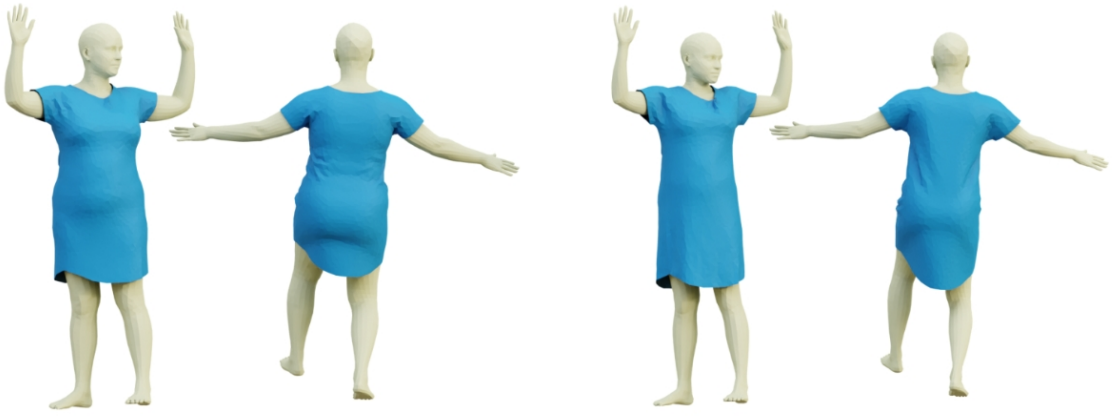

**Figure 3.16:** The dress model obtained from the 3D scan (Figure 3.15) has been retargated to two new SMPL bodies.

**Figure 3.17:** The recovered garments have been reanimated to new body motions bodies for dynamics.

(3rd row). These results confirm the importance of both loss terms and the integration of physical properties in the optimization process.

Figure 3.18 illustrates the reconstructed models obtained from the ablation study. We observe that our model (Figure 3.18(e)) bears the closest visual resemblance to the target. The use of weighted Chamfer distance allows for better capture of the armpit region and the lower part of the sleeve (Figure 3.18(b)). The absence of seam loss leads to a puckered seam around the shoulder, resulting from the extra tension exerted on the shorter seam (Figure 3.18(c)). The optimization of physical parameters helps to recover fine wrinkles, as well as more plausible pattern estimation. As we can see in Figure 3.18(d), the "pear shape" of the body/dress has been solely attributed to the increasing panel width along the torso, when the material parameters were disregarded during the optimization.

| Method | Chamfer distance (CF) | Normal similarity |
|---|---|---|
| Ours(w/o curvature CF) | 0.115 | 0.085 |
| Ours(w/o seam loss) | 0.117 | 0.076 |
| Ours(w/o physics) | 0.113 | 0.081 |
| Ours | **0.110** | **0.075** |

**Table 3.4:** Quantitative results of ablation studies. We report the metrics for the dress garment.

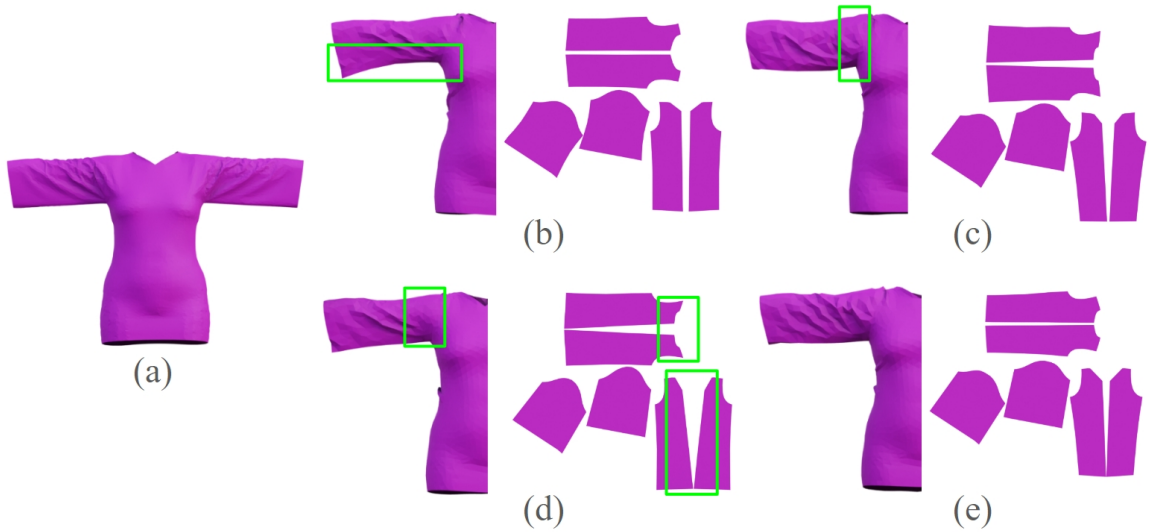


**Figure 3.18:** Results of our ablation study on loss terms: (a) target mesh; (b) results with vanilla Chamfer loss (without curvature weights, (c) without seam loss, and (d) without physical parameters; (e) our results.

## 3.5 Conclusion

We have presented a method to recover simulation-ready garment models from a given 3D geometry of a dressed person. Basing our work on a differentiable simulator, we refine the 2D sewing pattern shape through inverse simulation, ensuring that the physically based draping of the corresponding sewn garment closely matches the given target. Our experimental results confirm that our system can produce simulation- and fabrication-ready patterns on a range of representative garment geometries, outperforming comparable state-of-the-art methods.

Our approach presents several limitations that suggest avenues for future exploration. First, the iterative optimization process involving forward and reverse simulation is time-consuming. Further acceleration can be employed to achieve faster convergence in optimization processes Jang et al. (2023). Second, although our linear grading scheme effectively adjusts the base model to align with the target prior to the optimization, the results can be sensitive to initial values, potentially resulting in different local minima. As well, predefined design choices such as mesh resolution and the identification of control points on the panels can also impact the outcome.

CHAPTER 4

# PHYDEFORMER: HIGH-QUALITY NON-RIGID GARMENT REGISTRATION WITH PHYSICS-AWARENESS

## 4.1 Introduction

This chapter introduces a mesh deformation-based method designed to generate detailed garment geometry that replicates target clothing. One approach for addressing garment reconstruction is the use of full-body reconstruction techniques Habermann et al. (2020); Zhang et al. (2017). However, the resulting geometries are typically watertight and coarse, necessitating manual post-processing, such as segmenting for open contours of neck, arms, legs, and waists and re-meshing to modify the mesh topology. Alternatively, using a limited number of view renderings, Neural radiance field(NeRF)-based methods Feng et al. (2022); Wang et al. (2023, 2021) can achieve relatively high reconstruction quality, though their outputs are primarily suitable for rendering are challenging to use for other downstream tasks, and are affected by slow optimization and rendering speed. Another research direction involves fitting garments by adjusting the 2D garment patterns, which can be optimized using differentiable simulation, as in the last chapter. While these approaches Li et al. (2024c); Yu et al. (2024) produce simulation-ready garments, they are computationally expensive as a complete quasi-static simulation is executed at each iteration. Additional setups are required to generalize to new garment types. Moreover, recent developments have reduced the cost and difficulties in obtaining segmented 3D scans Wang et al. (2024), which makes it easier to leverage 3D inputs for research problems and commercial applications. Aligning virtual template garments to 3D data of real clothing can accelerate the recovery and reproduction of 3D garment geometries, with applications across various domains, including gaming, film, fashion, and virtual try-on.

In the previous chapter, 3D garment reconstruction through inverse modeling is studied, where qualitative results indicate that it yields satisfactory outcomes. In this chapter, the focus is more on capturing higher-frequency details, such as the exact folds and wrinkles exhibited on the targets, that are more challenging to recover. Therefore, we would like to propose a faster method with higher quality output geometries, which delves deeper into reproducing finer geometric details for applications with higher precision requirements.

To this end, we proposed PhyDeformer to resolve these challenges from a mesh deformation perspective, which works in a two-stage manner. In the first stage, we conduct the garment grading for a coarse 3D alignment of the template and the target mesh, accounting for the proportional scaling and fit same to Section 3.2.3. In the second stage, the graded garment mesh is further refined with a Jacobians-based deformation by using the reconstruction loss and other physics-inspired losses. We remark that the necessity of refinement of garments is rooted in their highly non-rigid nature in garment geometry.

Our approach stems from performing the mesh deformation directly in the 3D space to eliminate scale and pose ambiguities that are difficult to avoid when using 2D images. In addition, physics-based constraints are also added in the refinement stage, the output is regularized to be physically plausible.

We demonstrate that our two-stage pipeline is capable of smoothly deforming a wide variety of garment meshes, achieving both large modifications, e.g., proportions in sizing, as well as adding fine details, such as folds and wrinkles.

## 4.2 Method

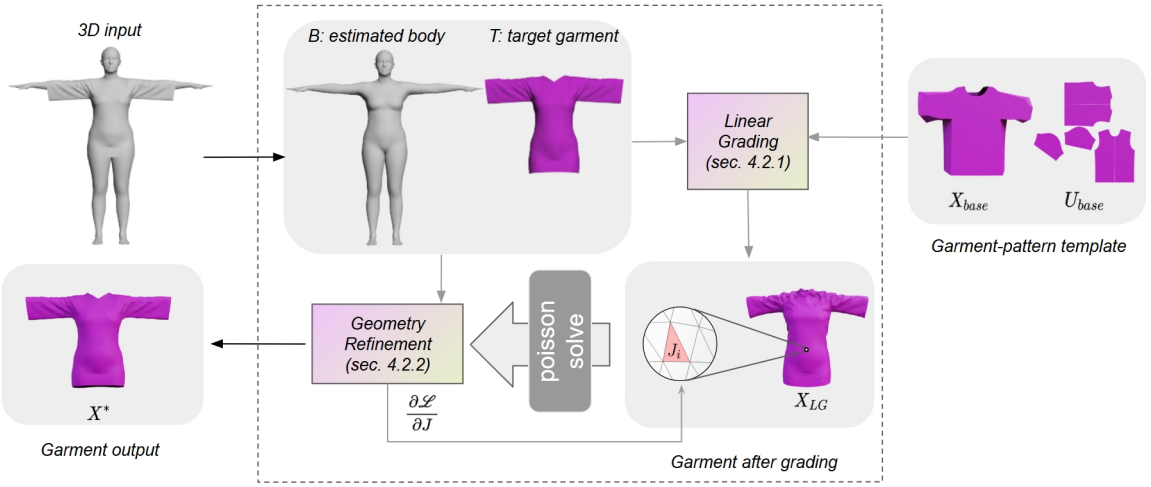


**Figure 4.1:** Given an input 3D base mesh and a target garment image, we perform linear grading on the base mesh to achieve an initial alignment. Then the geometry deforms by optimizing per triangle Jacobians. Two sorts of losses are utilized as a soft supervision signal during the deformation process: reconstruction loss drives the deformation toward the target geometry, and a set of physics losses ensures a physics-coherent deformation. The output geometry accurately captures the geometry of the target.

Figure 4.1 shows an overview of our method. Given an input mesh, PhyDeformer enables the manipulation of geometry in two stages. In the first stage, the linear grading component accounts for capturing the coarse geometry such as size and proportion. Then, in the second refinement step, we optimize a displacement map $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ over the vertices through Jacobians guided by a set of losses. We represent the geometry of the garment shape using a triangular mesh $X$ defined by a set of vertices $x \in \mathbb{R}^3$ and faces $F$.

### 4.2.1 Coarse grading

Given a template garment $X_{base}$ and its sewing pattern $U_{base}$, the initial draping shape $X_{init}$ is simulated on the underlying body $B$ as in Section 3.2.2. The gap between the template mesh and the target can be quite large. Therefore, a coarse geometric deformation is performed to capture the overall geometric shape of the target garment, such as length and proportion, by using a set of key measurements in 3D.

The clothing boundaries offer valuable information regarding the overall orientation and deformation of the garment. Taking advantage of the fact that each garment has a fixed number of boundaries (e.g., two for a skirt and four for a t-shirt), the template mesh boundary(open contours) can be associated with the counterparts on the target clothing boundary by simple query. With this coarse boundary correspondence established, we proceed to a coarse garment deformation by linear grading as presented in Section 3.2.3.

### 4.2.2 Non-rigid geometry refinement

We further refine the simulated garment obtained from the previous stage through mesh deformation. Following the naming convention of mesh deformation, we refer to $X_{LG}$ as the source mesh later. A straightforward approach to deformation involves directly optimizing the coordinates of mesh vertices. However, in this method, the gradient of each vertex affects only its own displacement, which can result in the exposure of excessive high-frequency details, leading to undesirable artifacts(Figure 4.4) in the deformation. A potential solution is to employ a multi-stage alignment scheme Chen et al. (2021) that progressively optimizes the vertices. However, the consecutive intermediate optimizations introduce system complexity, which may hinder reproducibility.

Aiming to preserve the structure and topology of the source mesh while still fitting to the appearance of the target, we adopt an approach that deforms the vertices $X$ of the garment mesh using a set of per-triangle Jacobians, inspired by Aigerman et al. (2022); Gao et al. (2023). We denote per-triangle Jacobian as matrix $\mathbf{J}_i \in \mathbb{R}^{3x3}$. The Jacobians derived from the optimization with constraints are fused using a global Poisson equation to form a deformation map $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ over the vertices. The deformation gradient $\mathbf{F}_i = \nabla_i(\phi)$ of this mapping for each triangle should be closest to $\mathbf{J}_i$ in the least square sense, so that the gradients propagated through $\{\mathbf{J}_i\}$ influence a large surface mesh area, leading to a

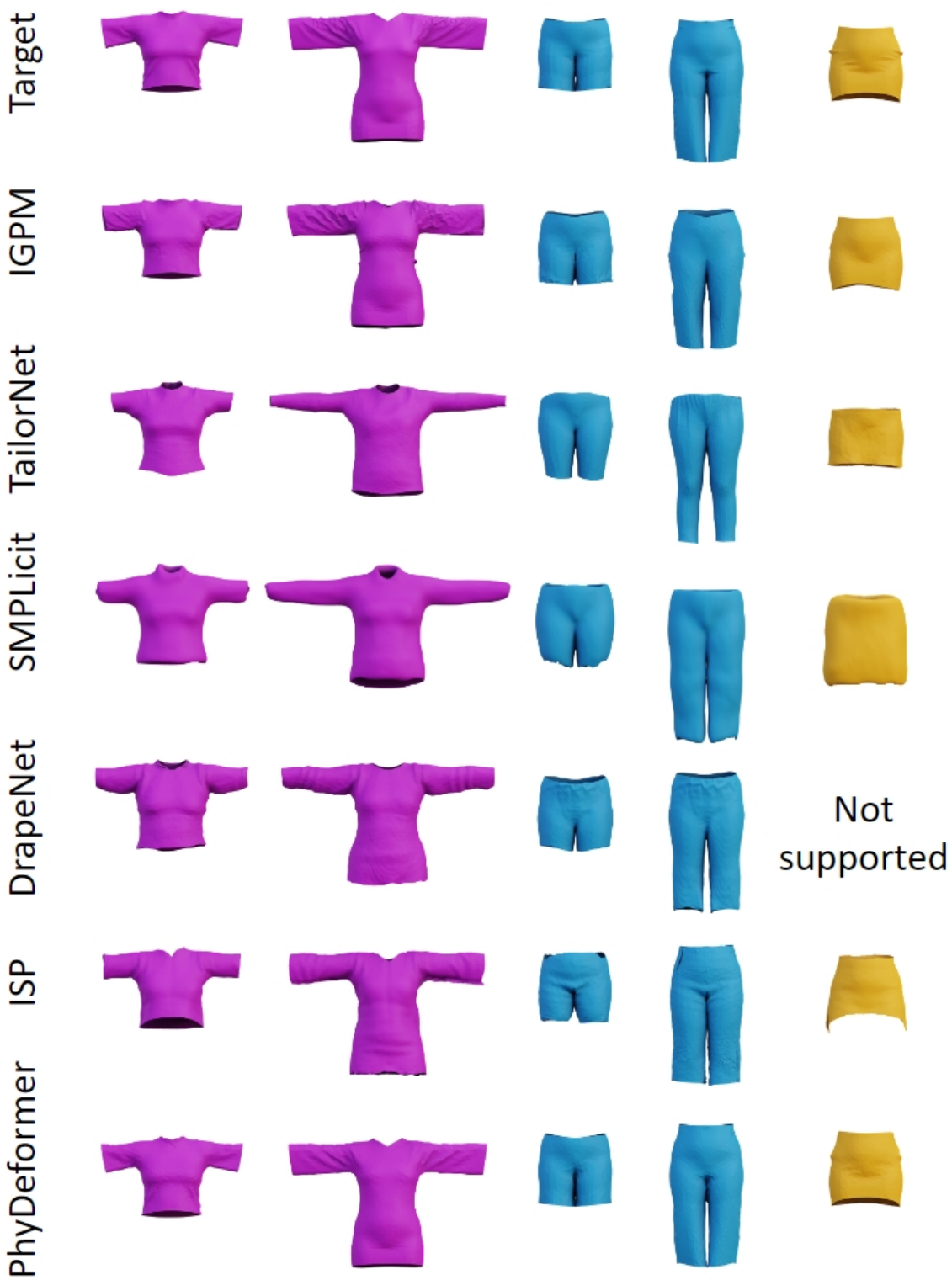


**Figure 4.2:** Comparison of 3D garment reconstruction of our method with others Corona et al. (2021); De Luigi et al. (2023); Li et al. (2024b); Patel et al. (2020). Our method reproduces faithful garment shapes, even accounting for intricate geometry details like wrinkles on large sleeves. Best viewed on screen zoomed-in.

globally-coherent deformation $\phi$ which is indirectly optimized by $\mathbf{J}_i$.

$$\phi^* = \min_{\Phi} \sum_{i=1}^{|F|} A_i \left\| \mathbf{F}_i - \mathbf{J}_i \right\|^2. \tag{4.1}$$

With the Jacobian representation established, we further optimize the triangles by introducing multiple loss terms, each targeting a specific purpose.

**Losses.** At each iteration, the garment geometry $\phi^*(X)$ is compared with the target $T = \{t_i\}$, and a loss function is utilized by a gradient-based algorithm to deform the garment mesh. We employ a set of loss terms to ensure that the refined garment mesh conforms to the target and the deformation is physically realistic. Several of these loss terms, specifically those arising from membrane strain energy and bending energy ($\mathscr{L}_s$, $\mathscr{L}_b$) have been proposed in the previous literature, notably in SNUG Santesteban et al. (2022a). Additionally, we introduce several novel loss terms.

We employ a reconstruction loss to evaluate the similarity between $X$ and $T$, which is composed of the Chamfer distances $\mathscr{L}_{CF}$ measured both between the surfaces ($X$ and $T$) and among open contours ($X_{open}$ and $T_{open}$),

$$\mathscr{L}_{rec} = \mathscr{L}_{CF}(X, T) + \mathscr{L}_{CF}(X_{open}, T_{open}). \tag{4.2}$$

The cosine distance of normals $\mathscr{L}_n$ between $X$ and $T$ is also measured, which is written as:

$$\mathscr{L}_n = \frac{1}{|X|} \sum_{x}^{|X|} \left(1 - \left\langle \mathbf{n}_x, \mathbf{n}_{\tilde{t}} \right\rangle\right) + \frac{1}{|T|} \sum_{t}^{|T|} \left(1 - \langle \mathbf{n}_t, \mathbf{n}_{\tilde{x}} \rangle\right), \tag{4.3}$$

where $\mathbf{n}_x$ and $\mathbf{n}_{\tilde{t}}$ are the unit normal vectors at point $x$ and $\tilde{t} = \arg\min_{t \in T} (\|x - t\|)$ respectively, and vice versa for $\mathbf{n}_t$ and $\mathbf{n}_{\tilde{x}}$.

Bending loss is employed to penalize changes in discrete curvature as a function of the dihedral angle between edge-adjacent triangles, which is formulated as follows:

$$\mathscr{L}_b = \sum^{|E|} \frac{k}{2} \alpha^2, \tag{4.4}$$

where $k$ represents the bending stiffness set to $4e-5$, $e$ is the edge, and $\alpha$ is the radian angle between two adjacent triangles. This loss term effectively constrains the deformation of the garment to prevent excessive bending.

Strain loss is employed to ensure that the shape of the triangles in the deformed garment resists stretching. This loss term is based on the Saint Venant Kirchhoff (StVK) elastic

material model Section 2.7.3, which is formulated as:

$$\mathscr{L}_s = \sum^{|F|} \left( \frac{\lambda}{2} tr\left(\mathbf{G}\right)^2 + \mu tr\left(\mathbf{G}^2\right) \right) A, \tag{4.5}$$

where $\lambda$ and $\mu$ represent the Lamé coefficients, which are respectively set to 16.364 and to 13.535 in our experiments. $A$ is the area of each triangle. $\mathbf{G}$ denotes the green strain tensor. We employ this as a regularization term to constrain the deformation, ensuring it does not deviate excessively from the original undeformed geometry.

Finally, collision loss is employed to prevent interpenetration of garment vertices with the body mesh. This is achieved by penalizing the negative distance between garment nodes and their closest point on the body mesh with a cubic energy term:

$$\mathscr{L}_c = \sum^{|X|} \max(\epsilon - sdf(x), 0)^3, \tag{4.6}$$

$sdf$ represents the distance between the query point and the body surface. Additionally, $\epsilon$ represents the given minimal distance threshold between the body and the garment. This loss term constrains the deformation of the garment to prevent intersection with the body mesh, resulting in a more plausible outcome.

All of these loss terms are combined to form the final loss function $L$ to guide the deformation of the source shape, which is formulated as follows:

$$\mathscr{L} = \mathscr{L}_{rec} + \lambda_n \mathscr{L}_n + \lambda_s \mathscr{L}_s + \lambda_b \mathscr{L}_b + \lambda_c \mathscr{L}_c. \tag{4.7}$$

## 4.3 Experiments

| | Results on LAVA lab dataset | | | | | |
|---|---|---|---|---|---|---|
| | Chamfer distance / Normal similarity | | | | | |
| Garments | SMPLicit | TailorNet | Drapenet | ISP | IGPM | Ours |
| T-shirt | 1.4 /- | 0.331 / 0.081 | 0.689 / 0.129 | 0.297 / 0.094 | 0.112 / 0.049 | **0.101** / **0.017** |
| Dress | 3.2 / - | 1.305 / 0.161 | 0.619 / 0.135 | 0.189 / 0.131 | 0.110 / 0.075 | **0.096** / **0.029** |
| Shorts | 1.3 / - | 1.036 / 0.050 | 0.131 / 0.048 | 0.202 / 0.095 | 0.126 / 0.043 | **0.088** / **0.019** |
| Pants | 2.9 / - | 2.587 / 0.104 | 0.485 / 0.085 | 0.185 / 0.077 | 0.142 / 0.049 | **0.084** / **0.019** |
| Skirt | 6.5 / - | 1.30 / 0.063 | - / - | 0.435 / 0.093 | 0.106 / 0.014 | **0.086** / **0.007** |

**Table 4.1:** Quantitative comparison results in 3D garment reconstruction on LAVA lab dataset. Our method performs consistently better than others. The best results are in boldface.

### 4.3.1 Implementation details

We conducted the experiments using a NVIDIA 3090 GPU, 24Gb RAM, and an Intel i7-5220R CPU. We run the optimization for 1500 iterations until the convergence is reached, which takes approximately only 5 minutes. The stretching regularization term is added after 500 iterations. The learning rate was set to 0.002 in the ADAM optimizer. In terms of loss weights, the parameters were set as follows: $\lambda_{rec} = 1$, $\lambda_n = 0.01$, $\lambda_c = 0.01$, $\lambda_b = 0.1$, and $\lambda_s = 1$ for our experiments.

### 4.3.2 Evaluation on synthetic datasets

| Results on Sewfactory dataset<br>Chamfer distance / Normal similarity | | | | |
|---|---|---|---|---|
| Garments | Drapenet | ISP | IGPM | Ours |
| S01 | 0.346 / 0.235 | 0.431 /0.232 | 0.191 / 0.203 | **0.159** / **0.058** |
| S02 | 17.031 / 0.165 | 17.61 /0.214 | 0.083 / 0.089 | **0.047** / **0.021** |
| S03 | 0.798 / 0.156 | 0.182 / 0.164 | 0.205 / 0.124 | **0.076** / **0.061** |
| S04 | - / - | 3.024 / 0.340 | 0.398 / 0.276 | **0.228** / **0.27** |
| S05 | 1.831/ 0.268 | 0.87 / 0.186 | 0.090 / 0.07 | **0.138** / **0.028** |

**Table 4.2:** Quantitative evaluation in 3D garment reconstruction on Sewfactory dataset. Note that ours outperforms others in the 3D reconstruction of posed garments, measured by Chamfer distance (L2 norm, scaled by $e^3$) and the normal vector consistency.

**Dataset:** To validate our method for garment fitting, we leveraged the geometry of 3D garments from two synthetic 3D datasets as targets. LAVA lab dataset Korosteleva et Lee (2021) contains a wide range of high-quality garment meshes simulated over the average female body at T pose; Sewfactory Liu et al. (2023) extends to a more comprehensive dataset containing diverse garment styles and human shapes in various poses. We selected several representative 3D garment geometries from the datasets, and they serve as the targets. For the base meshes, we utilize a set of garment templates that cover basic clothing categories (e.g., shirts, skirts, dresses, etc.). These base meshes are presented in a canonical pose and will be made publicly available alongside the code to support future research in 3D garment modeling.

**State of the arts:** We evaluate our method with the garments sampled from the above datasets, and compare our approach to the related methods for garment fitting: Tailornet Patel et al. (2020), SMPLicit Corona et al. (2021), Drapenet and ISP De Luigi et al. (2023); Li et al. (2024b), and IGPM Yu et al. (2024).

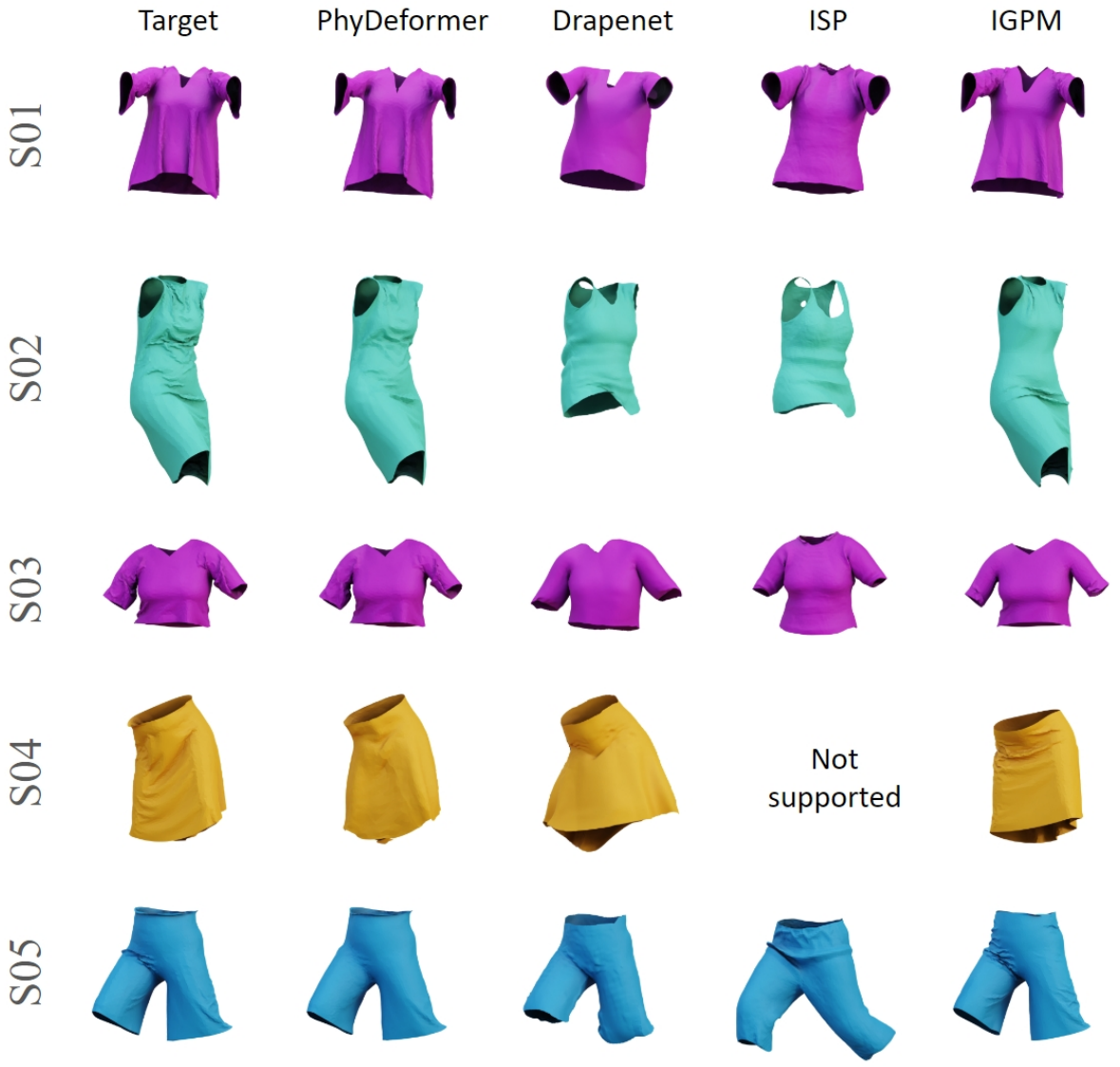


**Figure 4.3:** Qualitative comparison of 3D garment reconstruction using our method versus other approaches De Luigi et al. (2023); Li et al. (2024b); Yu et al. (2024). Our method reproduces faithful garment shapes, even accounting for intricate geometry details like wrinkles and folds.

We remind that the parameter space representing the garment geometry differs from each other: the $\gamma$ garment style parameter for TailorNet Patel et al. (2020), the latent vector $z = [z_{cut}, z_{style}]$ describing garment cut and style for SMPLicit Corona et al. (2021), the latent codes $z$'s encoding the garment characteristics in DrapeNet and ISP De Luigi et al. (2023); Li et al. (2024b), and the coordinates of effective control points and material parameters in IGPM. The geometric similarity is quantitatively measured using two metrics: Chamfer distance to the ground truth mesh vertices, and the cosine normal similarity to measure the similarity of the computed normal vectors. As illustrated in Table 4.1 and Table 4.2, our method is consistently better than others, which is confirmed by qualitative results, shown in Figure 4.2, Figure 4.3.

TailorNet has learned over tight-fit datasets, so it does not generalize well when fitting to loose styles. SMPLicit is based on the signed distance function, resulting in watertight meshes instead of open surfaces after meshing, which inherently reduces the fitting accuracy with the inflated surface. Drapenet uses unsigned distance functions (UDFs) and requires extra computations for meshing, and it is sensitive to the initialization of latent code for optimization. ISP and Drapenet can produce certain geometric details in the registration to garments, but when it comes to more challenging posed garment samples (Figure 4.3), the performance decreases and leads to visible defects. IGPM exploits the inverse cloth simulation to achieve coarse-to-fine alignment, but it fails to capture the exact wrinkle patterns and the inverse simulation is relatively expensive to execute. In contrast, PhyDeformer reconstructs accurate 3D geometry, for both loose and tight garments of different subjects with less computational cost.

### 4.3.3 Ablation study

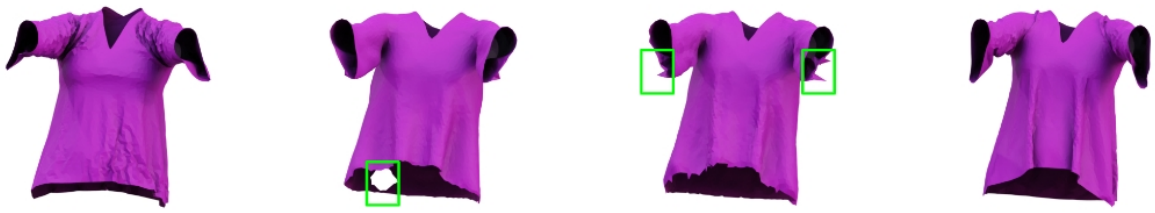


**Figure 4.4:** Removing Jacobians and optimizing displacements instead. From left to right: (a) source mesh, (b) deformed mesh with naive vertex-displacements optimization, (c) deformed mesh with clipped gradients to avoid holes caused by "NaNs", (d) target mesh.

We conduct ablation studies to evaluate how the components affect the overall fitting

quality. We present some metrics for S01 in Table 4.3 since it is the most representative case in our experiments that can clearly reflect the effect of different modules.

**Without linear grading and without Jacobians.** Optimizing vertex-wise displacements or removing the linear grading based coarse alignment can result in distorted shapes with unwanted artifacts.

In Figure 4.4, we show that replacing Jacobians with vertex displacements in the optimization decreases the resulting surface quality. Representing the deformation through vertex displacements exposes the per-vertex high-frequency mode of the deformation, thus leading to localized noisy gradients that deteriorate the triangulation of the meshes(holes) as illustrated in Figure 4.4 (b). Clipping the gradients can alleviate this but still causes sharp "burrs"Figure 4.4 (c) on the deformed mesh.

In Figure 4.5 (b) we showcase our method's performance declines if the linear grading stage (stage 1) is disabled. The advantage of using linear grading is that it provides better initialization, thus improving overall performance.


**Figure 4.5:** From left to right: (a) posed template mesh, (b) deformed mesh, (c) target mesh

**Without loss terms.** Figure 4.6 shows that our refinement stage with all losses can lead to large and smooth deformations for the final fitting. To assess the effectiveness of each loss term, we conducted an ablation study by sequentially omitting individual losses — specifically, the contour loss, normal loss, and bending loss.

As shown in Table 4.3, the removal of the open contour loss term did not affect much the global geometric shape, though it led to an increase in Chamfer distance. The consequence is more perceptible in Figure 4.6 (b), where a visible failure around the collar region can be observed. Eliminating the normal restriction significantly compromised

the resulting deformed mesh, as demonstrated in Figure 4.6 (c). Similarly, omitting the bending loss introduced evident bumpy artifacts, shown in Figure 4.6 (d). Quantitative results in Table 4.3 confirm that our two-phase refining strategy enhances the geometric accuracy of the output.

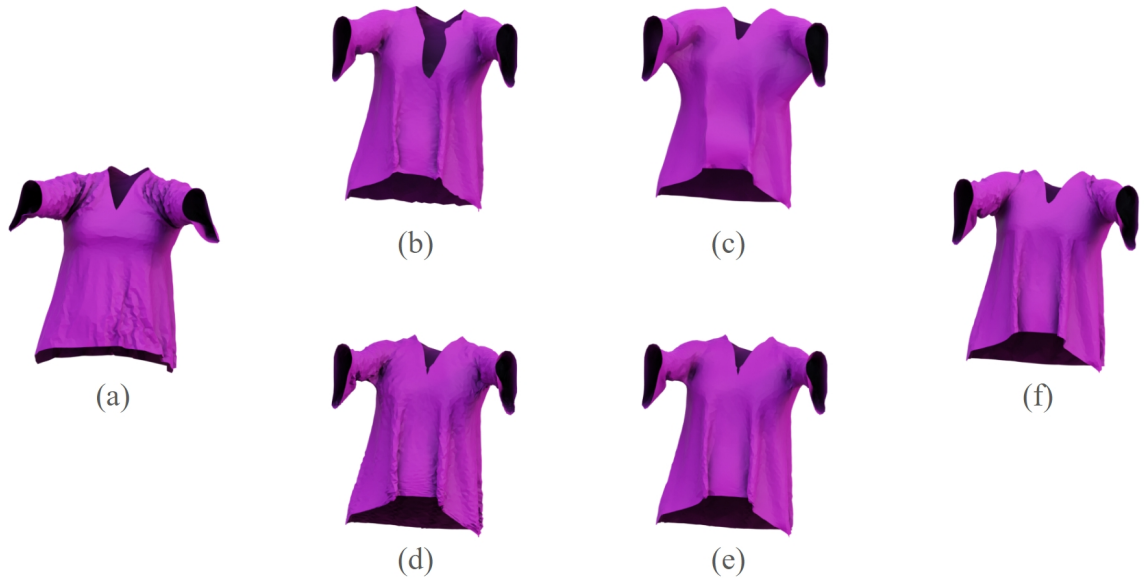


**Figure 4.6:** Results of our ablation study on loss terms: (a) source mesh, (b) without contour loss, (c) without normal loss, (d) without bending loss, (e)with all losses, (f) target mesh.

| Method | Chamfer distance (CF) | Normal similarity |
|---|---|---|
| w/o linear grading | 0.188 | 0.106 |
| w/o contour loss | 0.088 | 0.056 |
| w/o normal consistency | 0.124 | 0.173 |
| w/o bending loss | 0.158 | 0.052 |
| Ours | **0.047** | **0.021** |

**Table 4.3:** Quantitative results of ablation studies. We report the metrics for the garment S01. The best results are in boldface.

### 4.3.4 Robustness

To comprehensively understand PhyDeformer's behavior, particularly the effectiveness of the second stage (the role of the first stage has already been analyzed in the ablation study), we evaluated its robustness to different types of deformation source meshes and to different levels of noise.

**Initial condition.** In linear grading, the sizing alteration is cast in 3D garment mesh through a simulation in the T-pose space. However, when it comes to the posed target garments as in Figure 4.3, a point $x$ in the T-pose space should be moved to the pose space (defined by the SMPL pose parameter $\theta$). The articulation of garments is either achieved by a simulation with SMPL body meshes or a linear blend skinning (LBS) transformation Santesteban et al. (2021). We found that PhyDeformer exhibits robustness to both simulation and linear blend skinning (LBS) transformations as shown in Figure 4.7, despite the geometric variations. The intrinsic Jacobian deformation mechanism is capable of accommodating relatively large deformations, which inherently enhances the stability of producing high-quality outputs. Therefore, it supports the feasibility of replacing costly simulations with linear blend skinning (LBS). Reducing reliance on simulation and adopting a geometric solution makes the entire pipeline more lightweight.

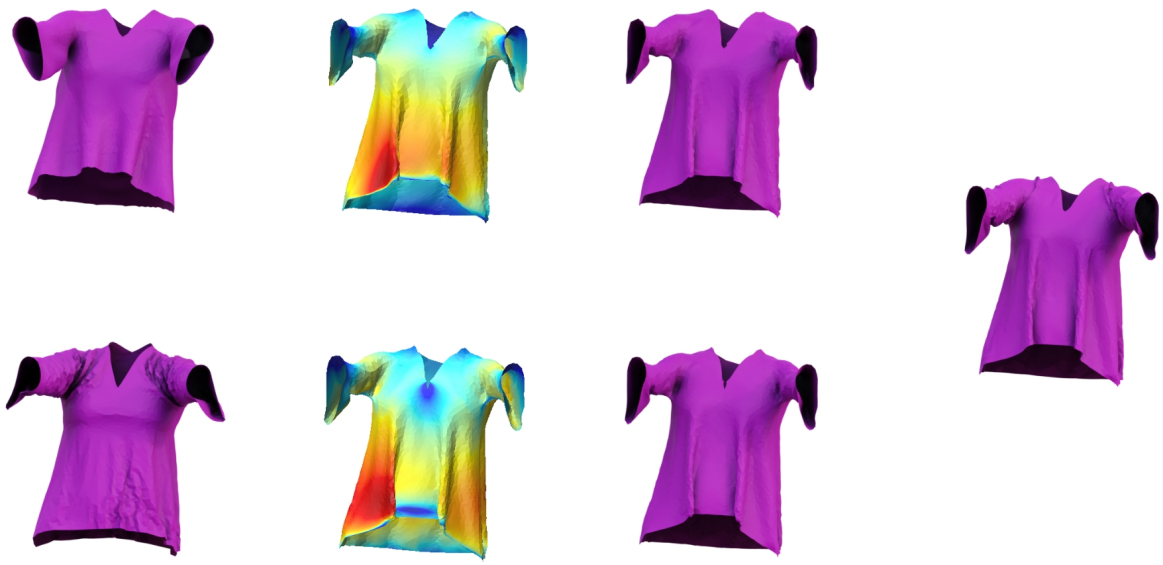

**Figure 4.7:** Robustness to different source meshes: the leftmost column displays the source meshes (with the upper images posed by LBS and the lower images draped by simulation), the second column shows the qualitative colormap of displacement magnitude, the third column presents the deformed meshes, and the rightmost column depicts the target mesh.

**Noisy target.** To get a complete picture of the PhyDeformer behavior, we also assess its robustness against noise present in the input. We simulate the noisy artifacts by introducing Gaussian noise to the vertices coordinates of the target mesh, adjusting the standard deviation between 0.5 and 1 cm. When evaluating PhyDeformer with these modified targets (see Figure 4.8), we observe a decline in performance as noise levels increase. This indicates that PhyDeformer can effectively manage minimal noise levels, typical of high-

accuracy 3D scanning systems. Enhancing its robustness to handle much noisier data would be a valuable avenue for future research.

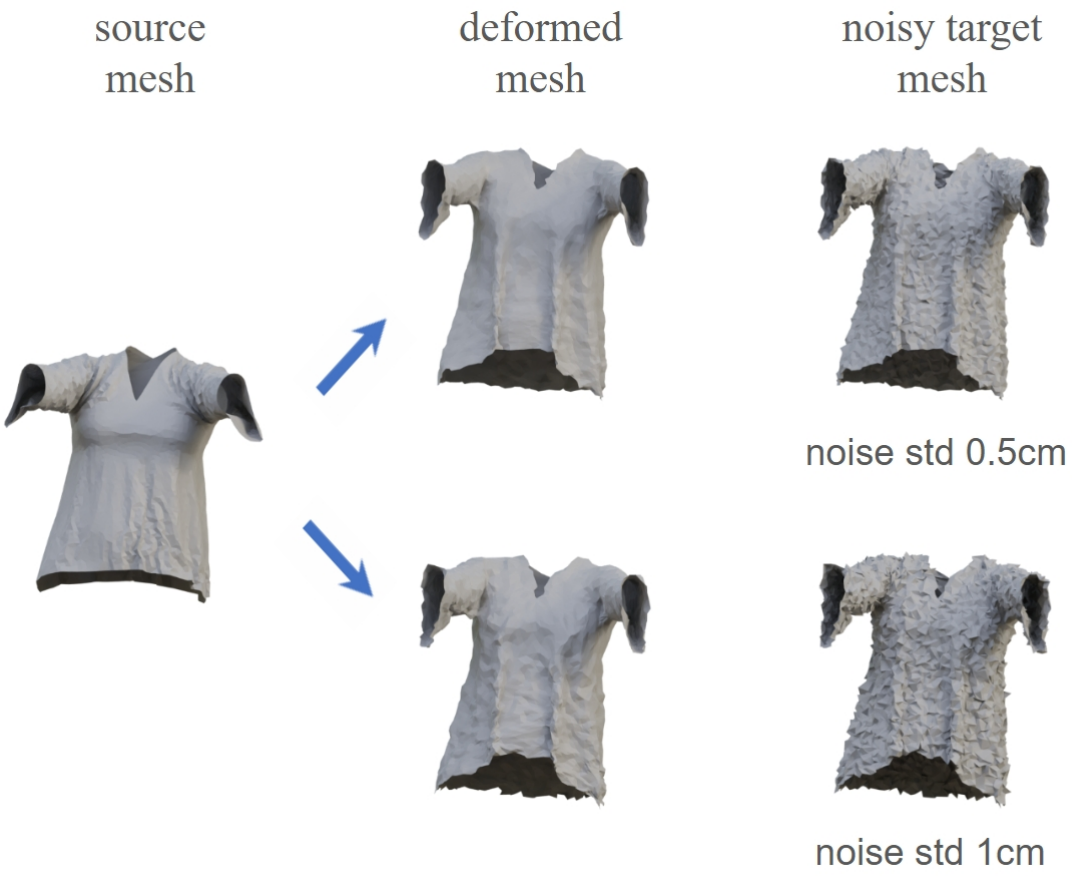


**Figure 4.8:** The alignment result for data with synthetic noise. Different levels of noise are applied to the target mesh.

### 4.3.5 Evaluation on 3D scan data

To demonstrate the capability of our refinement stage in handling fine-grained, wrinkle-level details from real scans, as well as its ease of integration with other techniques, we performed a qualitative evaluation using the GarmCap dataset Lin et al. (2023). The dataset contains high-quality 3D posed garment scans of four different garments: G01, a T-shirt with rich wrinkles; G02, a knee-long coat; G03, a thick coat and G04, a down coat. We adopted their rigged smooth template to obtain the coarse fitting and refined the alignment with our second stage. As shown in Figure 4.9, our method outputs reasonable, high-quality reconstruction of the 3D scanned garments.


**Figure 4.9:** Qualitative results on GarmCap dataset. Leftmost column: canonicalized templates, second column: posed templates, third column: coarse fittings, fourth column: refined fittings by PhyDeformer, Rightmost column: target shape. Best viewed zoomed-in.

### 4.3.6 Registration for inverse simulation

In both the previous chapter and this one, we employ a coarse-to-fine strategy, with the first stage being linear grading. The primary difference lies in the second stage, which transitions from a differentiable simulation based optimization to a geometric mesh deformation based optimization. The focus of PhyDeformer is more on mesh deformation to align the template mesh as closely as possible with the target mesh in geometry.

Applying physics-based loss functions effectively normalizes deformation and produces visually realistic garments. While these outputs differ from the simulation-ready results seen in the previous chapter, this method offers significantly better efficiency than fully physics-based simulation-embedded optimization. Recognizing the unique strengths of both approaches, we explore ways to combine them, aiming to leverage their respective advantages for a "hybrid" solution. In addition to being an efficient alternative approach to IGPM, PhyDeformer can function as a clothing alignment module, providing a

registered target mesh with consistent topology. After the linear grading stage, the output is used to register the raw target mesh. This process involves running the Jacobian-deformation optimization to align the source and target meshes. The resulting deformed mesh retains the same connectivity as the source mesh. It can be utilized as a "new registered" target for the differentiable simulation optimization, with the source mesh and its graded pattern serving as the initial state. Since the topology is the same, a per-vertex loss(mean square error) can be used which is more expressive. We refer to this scheme as "hybrid" in contrast to the baseline IGPM in Chapter 3. In Figure 4.10, we showcase how the "hybrid" strategy outperforms the vanilla IGPM in terms of the speed of convergence, which also produces more accurate reconstruction.

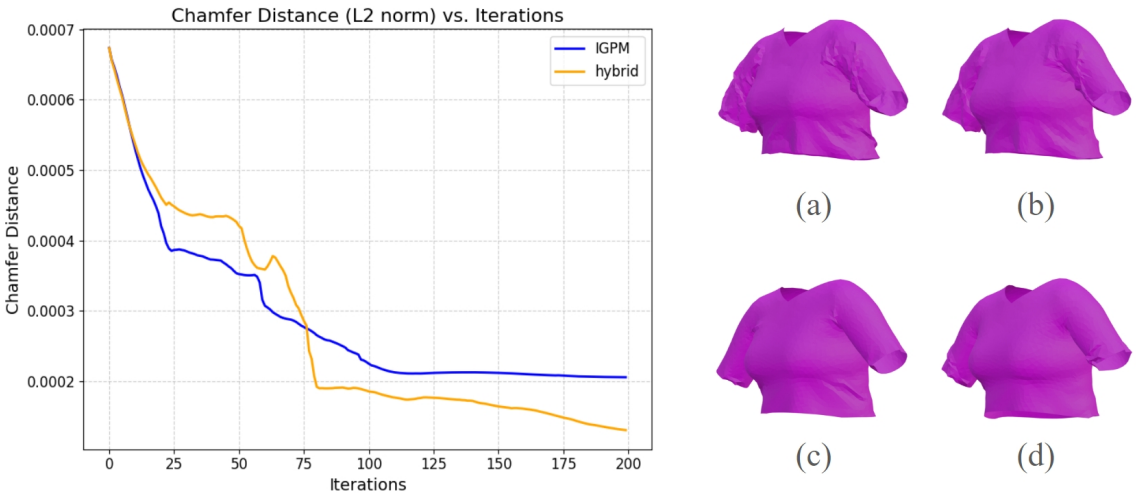


**Figure 4.10:** We evaluate the Chamfer Distance error for hybrid and IGPM. To the right, we also illustrate the qualitative results: (a) target mesh (b) deformed mesh by PhyDeformer (c) hybrid (d) IGPM.

## 4.4 Conclusion

In this chapter, we introduce PhyDeformer, a light weight mesh deformation technique that operates without the need for training on any 3D dataset. Instead, we employ a collection of loss functions that not only align the geometry with specific targets but also ensure approximate adherence to physical principles, thereby significantly reducing visual artifacts. Our method aims to generate high-quality geometric outputs by reproducing both low-frequency shape changes and high-frequency details through source shape deformations. We optimize per-face Jacobians to predict smooth mesh deformations, preserving the intrinsic topology of the source shape. This approach results in high-quality

mesh outputs with useful geometry and minimizes local artifacts typically associated with vertex displacements.

**Future Work.** Aligned garment meshes enable a wide range of applications, such as texture transfer, deformation capture, and physical parameter estimation for draping on human avatars. In our current setting, it is assumed that garment data are well segmented scans/synthetic data. Future work should extend this pipeline to support imperfections such as noise, and segmentation failures. Besides, we also aim to explore the potential of integrating PhyDeformer's outputs with a 2D pattern estimation method. This approach could offer greater efficiency, as Jacobians can be optimized on a case-by-case basis, avoiding the need for extensive optimization in a differentiable simulation framework or the training of deep learning models on large datasets. Furthermore, we plan to connect our method with an interactive interface, creating a more comprehensive, artist-driven tool that allows users to explore results from different garment designs.

CHAPTER 5

# CONCLUSION

In this concluding chapter, we provide a summary of the key findings and contributions of this research study, emphasizing the contributions in both practical applications and methodological advancements.

## Summary and discussion

In Chapter 3, we present an approach that utilizes differentiable simulation for clothing recovery to generate high-quality, physically plausible assets that can be used for simulation applications. One major existing challenge is the high diversity of shapes and styles in the garment domain. We propose to use the parameterized garment sewing pattern as a realistic and compact garment descriptor, which allows a unified representation to cope with the pattern variations and facilitates the estimation process. Another critical task is the disentanglement of the intrinsic garment shape from physical deformations due to fabric properties, physical forces, and contact with the body. We address this challenge by representing the draping garment geometry as the outcome of both the pattern shape from which the garment is constructed and the physical interactions between the garment and the underlying body. Specifically, we formulate the problem as estimating cloth simulation parameters from the clothing parts of 3D input data, which is either the result of a physics-based simulation or a 3D scan of a real-world dressed human. The solution involves fitting the desired cloth geometry by jointly optimizing over a continuous space

of pattern shape starting from a template base mesh, along with the material parameters initialized with a predefined set of fabrics. While we employ dynamic simulation, we focus on states near quasi-equilibrium to ensure numerical stability. Our method considers the complex non-linear behavior of cloth and its intricate interaction with the underlying body when co-optimizing the parameters, thereby accounting for the interplay between all components involved.

We showcased that our method outperforms prior works across different metrics, producing high-quality garment results in both 3D garment and 2D pattern space and generating simulation-ready assets. Our method is designed to be agnostic to the type of clothes, e.g., T-shirts, dresses, and pants. The approach does not require a dataset for training, has the understanding of physics, and results in interpretable and plausible cloth behavior. Having the altered garment patterns allows adopting the garments in novel conditions for downstream tasks, we can synthesize new cloth animations by dressing them on different body sequences in action, which could be on different subjects. It also enables the users to adjust the design of a captured garment. This competence of convenient garment reanimation and garment retargeting is highly desirable for virtual try-on, garment design, or avatar creation.

We improve the quality of reconstructed geometry as detailed in PhyDeformer Chapter 4 by transitioning the kernel of refinement from a differentiable simulation to a Jacobian-based deformation approach within the optimization loop. This modification results in a notable improvement in the quality of the mesh outputs while being more efficient in time. Although the reconstructed meshes using PhyDeformer(focusing solely on geometric recovery) are less compatible with PBS compared to IGPM, they can serve as registered targets for IGPM and yield a suboptimal yet simulation-ready reconstructed mesh, as depicted in Section 4.3.6.

In conclusion, this thesis presents solutions that contribute to the automation of clothed virtual avatar creation. Our work spans the topics of garment modeling, differentiable simulation, and geometric deformation utilizing physics-inspired losses to ensure plausibility. The proposed models generate realistic clothing suitable for a range of applications, particularly in virtual and augmented reality. Overall, this research highlights the value and impact of advancing scene recovery methods to produce high-quality, physically plausible virtual clothing, which can potentially benefit future exploration within the field.

## Limitations and prospectives

Our proposed methods have demonstrated encouraging outcomes in garment reconstruction and pattern estimation. However, several limitations persist and require further attention. In this part, we address these limitations and propose potential directions for future research and enhancement.

The scope of this thesis is confined to single-layer garments. However, we believe that our methods have the potential to be extended to cope with multi-layered clothing. In multi-layered scenarios, garments may be occluded, and the interactions between layers become more complex due to frequent contact and collisions, making the accurate recovery of multi-layered garments significantly challenging. Additionally, while we can replicate precisely the overall style and fit of the target garment's geometry, we did not address the synthesis of photorealistic textures. This aspect is also crucial for enhancing realism, particularly in narrowing the gap between virtual models and real human scans through texture extraction and transfer as present in Xiang et al. (2022, 2021).

Although IGPM remains within the framework of FEM, it is not limited to the specific implementation. We believe it demonstrates the feasibility of the approach and may encourage further exploration with alternative solvers, such as constraint-based methods or projected dynamics. These methods could offer advantages in terms of computational time efficiency, both in forward and backward simulations, though potentially at the expense of accuracy. Additionally, a more rigorous investigation into the feasible set of material parameter combinations should be conducted to enhance the robustness of the approach. Another limitation of IGPM stems from its approach to pattern adjustment. Specifically, it lacks the ability to automatically manage changes in the number of pattern panels, whether additional panels are required or redundant panels need to be eliminated. Nevertheless, users can manually intervene using the provided graphical user interface (GUI) to make such modifications. A possible future direction could involve integrating learning-based methods to automatically predict base patterns based on semantic information.

More potential pistes for future research include investigating the dynamics of clothing. Given that our method incorporates an inverse quasi-static simulation to adjust the garment shape, a logical extension would be to match dynamic garment sequences and recover parameters associated with garment motion. While previous efforts have been made to reconstruct the dynamic geometry of clothed humans, they do not fully address the reconstruction of arbitrary garments in motion. Guo et al. (2021) considers the dy-

namics of the cloth geometry are only controlled by the movement history of the body states; Zheng et al. (2024b) goes further and also searches for the optimal physical parameters minimizing the difference between the simulated and the ground truth garment meshes. Both methods rely on a fixed, simulation-ready initial garment mesh that closely resembles the targets but do not incorporate optimization of the garment itself during reconstruction, which limits the practicability in terms of generalization.

The differentiable simulation may be extended to handle multi-layers due to its strong physical priors. However, fully resolving all cloth interactions presents substantial computational costs. Therefore, advancements in real-time simulation techniques aimed at boosting computational efficiency could be utilized to further improve performance.

Besides, the Jacobians-based mesh deformation method allows for the direct integration of more physics-based losses into its pipeline. More terms can be tested for objectives like preventing self-intersection within the garment, ensuring that the lower-layer garment is positioned closer to the underlying body than the upper garment, preventing penetrations between the clothes etc. Another promising avenue would be to investigate the use of differentiable rendering Laine et al. (2020) to connect 3D garment mesh geometry to powerful Vision Language Models, for garment stylization and texturization.

# BIBLIOGRAPHY


Aigerman, N., Gupta, K., Kim, V. G., Chaudhuri, S., Saito, J., et Groueix, T. (2022). Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH*.

Arkin, E., Chew, L., Huttenlocher, D., Kedem, K., et Mitchell, J. (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216.

Bang, S., Korosteleva, M., et Lee, S.-H. (2021). Estimating garment patterns from static scan data. In *Computer Graphics Forum*, volume 40, pages 273–287. Wiley Online Library.

Baraff, D. et Witkin, A. (1998). Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 43–54, New York, NY, USA. Association for Computing Machinery.

Bartle, A., Sheffer, A., Kim, V. G., Kaufman, D. M., Vining, N., et Berthouzoz, F. (2016). Physics-driven pattern adjustment for direct 3d garment editing. *ACM Trans. Graph.*, 35(4):50–1.

Bednarik, J., Parashar, S., Gundogdu, E., Salzmann, M., et Fua, P. (2020). Shape reconstruction by learning differentiable surface representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4716–4725.

Bergou, M., Wardetzky, M., Harmon, D., Zorin, D., et Grinspun, E. (2006). A quadratic bending model for inextensible surfaces. In *Symposium on Geometry Processing*, volume 227, page 230.

Bertiche, H., Madadi, M., et Escalera, S. (2020). Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer.

Bertiche, H., Madadi, M., et Escalera, S. (2021). Pbns: Physically based neural simulation for unsupervised garment pose space deformation. *ACM Trans. Graph.*, 40(6):14.

Bhatnagar, B. L., Sminchisescu, C., Theobalt, C., et Pons-Moll, G. (2020a). Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer.

Bhatnagar, B. L., Sminchisescu, C., Theobalt, C., et Pons-Moll, G. (2020b). Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bhatnagar, B. L., Tiwari, G., Theobalt, C., et Pons-Moll, G. (2019). Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5420–5430.

Bongratz, F., Rickmann, A.-M., Pölsterl, S., et Wachinger, C. (2022). Vox2cortex: fast explicit reconstruction of cortical surfaces from 3d mri scans with geometric deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20773–20783.

Bridson, R., Marino, S., et Fedkiw, R. (2005). Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*, pages 3–es.

Buffet, T., Rohmer, D., Barthe, L., Boissieux, L., et Cani, M.-P. (2019). Implicit untangling: A robust solution for modeling layered clothing. *ACM Transactions on Graphics (TOG)*, 38(4):1–12.

Chen, C.-H., Su, J.-W., Hu, M.-C., Yao, C.-Y., et Chu, H.-K. (2024). Panelformer: Sewing pattern reconstruction from 2d garment images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 454–463.

Chen, X., Pang, A., Yang, W., Wang, P., Xu, L., et Yu, J. (2021). Tightcap: 3d human shape capture with clothing tightness field. *ACM Trans. Graph.*, 41(1).

Chen, X., Wang, G., Zhu, D., Liang, X., Torr, P., et Lin, L. (2022). Structure-preserving 3d garment modeling with neural sewing machines. *Advances in Neural Information Processing Systems*, 35:15147–15159.

Cirio, G., Lopez-Moreno, J., Miraut, D., et Otaduy, M. A. (2014). Yarn-level simulation of woven cloth. *ACM Transactions on Graphics (TOG)*, 33(6):1–11.

Cirio, G., Lopez-Moreno, J., et Otaduy, M. A. (2015). Efficient simulation of knitted cloth using persistent contacts. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 55–61.

Corona, E., Pumarola, A., Alenyà, G., Pons-Moll, G., et Moreno-Noguer, F. (2021). Smplicit: Topology-aware generative model for clothed people. In *CVPR*.

Daněřek, R., Dibra, E., Öztireli, C., Ziegler, R., et Gross, M. (2017). Deepgarment: 3d garment shape estimation from a single image. In *Computer Graphics Forum*, volume 36, pages 269–280. Wiley Online Library.

De Luigi, L., Li, R., Guillard, B., Salzmann, M., et Fua, P. (2023). Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1451–1460.

Du, T., Wu, K., Ma, P., Wah, S., Spielberg, A., Rus, D., et Matusik, W. (2021). Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (TOG)*, 41(2):1–21.

Eberhardt, B., Weber, A., et Strasser, W. (1996). A fast, flexible, particle-system model for cloth draping. *IEEE computer graphics and applications*, 16(5):52–59.

Feng, Y., Choutas, V., Bolkart, T., Tzionas, D., et Black, M. J. (2021). Collaborative regression of expressive bodies using moderation. In *2021 International Conference on 3D Vision (3DV)*, pages 792–804. IEEE.

Feng, Y., Yang, J., Pollefeys, M., Black, M. J., et Bolkart, T. (2022). Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9.

Gao, W., Aigerman, N., Groueix, T., Kim, V., et Hanocka, R. (2023). Textdeformer: Geometry manipulation using text guidance. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11.

Gast, T. F., Schroeder, C., Stomakhin, A., Jiang, C., et Teran, J. M. (2015). Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics*, 21(10):1103–1115.

Goto, C. et Umetani, N. (2021). Data-driven garment pattern estimation from 3d geometries. In *Eurographics (Short Papers)*, pages 17–20.

Grigorev, A., Thomaszewski, B., Black, M. J., et Hilliges, O. (2023). HOOD: Hierarchical graphs for generalized modelling of clothing dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Gundogdu, E., Constantin, V., Seifoddini, A., Dang, M., Salzmann, M., et Fua, P. (2019). Garnet: A two-stream network for fast and accurate 3d cloth draping. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Guo, J., Li, J., Narain, R., et Park, H. S. (2021). Inverse simulation: Reconstructing dynamic geometry of clothed humans via optimal control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14698–14707.

Habermann, M., Xu, W., Zollhofer, M., Pons-Moll, G., et Theobalt, C. (2020). Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063.

Harmon, D., Vouga, E., Tamstorf, R., et Grinspun, E. (2008). Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*, pages 1–4.

Hassan, M., Choutas, V., Tzionas, D., et Black, M. J. (2019). Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, pages 2282–2292.

He, K., Yao, K., Zhang, Q., Yu, J., Liu, L., et Xu, L. (2024). Dresscode: Autoregressively sewing and generating garments from text guidance. *arXiv preprint arXiv:2401.16465*.

Jang, J., Yun, W.-H., Kim, W. H., Yoon, Y., Kim, J., Lee, J., et Han, B. (2023). Learning to boost training by periodic nowcasting near future weights. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., et Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14730–14757. PMLR.

Jiang, B., Zhang, J., Hong, Y., Luo, J., Liu, L., et Bao, H. (2020). Bcnet: Learning body and cloth shape from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 18–35. Springer.

Kingma, D. P. et Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Korosteleva, M. et Lee, S.-H. (2021). Generating datasets of 3d garments with sewing patterns. In Vanschoren, J. et Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Korosteleva, M. et Lee, S.-H. (2022). Neuraltailor: reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Transactions on Graphics (TOG)*, 41(4):1–16.

Lahner, Z., Cremers, D., et Tung, T. (2018). Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European conference on computer vision (ECCV)*, pages 667–684.

Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., et Aila, T. (2020). Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)*, 39(6):1–14.

Larionov, E., Eckert, M.-L., Wolff, K., et Stuyck, T. (2022). Estimating cloth elasticity parameters using position-based simulation of compliant constrained dynamics. *arXiv preprint arXiv:2212.08790*.

Li, R., Dumery, C., Guillard, B., et Fua, P. (2024a). Garment Recovery with Shape and Deformation Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, R., Guillard, B., et Fua, P. (2024b). Isp: Multi-layered garment draping with implicit sewing patterns. *Advances in Neural Information Processing Systems*, 36.

Li, R., Guillard, B., Remelli, E., et Fua, P. (2022a). Dig: Draping implicit garment over the human body. In *Proceedings of the Asian Conference on Computer Vision*, pages 2780–2795.

Li, Y., Chen, H.-y., Larionov, E., Sarafianos, N., Matusik, W., et Stuyck, T. (2024c). DiffAvatar: Simulation-ready garment optimization with differentiable simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y., Du, T., Wu, K., Xu, J., et Matusik, W. (2022b). Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics (TOG)*, 42(1):1–20.

Li, Y., Habermann, M., Thomaszewski, B., Coros, S., Beeler, T., et Theobalt, C. (2021). Deep physics-aware inference of cloth deformation for monocular human performance capture. In *2021 International Conference on 3D Vision (3DV)*, pages 373–384. IEEE.

Liang, J., Lin, M., et Koltun, V. (2019). Differentiable cloth simulation for inverse problems. *Advances in Neural Information Processing Systems*, 32.

Lin, S., Zhou, B., Zheng, Z., Zhang, H., et Liu, Y. (2023). Leveraging intrinsic properties for non-rigid garment alignment. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

linctex (2024). Style3d studio. Style3D.

Liu, L., Xu, X., Lin, Z., Liang, J., et Yan, S. (2023). Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (TOG)*, 42(6):1–15.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., et Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16.

Ma, Q., Saito, S., Yang, J., Tang, S., et Black, M. J. (2021a). Scale: Modeling clothed humans with a surface codec of articulated local elements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16082–16093.

Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., et Black, M. J. (2020). Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478.

Ma, Q., Yang, J., Tang, S., et Black, M. J. (2021b). The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10974–10984.

Macklin, M., Müller, M., et Chentanez, N. (2016). Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., et Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451.

Meng, Y., Mok, P. Y., et Jin, X. (2012). Computer aided clothing pattern design with 3d editing and pattern alteration. *Computer-Aided Design*, 44(8):721–734.

Miguel, E., Bradley, D., Thomaszewski, B., Bickel, B., Matusik, W., Otaduy, M. A., et Marschner, S. (2012). Data-driven estimation of cloth simulation models. In *Computer Graphics Forum*, volume 31, pages 519–528. Wiley Online Library.

Müller, M., Heidelberger, B., Hennix, M., et Ratcliff, J. (2007). Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118.

Nag, S., Qi, A., Zhu, X., et Shamir, A. (2023). Personaltailor: Personalizing 2d pattern design from 3d garment point clouds. *arXiv preprint arXiv:2303.09695*.

Narain, R., Samii, A., et O'brien, J. F. (2012). Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)*, 31(6):1–10.

NVIDIA (2024). Kaolin library documentation. Kaolin.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., et Guibas, L. (2012). Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11.

Park, J. J., Florence, P., Straub, J., Newcombe, R., et Lovegrove, S. (2019). Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Patel, C., Liao, Z., et Pons-Moll, G. (2020). Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., et Black, M. J. (2019a). Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., et Black, M. J. (2019b). Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985.

Pietroni, N., Dumery, C., Falque, R., Liu, M., Vidal-Calleja, T., et Sorkine-Hornung, O. (2022). Computational pattern making from 3d garment models. *ACM Transactions on Graphics (TOG)*, 41(4):1–14.

Pons-Moll, G., Pujades, S., Hu, S., et Black, M. J. (2017). Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (ToG)*, 36(4):1–15.

Provot, X. et al. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society.

Punnakkal, A. R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., et Black, M. J. (2021). BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731.

Ranjan, A., Bolkart, T., Sanyal, S., et Black, M. J. (2018). Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 725–741.

Romero, J., Tzionas, D., et Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6).

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., et Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *The IEEE International Conference on Computer Vision (ICCV)*.

Saito, S., Yang, J., Ma, Q., et Black, M. J. (2021). SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Santesteban, I., Otaduy, M. A., et Casas, D. (2019). Learning-Based Animation of Clothing for Virtual Try-On. *Computer Graphics Forum (Proc. Eurographics)*.

Santesteban, I., Otaduy, M. A., et Casas, D. (2022a). Snug: Self-supervised neural dynamic garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8140–8150.

Santesteban, I., Otaduy, M. A., Thuerey, N., et Casas, D. (2022b). ULNeF: Untangled layered neural fields for mix-and-match virtual try-on. In *Advances in Neural Information Processing Systems, (NeurIPS)*.

Santesteban, I., Thuerey, N., Otaduy, M. A., et Casas, D. (2021). Self-supervised collision handling via generative 3d garment models for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11763–11773.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Sharp, N. et Crane, K. (2018). Variational surface cutting. *ACM Transactions on Graphics (TOG)*, 37(4):1–13.

Shen, Y., Liang, J., et Lin, M. C. (2020). Gan-based garment generation using sewing pattern images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 225–247. Springer.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., et Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

Sofien, B., Sebastian, M., Tiantian, L., Ladislav, K., et Mark, P. (2014). Projective dynamics: Fusing constraint projections for fast simulation. *ACM Transactions on Graphics*, 33(4):1–11.

Stuyck, T. et Chen, H.-y. (2023). Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–14.

Su, Z., Hu, L., Lin, S., Zhang, H., Zhang, S., Thies, J., et Liu, Y. (2023). Caphy: Capturing physical properties for animatable human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14150–14160.

Tang, M., Manocha, D., et Tong, R. (2010). Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 7–13.

Thomaszewski, B., Pabst, S., et Strasser, W. (2009). Continuum-based strain limiting. In *Computer Graphics Forum*, volume 28, pages 569–576. Wiley Online Library.

Tiwari, G., Bhatnagar, B. L., Tung, T., et Pons-Moll, G. (2020). Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 1–18. Springer.

Vaxman, A., Campen, M., Diamanti, O., Panozzo, D., Bommes, D., Hildebrandt, K., et Ben-Chen, M. (2016). Directional field synthesis, design, and processing. In *Computer graphics forum*, volume 35, pages 545–572. Wiley Online Library.

Vidaurre, R., Santesteban, I., Garces, E., et Casas, D. (2020). Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Computer Graphics Forum (Proc. SCA)*.

VITRONIC Corporation (2024). VITRONIC VITUS 3D Human Solutions. https://www.vitronic.com/en-us/3d-bodyscan.

Volino, P., Courchesne, M., et Magnenat Thalmann, N. (1995). Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 137–144.

Volino, P. et Magnenat-Thalmann, N. (2005). Accurate garment prototyping and simulation. *Computer-Aided Design and Applications*, 2(5):645–654.

Volino, P., Magnenat-Thalmann, N., et Faure, F. (2009). A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics*, 28(4):Article–No.

von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., et Pons-Moll, G. (2018). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*.

Wang, H. (2018). Rule-free sewing pattern adjustment with precision and efficiency. *ACM Transactions on Graphics (TOG)*, 37(4):1–13.

Wang, H., O'Brien, J., et Ramamoorthi, R. (2010). Multi-resolution isotropic strain limiting. *ACM Transactions on Graphics (TOG)*, 29(6):1–10.

Wang, H., O'Brien, J. F., et Ramamoorthi, R. (2011). Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)*, 30(4):1–12.

Wang, K., Zhang, G., Cong, S., et Yang, J. (2023). Clothed human performance capture with a double-layer neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21098–21107.

Wang, T. Y., Ceylan, D., Popović, J., et Mitra, N. J. (2018). Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 37(6).

Wang, W., Ho, H.-I., Guo, C., Rong, B., Grigorev, A., Song, J., Zarate, J. J., et Hilliges, O. (2024). 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Z., Wu, S., Xie, W., Chen, M., et Prisacariu, V. A. (2021). Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Wolff, K., Herholz, P., Ziegler, V., Link, F., Brügel, N., et Sorkine-Hornung, O. (2021). 3d custom fit garment design with body movement. *arXiv preprint arXiv:2102.05462*, pages 1–12.

Xiang, D., Bagautdinov, T., Stuyck, T., Prada, F., Romero, J., Xu, W., Saito, S., Guo, J., Smith, B., Shiratori, T., et al. (2022). Dressing avatars: Deep photorealistic appearance for physically simulated clothing. *ACM Transactions on Graphics (TOG)*, 41(6):1–15.

Xiang, D., Prada, F., Bagautdinov, T., Xu, W., Dong, Y., Wen, H., Hodgins, J., et Wu, C. (2021). Modeling clothing as a separate layer for an animatable human avatar. *ACM Transactions on Graphics (TOG)*, 40(6):1–15.

Xu, Y., Yang, S., Sun, W., Tan, L., Li, K., et Zhou, H. (2019). 3d virtual garment modeling from rgb images. In *2019 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 37–45. IEEE.

Yang, S., Ambert, T., Pan, Z., Wang, K., Yu, L., Berg, T., et Lin, M. C. (2016). Detailed garment recovery from a single-view image. *arXiv preprint arXiv:1608.01250*.

Yu, B., Cordier, F., et Seo, H. (2024). Inverse garment and pattern modeling with a differentiable simulator. *arXiv preprint arXiv:2403.06841*.

Zhang, C., Pujades, S., Black, M. J., et Pons-Moll, G. (2017). Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4191–4200.

Zhao, F., Li, Z., Huang, S., Weng, J., Zhou, T., Xie, G.-S., Wang, J., et Shan, Y. (2023). Learning anchor transformations for 3d garment animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 491–500.

Zheng, J., Potamias, R. A., et Zafeiriou, S. (2024a). Design2cloth: 3d cloth generation from 2d masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1748–1758.

Zheng, Y., Zhao, Q., Yang, G., Yifan, W., Xiang, D., Dubost, F., Lagun, D., Beeler, T., Tombari, F., Guibas, L., et al. (2024b). Physavatar: Learning the physics of dressed 3d avatars from visual observations. *arXiv preprint arXiv:2404.04421*.

Zheng, Z., Yu, T., Wei, Y., Dai, Q., et Liu, Y. (2019). Deephuman: 3d human reconstruction from a single image. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zhou, B., Zhou, H., Liang, T., Yu, Q., Zhao, S., Zeng, Y., Lv, J., Luo, S., Wang, Q., Yu, X., et al. (2023). Clothesnet: An information-rich 3d garment model repository with simulated clothes environment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20428–20438.

Zhu, H., Cao, Y., Jin, H., Chen, W., Du, D., Wang, Z., Cui, S., et Han, X. (2020). Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 512–530. Springer.

Zou, X., Han, X., et Wong, W. (2023). Cloth4d: A dataset for clothed human reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12847–12857.

# APPENDIX

**Pattern matcher GUI.** This allows a user to compare two different clothing patterns representing the same type of garment – a pattern for a pair of shorts developed by the team and one from elsewhere, for instance. The user is shown two rows of pattern pieces, one from each pattern, and is tasked with matching the ones that correspond to the same part of the garment. After this, we use the turning-function library to quantify the similarity of the panels in each pair. The user can move the pattern pieces around the screen or rotate them for easier comparison, and color coding indicates which panels have been paired together.

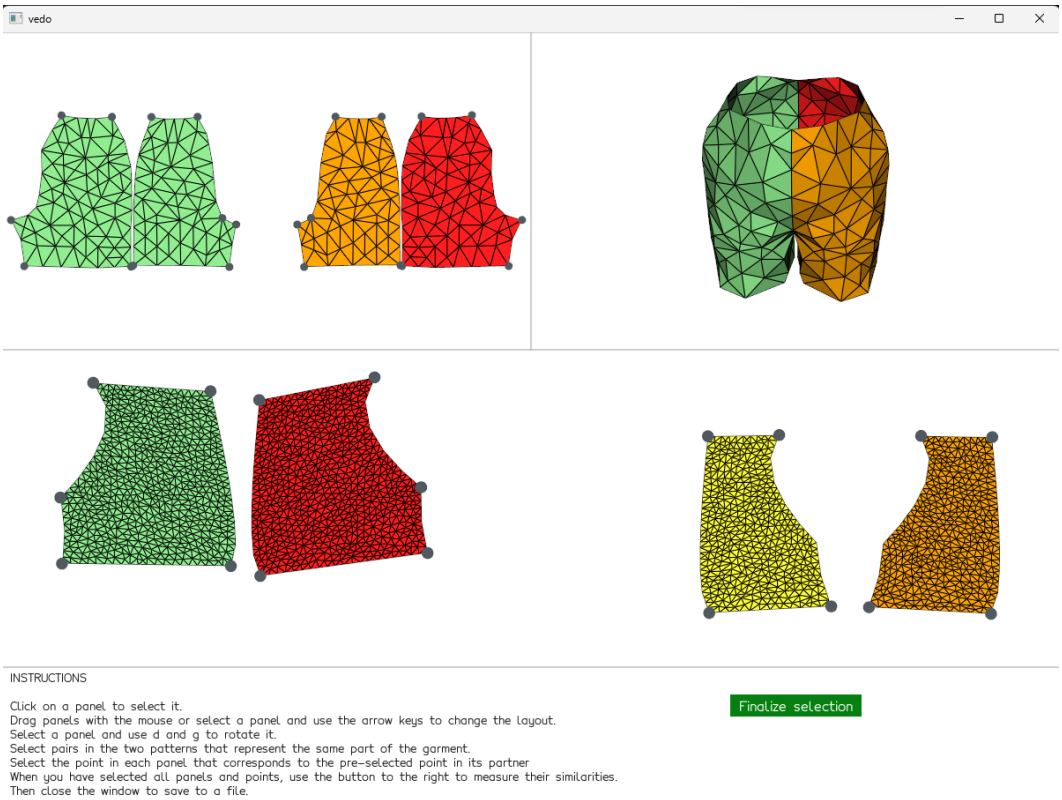


**Figure 5.1:** Pattern matcher GUI.

**Algorithm 2:** Pattern symmetry detection

**Input:** $C = \{C_p\}$, where $C_p$ is an ordered set of control points of panel $p$

**Output:** $K$: Effective control points

$InterSym$: A set of panel pairs in correspondence and the orthogonal matrix $\{(C_p, C_{p'}, M_p)\}$, so that $c_p^i \in C_p$ corresponds to $c_{p'}^i \in C_{p'}$ and $(C_p \times M) \simeq C_{p'}$.

$IntraSym$: A set of control point pairs in mirror symmetry and the reflection matrix per panel $\{(Q, Q', M)\}$ such that $(Q \times M) \simeq Q'$.

`SymmetryDetection(`$C$`)`:

$K, InterSym \leftarrow$ `InterSymmetry(`$C$`)`

$IntraSym \leftarrow \emptyset$

**foreach** $C_p \in K$ **do**

  $(Q, Q', M) \leftarrow$ `IntraSymmetry(`$C_p$`)`

  **if** $(Q, Q', M) \neq (Null)$ **then**

    1. $IntraSym \leftarrow IntraSym \cup (Q, Q', M)$

    2. $K \leftarrow K \setminus Q'$

  **end**

**end**

**return** $K, InterSym, IntraSym$

`InterSymmetry(`$C$`)`:

$K \leftarrow C, InterSym \leftarrow \emptyset$

**foreach** $C_p$ *in* $K$ **do**

  $error_p \leftarrow \infty, M_p \leftarrow I, p' \leftarrow -1$

  **foreach** $C_q$ *in* $K$ **do**

    1. $C_q^r \leftarrow$ `ReversePointsOrder(`$C_q$`)`

    2. $error_{tmp}, M_{tmp}, C_{tmp} \leftarrow$ `RigidAlignment(`$C_p, C_q^r$`)`

    **if** $error_{tmp} < \epsilon$ ***and*** $error_{tmp} < error_p$ **then**

      $p' \leftarrow q, error_p \leftarrow error_{tmp}, M_p \leftarrow M_{tmp}, C_p \leftarrow C_{tmp}$

    **end**

  **end**

  **if** $p' \neq -1$ **then**

    1. $InterSym \leftarrow InterSym \cup (C_p, C_{p'}, M_p)$

    2. $K \leftarrow K \setminus C_{p'}$

  **end**

**end**

**return** $K, InterSym$

`IntraSymmetry(`$C_p$`)`:

$l \leftarrow |C_p|$

**for** $i = 0 \ldots l/2$ **do**

  $U \leftarrow \left\{c_p^u\right\}, u = \{i \ldots i + \lfloor l/2 \rfloor - 1\}$

  $V \leftarrow \left\{c_p^v\right\}, v = \{(i + \lceil l/2 \rceil)\%l \ldots (i + l - 1)\%l\}$

  $V^r \leftarrow$ `ReversePointsOrder(`$V$`)`

  $error, M, U' \leftarrow$ `RigidAlignment(`$U, V^r$`)`

  **if** $error < \epsilon$ **then**

    **return** $U, U', M$

  **end**

  **else**

    **return** $Null$

  **end**

**end**

`RigidAlignment(`$C_1, C_2$`)`:

$l \leftarrow |C_2|, error_{min} \leftarrow \infty, M^* \leftarrow I, C_2^* \leftarrow C_2$

**for** $i = 0 \ldots l$ **do**

  $C_2 \leftarrow$ `Concatenate(`$C_2[i:], C_2[0:i]$`)`

  $error, matrix \leftarrow$ `OrthogonalProcrustes(`$C_1, C_2$`)`

  **if** $error < error_{min}$ **then**

    $M^* \leftarrow matrix, C_2^* \leftarrow C_2, error_{min} \leftarrow error$

  **end**

**end**

**return** $error_{min}, M^*, C_2^*$

**Algorithm 3:** Axial distance computation among open contours.

**Input:** $\{B_k\}$: Bone vectors from SMPL joints
$\{O^s\}$: Open contours on a source mesh $S$
$\{O^t\}$: Open contours on a target mesh $T$
**Output:** $\{O^s, O^t, d^{st}\}$: A set of open contour pairs ($O^s \in S, O^t \in T$) and their distance along the skeleton

`MeasureDistanceAlongBones(`$\{B_k\}$, $\{O^s\}$, $\{O^t\}$`):`
  **foreach** $O^s$ **do**
    $\{L_i^s\} \leftarrow$ `EncircledBones(`$O^s, \{B_k\}$`)`
  **end**
  **foreach** $O^t$ **do**
    $\{L_j^t\} \leftarrow$ `EncircledBones(`$O^t, \{B_k\}$`)`
  **end**
  $MAP \leftarrow \emptyset$
  **foreach** $(O^s, O^t)$ *pair* **do**
    $MAP^{st} \leftarrow \emptyset$
    **foreach** $(L_i^s, L_j^t)$ *pair* **do**
      **if** `bodypartname(`$B_i^s$`) == bodypartname(`$B_j^t$`)` **then**
        $MAP^{st} \leftarrow MAP^{st} \cup \{(O^s, O^t, L_i^s, L_j^t)\}$
      **end**
    **end**
    $MAP \leftarrow MAP \cup \{MAP^{st}\}$
  **end**
  **foreach** $MAP^{st}$ *in* $MAP$ **do**
    1. Take $\{(L_i^s, L_j^t)\}$ from $MAP^{st}$
    2. $i^* \leftarrow \arg\min_i \|h_i^s \in L_i^s\|$
    3. Take $P_{i^*}^s, P_{j^*}^t$ from $(L_{i^*}^s, L_{j^*}^t)$
    4. $d(O^s, O^t) \leftarrow$ `distToRoot(`$P_{j^*}^t$`) - distToRoot(`$P_{i^*}^s$`)`
  **end**
  **return** $\{O^s, O^t, d^{st} = d(O^s, O^t)\}$

`EncircledBones(`$O, \{B_k\}$`):`
  **Input:** $\{B_k\}$: Bone vectors from SMPL joints
$O$: Open contour on a mesh
**Output:** $L = \{L_i\}$, where $L_i = \{(B_i, u_i, P_i, h_i)\}$, and $B_i$ is a bone vector enclosed by $O$, $u_i$ is the axial distance, and $h_i$ is the perpendicular distance of the contour center to $B_i$

  $L \leftarrow \emptyset$, $O_{center} \leftarrow \text{center}(O)$
  **foreach** $B_i$ *in* $\{B_k\}$ **do**
    1. $J_i^{parent}, J_i^{child} \leftarrow$ Two joints of bone vector $B_i$
    2. $u_i \leftarrow \dfrac{\overrightarrow{J_i^{parent} J_i^{child}} \cdot \overrightarrow{J_i^{parent} O_{center}}}{\|\overrightarrow{J_i^{parent} J_i^{child}}\|^2}$ // axial distance from the parent
    3. $P_i \leftarrow J_i^{parent} + u_i \times \overrightarrow{J_i^{parent} J_i^{child}}$ // projection of $O_{center}$ onto the bone
    4. $h_i \leftarrow \|O_{center} - P_i\|$ // distance to the bone
    **if** $0 < u_i < 1$ **then**
      $L_i \leftarrow \{B_i, u_i, P_i, h_i\}$
      $L \leftarrow L \cup L_i$
    **end**
  **end**
  **return** $L = \{L_i\}$

# List of Figures

# List of Tables

**UNIVERSITE DE STRASBOURG**

**ECOLE DOCTORALE MATHEMATIQUES SCIENCES DE L'INFORMATION ET DE L'INGENIEUR**

# RESUME DE LA THESE DE DOCTORAT

Discipline : Informatique
Spécialité (facultative) :

Présentée par : Yu Boyang
*(Nom Prénom du candidat)*

**Titre : Reconstruction de haute qualité des modèles de vêtements et des patrons de couture à partir de données 3D d'humains vêtus.**

Unité de Recherche : ICube UMR 7357
*(N° et Nom de l'Unité)*

Directeur de Thèse : Seo Hyewon - Directrice de recherche
*(Nom Prénom – Grade)*

Co-Encadrant de Thèse: Cordier Frédéric - Maître de conférences
*(Nom Prénom – Grade)*

Localisation : Strasbourg

Thèse confidentielle : ☒NON ☐ OUI

# 1 Organisation

Le manuscrit de la thèse est rédigé en anglais et se compose de cinq chapitres, comprenant une introduction, un contexte scientifique, deux chapitres contributifs et une conclusion.

L'introduction présente la motivation générale et les contextes de la modélisation des vêtements, en commençant par un aperçu du domaine des vêtements virtuels. Le deuxième chapitre fournit un cadre de référence sur divers aspects techniques et scientifiques de la thèse. Le chapitre met particulièrement l'accent sur les problématiques ainsi que sur les approches d'optimisation précédentes, par opposition aux méthodes d'apprentissage.

Le chapitre 3 présente la première contribution, IGPM, dont l'objectif principal est de convertir une forme 3D de vêtement habillé en une représentation animable en estimant la forme précise de son patron 2D. L'approche optimise simultanément la forme du patron et les paramètres physiques de manière itérative, afin d'aligner le maillage du vêtement drapé sur le corps estimé avec la forme cible du vêtement, sans nécessiter de données d'apprentissage. Le chapitre 4 présente la deuxième contribution, visant à résoudre les principales limitations de l'approche du chapitre précédent, notamment pour les détails géométriques fins à haute fréquence et la vitesse d'optimisation. Reposant sur la même base que celle du chapitre précédent, l'approche, nommée PhyDeformer, aborde ces défis avec un algorithme de déformation de maillage par perspective.

Le manuscrit se conclut par une conclusion qui résume les contributions scientifiques, analyse certaines limitations de l'approche et propose des pistes de recherche potentielles que la thèse ouvre.

Un résumé détaillé en français du texte est présenté ci-dessous.

# 2 Contexte

Les vêtements ont toujours occupé une place essentielle dans la vie humaine à travers l'histoire. Initialement conçus comme une protection pour la peau et un moyen de conserver la chaleur, ils ont évolué, passant de l'industrie textile et vestimentaire traditionnelle aux vêtements virtuels destinés aux avatars numériques. Le concept de vêtement continue de progresser au rythme des avancées technologiques. Ces dernières années, les développements rapides dans les domaines de la réalité virtuelle (VR), de la réalité augmentée (AR) et de la téléprésence ont suscité un intérêt croissant pour la conception de vêtements virtuels alliant plausibilité, interactivité et efficacité de rendu. Par ailleurs, la disponibilité accrue de scans de vêtements portés par des humains ouvre

la voie à la création de répliques numériques issues du monde réel, mettant en lumière la nécessité pressante de récupérer tenues virtuelles réalistes adaptées à des avatars personnalisés.

Bien que l'automatisation et les processus numériquement intégrés soient largement utilisés dans le développement, la conception assistée par ordinateur (CAO) et la simulation numérique de la production de vêtements, désignés par l'approche "du virtuel au réel", un intérêt croissant se porte sur le développement d'alternatives virtuelles à partir de données issues du monde réel, autrement dit "du réel au virtuel". Cependant, cette démarche exige la récupération d'éléments vestimentaires de haute qualité, ce qui constitue un défi de taille. En effet, un système de récupération de vêtements doit être capable de reproduire fidèlement la forme et l'apparence des vêtements réels sous forme de modèles 3D numériques. Pour qu'un tel système soit efficace, il doit assurer une reconstruction précise, produire des résultats réutilisables et être capable de se généraliser à une large variété de vêtements. Afin de répondre à ces exigences, l'objectif de cette thèse est de développer de nouvelles solutions automatisées pour la récupération de vêtements numériques, visant à faciliter la création d'avatars virtuels.

Un système de récupération de vêtements performant doit répondre à plusieurs critères essentiels : la haute fidélité, afin de reproduire avec précision la géométrie 3D d'un vêtement donné; l'adaptabilité, permettant de simuler les vêtements sur diverses formes et poses; et la capacité à récupérer des modèles 2D conformes aux processus de modélisation standard, utilisés aussi bien dans l'industrie de la mode que dans les logiciels de simulation de vêtements.

# 3 L'état de l'art

La modélisation et la déformation des vêtements sont étudiées par les chercheurs depuis les années 1990. Ces dernières années, avec l'avènement de l'intelligence artificielle, de nombreuses méthodes basées sur l'apprentissage ont été développées. En général, les réseaux de neurones sont entraînés pour approximer une fonction qui prend en entrée des paramètres tels que la pose, la forme et le style de conception des vêtements, et produit le vêtement 3D correspondant. Les modèles ainsi entraînés peuvent être appliqués à la reconstruction des vêtements cibles. Cette opération est réalisée par l'optimisation des paramètres d'entrée à l'aide de gradients. Parmi les méthodes pilotées par les données, les représentations des vêtements peuvent varier considérablement. Les déplacements de sommets, utilisés comme une couche de décalage, sont courants, tandis que d'autres approches, telles que les champs implicites, les croquis ou les surfaces basées sur des patchs, ont également été explorées. Il convient de noter que, malgré ces choix, les patrons paramétriques restent très populaires en raison de leur interprétabilité et de leur contrôle explicite.

Concernant la reconstruction de vêtements, des méthodes plus géométriques ont également été proposées. Néanmoins, l'accent principal a été mis sur la déformation géométrique. Des méthodes comme Cape ou TightCap génèrent des maillages enregistrés à partir de scans 3D de haute résolution. La principale limitation de ces approches réside dans le fait qu'une représentation à une seule couche est utilisée pour modéliser à la fois le corps humain et les vêtements. De plus, comme l'enregistrement est réalisé de manière purement géométrique, les déformations du maillage ne sont pas nécessairement physiquement réalistes.

Pour reconstruire les vêtements en tant que couche distincte au-dessus du corps humain, des méthodes basées sur la segmentation ou l'enregistrement à partir de modèles de vêtements ont été proposées. En particulier, certaines d'entre elles tentent d'intégrer une simulation en temps réel dans leur pipeline, mais cela se limite souvent à un rôle de boîte noire ou à fournir une supervision faible.

Jusqu'à très récemment, la simulation différentiable a été introduite pour les tâches liées aux tissus, telles que la prise en compte des conditions externes et l'estimation des propriétés des matériaux de textiles. Quelques travaux ont tenté de l'étendre aux vêtements, mais ils partent du principe que les vêtements sont déjà disponibles et se contentent d'ajuster l'état du corps pour atteindre un certain mouvement.

Jusqu'à très récemment, la simulation différentiable a été introduite pour les tâches liées aux tissus, telles que la prise en compte des conditions externes et l'estimation des propriétés des matériaux textiles. Quelques travaux ont tenté de l'étendre aux vêtements, mais ils partent du principe que les vêtements sont déjà disponibles et se contentent d'ajuster l'état du corps pour atteindre un certain mouvement.

# 4 Contenu du mémoire

## 4.1 Paramétrisation et gradation linéaire

Il est bien connu que, même pour un type de vêtement donné, il peut exister de nombreuses variations en termes de styles et de tailles. Il serait souhaitable de les représenter dans un espace de conception restreint. Une méthode intuitive (et également la norme dans l'industrie) consiste à utiliser des modèles de patrons de couture, car cette approche permet une description efficace et homogène des vêtements partageant une topologie commune.

Dans le cadre de notre travail, nous adoptons principalement les modèles de vêtements fournis par la "UC Berkeley Garment Library". Bien qu'il soit possible d'optimiser di-

rectement les sommets du maillage du patron 2D, cette approche manque de régularisation et peut entraîner des formes mal conçues, voire des formes de repos inversées, ce qui conduit à l'échec de la simulation. Par conséquent, différentes stratégies de paramétrisation ont été adoptées dans les travaux antérieurs. Par exemple, on peut projeter les points caractéristiques du mannequin en 2D, ou utiliser des valeurs de tailleur telles que le tour de taille ou la longueur des manches. Une autre approche consiste à représenter ces paramètres sous forme de boucles de courbes de Bézier orientées.

Dans notre cas, pour chaque panneau, nous prenons les points de coin, également appelés points de gradation, qui sont des caractéristiques importantes car ils peuvent directement modifier les courbes de bordure, tandis que les sommets intérieurs ne sont que des échantillons spatiaux. De plus, à part les points de coin, où une discontinuité de la courbure locale se produit, les sommets appartenant à trois panneaux ou plus et associés ensemble lors de la couture sont également cruciaux. Ensemble, nous les utilisons comme points de contrôle.

En conséquence, pour chaque panneau du patron, la déformation peut être réalisée en manipulant les points de contrôle et en mettant à jour les bords. Les sommets intérieurs sont ensuite interpolés en appliquant les "Mean Value Coordinates" (MVC).

De plus, étant donné que la plupart des vêtements réels sont symétriques, le nombre de paramètres pourrait être davantage réduit en exploitant cette nature. Ainsi, nous commençons par détecter les panneaux appariés qui sont en inter-symétrie, de sorte qu'avec les points de contrôle d'un panneau, nous pouvons facilement restaurer leurs homologues. Ensuite, pour chaque paire symétrique, nous effectuons une détection intra-symétrique sur le premier groupe de chaque paire. Si la symétrie intra-panneau est confirmée, le nombre de points efficaces nécessaires pour restaurer tous les points de contrôle de cette paire est réduit de moitié.

En fin de compte, les paramètres ne correspondent pas à l'ensemble complet des points de contrôle, mais à un sous-ensemble des points efficaces. Cela permet une convergence plus rapide et garantit naturellement que la symétrie intrinsèque est bien préservée.

Étant donné que nous commençons toujours à partir d'un patron initial, l'écart entre celui-ci et la cible peut être très important. Par conséquent, nous cherchons à appliquer certaines déformations approximatives pour capturer la forme et la taille générales du modèle. Concrètement, à partir d'un patron de base, la forme drapée du vêtement est simulée en s'y référant. Ensuite, la mesure des distances des contours ouverts (par exemple, les contours de l'ourlet autour des hanches ou les contours des poignets sur les bras) est effectuée en 3D entre la forme drapée et la cible. En particulier, les distances entre les courbes de contour sont mesurées le long du squelette en exploitant l'arborescence des articulations du modèle corporel SMPL. Cette approche présente

l'avantage d'être insensible aux variations de pose.

Une fois les mesures effectuées, les différences de longueurs, ainsi que les circonférences des contours, sont utilisées pour ajuster les patrons 2D en modifiant les points de contrôle des panneaux avec des déplacements spécifiques. L'édition des panneaux 2D est suivie d'un processus de remaillage et de drapage pour obtenir un vêtement simulé en 3D, reflétant ainsi les modifications également dans l'espace tridimensionnel. Les résultats finaux sont présentés après cette gradation linéaire.

Cependant, un ensemble limité de distances ne peut pas entièrement décrire les différences de proportions et de formes entre les vêtements. De plus, la gradation linéaire ne prend pas en compte les effets de drapage, notamment en ignorant les phénomènes d'élasticité et d'autres déformations. Par conséquent, des étapes de raffinement sont nécessaires pour obtenir des résultats plus précis.

## 4.2 IGPM: optimisation basée sur une simulation différentiable

Un autre défi majeur consiste à désenchevêtrer la forme intrinsèque des vêtements des déformations physiques induites par les propriétés du tissu, les forces physiques et les interactions avec le corps. Nous abordons ce problème en considérant la géométrie drapée du vêtement comme le résultat de la combinaison de la forme du patron, à partir duquel le vêtement est conçu, et des interactions physiques entre le vêtement et le corps sous-jacent. Plus précisément, nous formulons ce problème comme une tâche d'estimation des paramètres de simulation des vêtements à partir de données 3D segmentées, qu'il s'agisse de modèles synthétiques d'humains habillés ou de scans 3D du monde réel.

Nous présentons donc une optimisation exploitant la simulation différentiable pour reconstruire des vêtements et générer des actifs de haute qualité, physiquement plausibles, adaptés aux applications de simulation. L'un des principaux défis actuels réside dans la grande diversité des formes et des styles vestimentaires. Pour y répondre, nous proposons d'utiliser le patron de couture paramétré comme expliqué dans la dernière section.Cette représentation unifiée permet de gérer efficacement les variations des patrons tout en facilitant le processus d'estimation.

À partir de la forme 3D cible d'un vêtement, la méthode en deux phases génère une réplique prête à être simulée en estimant précisément la forme de son patron de couture en 2D. Tout d'abord, la gradation linéaire permet de capturer les proportions et les dimensions globales, réduisant ainsi l'écart entre la géométrie cible et la forme du vêtement simulé. Ensuite, en s'appuyant sur un simulateur différentiable, la méthode exécute une simulation inverse où les patrons et les paramètres du matériau (initialisés à partir d'un ensemble préétabli de tissus) sont optimisés conjointement. La fonction

objectif globale à minimiser est regroupée en deux parties: Le premier groupe, axé sur la reconstruction, mesure la différence entre la géométrie du vêtement simulé et celle du vêtement cible. Le second groupe sert de régularisation, pénalise les incohérences des lignes de couture ainsi que les paramètres physiques non réalistes au cours de l'optimisation.

Bien que nous utilisions la simulation dynamique, nous nous concentrons sur des états proches de l'équilibre quasi-statique afin d'assurer la stabilité numérique. Notre méthode prend en compte le comportement non linéaire complexe des tissus et leur interaction détaillée avec le corps sous-jacent, en co-optisant tous les paramètres pour intégrer les interactions entre les différents composants.

Les propriétés souhaitées des habits, telles que la symétrie gauche-droite, sont préservées tout au long du processus. Les vêtements reconstruits peuvent être positionnés autour du modèle corporel du porteur et animés ultérieurement par une nouvelle simulation.

Nous avons démontré que notre méthode surpasse les travaux antérieurs selon diverses métriques, produisant des résultats de haute qualité tant dans les espaces 3D que 2D des patrons, tout en générant des actifs prêts pour la simulation. Notre approche est conçue pour être indépendante du type de vêtement, qu'il s'agisse de T-shirts, de robes ou de pantalons. Elle ne nécessite pas de jeu de données d'entraînement, intègre une compréhension des principes physiques et génère un comportement textile à la fois interprétable et plausible.

La modification des patrons permet d'adapter les vêtements à de nouvelles conditions pour des tâches en aval. Par exemple, il est possible de synthétiser de nouvelles animations de vêtements en les habillant sur différentes séquences corporelles en mouvement, et ce, sur divers sujets. Notre méthode permet également aux utilisateurs d'ajuster le design d'un vêtement capturé. Cette capacité de réanimation et de réaffectation des vêtements est particulièrement recherchée dans les applications d'essayage virtuel, de conception de vêtements et de création d'avatars.

## 4.3 PhyDeformer: optimisation par déformation basée sur les jacobians

La première pipeline présente de nombreux avantages, produisant des résultats de haute qualité tant en 3D que dans l'espace des patrons 2D, prêts pour la simulation. Cependant, en termes de coût temporel, elle peut durer plusieurs heures en fonction de la résolution du maillage. De plus, exécuter une simulation différentiable à grande échelle peut ne pas être nécessaire dans certains cas, comme pour des scénarios courants tels que l'enregistrement ou l'alignement de scans, où l'estimation des paramètres du pa-

tron et des propriétés physiques pour la compatibilité avec la simulation est moins souhaitable, et l'attention pourrait se concentrer uniquement sur la reconstruction 3D statique avec des détails fins.

Ainsi, nous réorientons un peu l'approche et proposons une nouvelle méthode similaire en deux étapes, mais visant à reproduire des détails géométriques plus fins pour répondre aux exigences de haute précision. Cette méthode permet de déformer efficacement une large variété de maillages de vêtements, rendant possibles à la fois des modifications substantielles (par exemple, un redimensionnement proportionnel) et l'ajout de détails complexes tels que des plis et des ondulations.

Dans la première étape, une gradation des vêtements est effectuée pour obtenir un alignement grossier en 3D entre le modèle de référence et le maillage cible, en prenant en compte le redimensionnement proportionnel et l'ajustement, comme dans la première pipeline. Dans la seconde étape, le maillage gradé du vêtement est affiné grâce à une déformation basée sur des jacobians par triangle, afin de correspondre aux détails fins de la cible 3D. Nous optimisons cette déformation en minimisant une fonction de coût liée à la reconstruction, qui guide la déformation vers la géométrie cible. En parallèle, une autre fonction de coût garantit que la déformation respecte la cohérence physique.

Nous montrons que cette méthode permet de mettre à jour la géométrie du vêtement pour produire des détails fins et à haute fréquence, d'améliorer la qualité de l'ajustement, ou d'être utilisée pour l'enregistrement non rigide. Cet affinement est essentiel en raison de la nature intrinsèquement non rigide de la géométrie des vêtements. De plus, les contraintes physiques introduites lors de l'étape d'affinement assurent que le résultat reste physiquement plausible. Les maillages de vêtements alignés permettent une large gamme d'applications, telles que le transfert de textures, la capture de déformations et l'estimation des paramètres physiques pour le drapage sur des avatars humains.

# 5 Conclusion

En conclusion, cette thèse propose des solutions innovantes pour automatiser la création d'avatars virtuels vêtus. Nos travaux explorent la modélisation des vêtements, la simulation différentiable et la déformation géométrique, en intégrant des fonctions de coût inspirées de principes physiques pour garantir la plausibilité. Les modèles développés génèrent des vêtements réalistes, adaptés à une large gamme d'applications, notamment en réalité virtuelle et augmentée. Dans l'ensemble, cette recherche met en évidence la valeur et l'impact des avancées dans les méthodes de récupération de scènes, permettant la production de vêtements virtuels de haute qualité et conformes aux lois physiques, tout en offrant aux professionnels des outils pertinents pour ce domaine en plein essor.

# Publication/Pré-Publication

- Boyang Yu, Frederic Cordier, Hyewon Seo. Inverse Garment and Pattern Modeling with a Differentiable Simulator accepted at Pacific Graphics 2024, (CGF 43(7)).
- Boyang Yu, Hyewon Seo. Phydeformer: High-quality Non-Rigid Garment Alignment with Physics-awareness, In progress.
- Kaifeng Zou, Boyang Yu, Hyewon Seo. 3D Facial Expression Generator Based on Transformer VAE, accepted at 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 2023, pp. 2550-2554.
- Kaifeng Zou, Sylvain Faisan, Boyang Yu, Sébastien Valette,Hyewon Seo 4D Facial Expression Diffusion Model, accepted for publication at ACM Transactions on Multimedia Computing, Communications, and Applications, published online: https://dl.acm.org/doi/ 10.1145/3653455.

# Présentations

- 6 séminaires aux ateliers du projet ANR Human4D, Lyon (11/2021), Grenoble (03/2022), Lille (09/2022), Strasbourg (03/2023), Lyon (10/2023), Grenoble (04/2024).
- 1 séminaire au lava lab, Korea Advanced Institute of Science and Technology (KAIST), Corée du Sud lors d'une courte visite (12/2022).
- 1 séminaire devant les équipes MLMS-MEMESIS (11/2023).
- 1 séminaire à 24th Pacific Conference on Computer Graphics and Applications, Huangshan (10/2024), Chine.