Inverse Garment and Pattern Modeling with a Differentiable Simulator

Boyang Yu, Frederic Cordier, Hyewon Seo

Inverse Garment and Pattern Modeling with a Differentiable Simulator

Boyang Yu [1] Frederic Cordier [2] and Hyewon Seo[1]

[1] ICube laboratory, CNRS–University of Strasbourg, France
[2] IRIMAS, University of Haute-Alsace, France

Abstract

The capability to generate simulation-ready garment models from 3D shapes of clothed people will significantly enhance the interpretability of captured geometry of real garments, as well as their faithful reproduction in the digital world. This will have notable impact on fields like shape capture in social VR, and virtual try-on in the fashion industry. To align with the garment modeling process standardized by the fashion industry and cloth simulation software, it is required to recover 2D patterns, which are then placed around the wearer's body model and seamed prior to the draping simulation. This involves an inverse garment design problem, which is the focus of our work here: Starting with an arbitrary target garment geometry, our system estimates 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. Experimental results on various real-world and synthetic data show that our method outperforms state-of-the-art methods in producing both high-quality garment models and accurate 2D patterns.

1. Introduction

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 paper, we address the challenging problem of converting a given 3D shape of a 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, our approach does not require training data and is capable of faithfully replicating intricate garment shapes. We evaluate our approach across various garment types and demonstrate that our method produces patterns and its 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.

In summary, our contributions are as follows:

- A new formulation of inverse pattern modeling based on a physics-based differentiable simulator, producing output that aligns with the current garment modeling and fabrication process.
- A compact parameterization of the sewing pattern to streamline the optimization, ensuring the preservation of desirable pattern properties;
- An upgraded differentiable simulator [LLK19, NSO12] aimed at improving computation speed, operating seamlessly within an end-to-end process.

Our generated data and code will be made available for research purposes at https://github.com/MLMS-CG/IGPM.

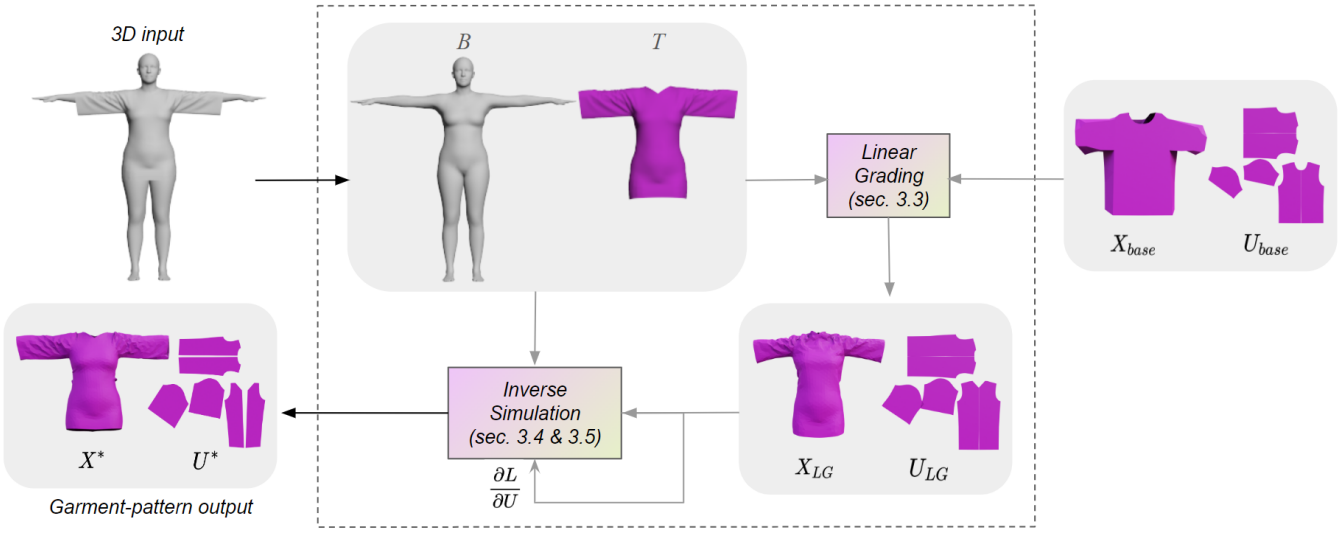

Figure 1: *Given a 3D shape of a dressed garment, our method generates a simulation-ready garment by estimating its precise 2D pattern shape. The garment mesh T is used as the target in two-phase garment fitting process, where the sewing pattern and material parameters are jointly optimized. First, a linear grading captures the overall proportions and sizes geometrically. This is followed by an inverse simulation, where optimization coupled with a differentiable simulation refine the fit and other geometric details.*

2. Related work

2.1. Garment shape recovery from 2D/3D data

Current methods, such as NeRF [WWX*21, XAS21] or PIFu [SH*19] reconstruct 3D mesh of a dressed person from one or more 2D images. However, the generated watertight meshes tend to be coarse and can not be directly draped or simulated on a body, even after a 3D segmentation to separate the garment part from the body [BKL21, PMPHB17, TBTPM20, CPY*21, MYR*20]. For the reconstructed garment geometry to be reusable to a new wearer, the 2D pattern structure or canonical shape should be recovered to ensure the compatability with existing physics-based or neural simulators. This presents a notable challenge, which we aim to tackle in this work and has been addressed by a few. For example, in [YPA*18], search-based optimization is utilized to recover both 2D sewing patterns (parameterized by numerical values such as sleeve length, waist width) and the 3D garment, so that it minimizes the 2D silhouette difference between the projected 3D garment and the input image. The draped 3D garment shape is obtained by using a physics-based simulator [NSO12]. Unlike theirs, in our work the estimation of pattern parameters are coupled with the physically-based simulation process, allowing for direct and precise mapping of 3D garment error to both the pattern geometry and material parameters.

Generative models have demonstrated their fitting capacity to given 2D or 3D inputs. Typically, they solve for optimal parameters in the model to obtain the best estimation of the body and the cloth on the input data. Early models [PLPM20, MYR*20, BTTPM19] learned the 3D clothing deformation as a displacement function, often conditioned on shape, pose, and style of garment deduced from desired or given targets. Such displacement-based representation assumes one-to-one mapping between the body and the garment surfaces, primarily suited for tight-fit clothes that closely conform to the body in terms of both topology and shape. To cope with loose garments, the blend shapes and skinning weights [STOC21] are diffused for 3D points around the body surface, have been used to drive the shape of garments according to the body pose. Alternative representations have been explored, such as implicit shape models [CPA*21, BRB*19, SOTC22], patch-based surface representation [MSY*21], or articulated dense point cloud [MYTB21].

2.2. Sewing pattern estimation from 3D data

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

3D-to-2D surface flattening. Several works considered the garment mesh as a developable surface [SGC18], obtaining 2D pattern panels by cutting the garment surface into 3D patches and then flattening 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 [BKL21], along curvature directions [VCD*16, PDF*22] on the surface, or through variational surface cutting [SC18] to minimize the distortion induced by cutting and flattening [WHZ*21]. 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 [BKL21] do not account for the fabric's elasticity in the physical body-cloth interaction during the draping process, often leading a sewing pattern that cannot accurately replicate the originally designed garment.

Pattern geometry optimization. Several works regard a 3D garment as the simulation result of its 2D pattern input, which is similar in spirit to our approach. [BSK*16] introduced 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. [Wan18] 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. The cost function is designed to accommodate garment grading for various body sizes, ensuring 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. [WHZ*21] solves another problem of garment customization, by optimizing the garment shape, thereby improving the comfort of the cloth to a specific individual. The 2D pattern is generated by computing patch lines followed by geometric flattening.

Learning based estimation. The work of [KL22] 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 variety of garment design, their 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, 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. A plethora of works have followed this approach such as [CSH*24, NQZS23, HYZ*24]. In particular, SewFormer [LXL*23] 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.

2.3. Differentiable cloth simulator

Despite the significant progress in differentiable simulation [LLK19, LDW*22, LEWS22, GLNP21], no previous works have addressed the simultaneous recovery of sewing patterns and the reconstruction of garments from the inputs of clothed humans. Concurrent to our work is DiffAvatar [LCL*23], who proposes a similar underlying idea of optimizing the sewing pattern and physical parameters to reproduce the captured 3D scan. Among the several distinctive differences between our work and theirs, one notable aspect is that our approach performs a first-stage geometric approximation to capture the overall shape of the target garment, prior to optimization through an inverse simulation. Consequently, our system is less sensitive to the initial pattern. Additionally, unlike theirs, our approach preserves desirable pattern properties such as symmetry.

2.4. Neural garment draping and dynamic modeling

Neural garment models train deep neural networks to learn draping or dynamic garment behavior on the wearer's body, aiming to replace computation-intensive physics-based simulations. While early models are trained in a supervised manner over 3D garment dataset [VSGC20, BME20, ZCM22, SOC19], later models [BME22, SOC22] adopt a self-supervised approach with physics-based losses. The idea of self-supervised learning has propelled advances of recent garment models. [GTBH23] combines the graph neural network training with the self-supervision loss terms; [DLLG*23] conditions on a latent code for more generic garments draping using UDF to represent garment surfaces; [LGF24] consists of flat 2D panels defined by 2D SDF, and each panel is associated a 3D surface parameterized by the 2D panel coordinates.

3. Method

3.1. Overview

In this section, we describe our method outlined in Figure 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). The first component of our system is the linear grading which accounts for capturing the coarse geometry such as size and proportion (section 3.3). 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 3.4, section 3.5), where the simulated garment is iteratively altered along with the physical parameters.

3.2. 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. 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 [NSO12]. These base models, i.e. 2D patterns and their corresponding sewn 3D meshes (U_{base} and X_{base} in Figure 1, respectively), are available for representative garment categories. Although several methods exist for estimating a base model given a garment mesh, which we could here, 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 2 shows three base models (t-shirt/dress, pants, and skirt) used in our experiments.

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, two curves join at a control point c^i, which is typically the vertex of C^0 curvature discontinuity (*corner*

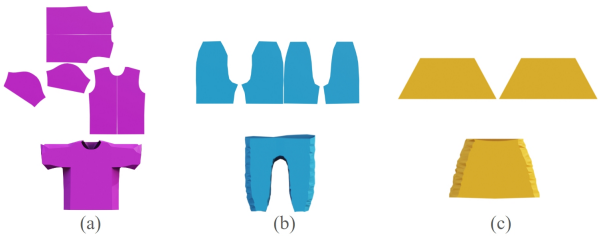

Figure 2: *Base models for three garment topologies. (a) t-shirt/dress; (b) pants; and (c) skirt.*

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 fig. 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 3.3 for the repositioning method). This effectively reduces the dimensionality of the solution space.

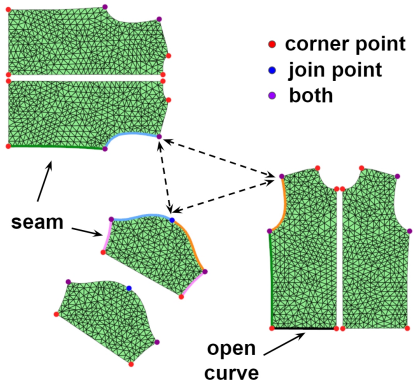

Figure 3: *Control points on the T-shirt base pattern mesh. Specific seams are colored consistently across different panels for clarity.*

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 optimization, 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 [Sch66] 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 1 of Supplementary material.

Sewn garment shape. The sewn 3D 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 [LMR*15] to represent the body, for which several efficient registration methods to 3D data exist. We used the method by [BSTPM20a, BSTPM20b] to fit SMPL parameters to the wearer's body. The resulting model is denoted as $B = SMPL(\beta^*, \theta^*)$, β and θ are respectively the shape and pose parameters in SMPL.

3.3. Pattern linear grading

In this phase, we aim to perform a preliminary 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 for an example). A 3D open contour is composed of edges connected to only one adjacent triangle, which often carry design features, representing elements 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 2 of Supplementary material.

In Figure 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 normalizing the midpoint of the

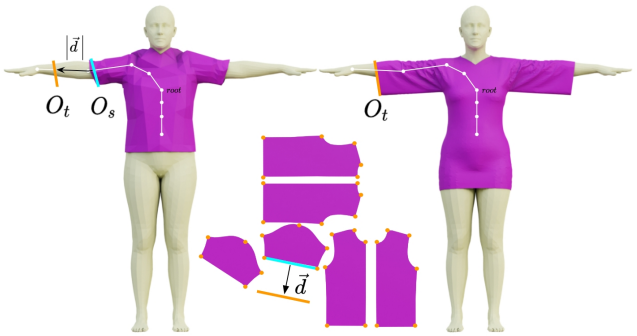

Figure 4: *An example of linear grading. The axial distance $\vec{d}$ between open contours from the source mesh (left) to the target mesh (right) is used to displace the corresponding open curve (in blue) in the 2D panel.*

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 [MMJ12]. 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. 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 3.5.

3.4. 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 [LLK19,NSO12] by revisiting both the dynamic solve and body-cloth interaction.

3.4.1. 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, which is written as:

$$(M-\Delta t^2 J)\Delta v = \Delta t(f+vJ\Delta t), \tag{1}$$

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 Δt, we solve eq. (1) for Δv 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 step (10 to 20 in our experiments), a loss $\mathscr{L}$ (Section 3.5) 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. Taking the implicit differentiation from [LLK19], we use the analytical derivatives of the linear solver to compute $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{M}}}$ and $\frac{\partial \mathscr{L}}{\partial \hat{\mathbf{f}}}$ with the gradients $\frac{\partial \mathscr{L}}{\partial \mathbf{a}}$ backpropagated from $\mathscr{L}$.

3.4.2. Material model

We employ the linear orthotropic stretching model [SB12] to quantify the extent of planar internal forces in response to cloth deformation. The model defines the relation between stress σ and strain ϵ using a constant stiffness matrix H: $\sigma = H\epsilon$, where

$$H=\begin{bmatrix} H_{00} & H_{01} & 0 \\ H_{01} & H_{11} & 0 \\ 0 & 0 & H_{22} \end{bmatrix}. \tag{2}$$

The bending forces are modeled with piecewise dihedral angles which describe how much the out-of-plane forces would be when subject to cloth bending, as used in [BMF05]:

$$\mathbf{f}_i = k\frac{\|\mathbf{e}\|^2}{\|\mathbf{A}_1\|+\|\mathbf{A}_2\|}\sin\left(\frac{\pi-\alpha}{2}\right)\mathbf{u}_i, \tag{3}$$

where α is the dihedral angle, $\mathbf{e}$ is the edge vector, $\mathbf{f}_i$ represents the bending force applied on the i-th vertex (i=1,...,4), $\mathbf{A}_1$ and $\mathbf{A}_2$ 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.

3.4.3. Acceleration of force vector/Jacobian matrix assembly

ARCSim [NSO12,LLK19] 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 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 5 (e), where we use mn to denote $\frac{\partial f_m}{\partial x_n}$ (m, n: global indices) for a compact representation.

In ARCSim [LLK19, NSO12], 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 5(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 5(d)), and by encoding the mapping from the per-element Jacobian to the compressed Jacobian vector as a static mapping matrix (Figure 5(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 1). Finally, the Jacobians in the compressed vector are transferred to $\hat{\mathbf{M}}$ (Figure 5(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.

3.4.4. Efficient body cloth interaction

One important component for 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 [BMF05, TMT10, HVTG08]. In particular, the one based on non-rigid impact zones [HVTG08] has been made differentiable by Liang et al [LLK19]. 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 light 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:

$$\tilde{x} = x + (\delta - sdf(x)) \cdot \mathbf{n}, \tag{4}$$

where $\mathbf{n} = \frac{\nabla sdf(x)}{|\nabla sdfx|}$ is the spatial gradient of $sdf(\cdot)$ (also the surface normal), and δ 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 [NVI]). In the latter case, as it involves only unit operations, it can also be seamlessly integrated in 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.

3.5. 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 points K and the physical parameters Γ:

$$\begin{aligned}\mathscr{L} = \mathscr{L}_{rec}(X = Sim(U(K), \Gamma; SMPL(\theta, \beta)), T) \\ + \lambda_{seam}\mathscr{L}_{seam}(U(K)),\end{aligned} \tag{5}$$

where λ'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}),

$$\mathscr{L}_{rec} = \mathscr{L}_{CF}(X, T) + \lambda_{open}\mathscr{L}_{CF}(X_{open}, T_{open}) + \lambda_{mat}\mathscr{L}_{mat}(\Gamma). \tag{6}$$

The Chamfer distance, widely used for fitting deformable surfaces, has proven to work well in most cases. 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 6). Hence, we use curvature-weighted Chamfer distance [BRPW22] instead, which prioritizes high-curvature regions, subsequently improving the reconstruction of densely folded regions:

$$\begin{aligned}\mathscr{L}_{CWCF}(X, T) = \frac{1}{|X|}\sum_{x \in X} \kappa(\tilde{t}) \min_{t \in T} \|x - t\|^2 \\ + \frac{1}{|T|}\sum_{t \in T} \kappa(t) \min_{x \in X} \|t - x\|^2,\end{aligned} \tag{7}$$

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

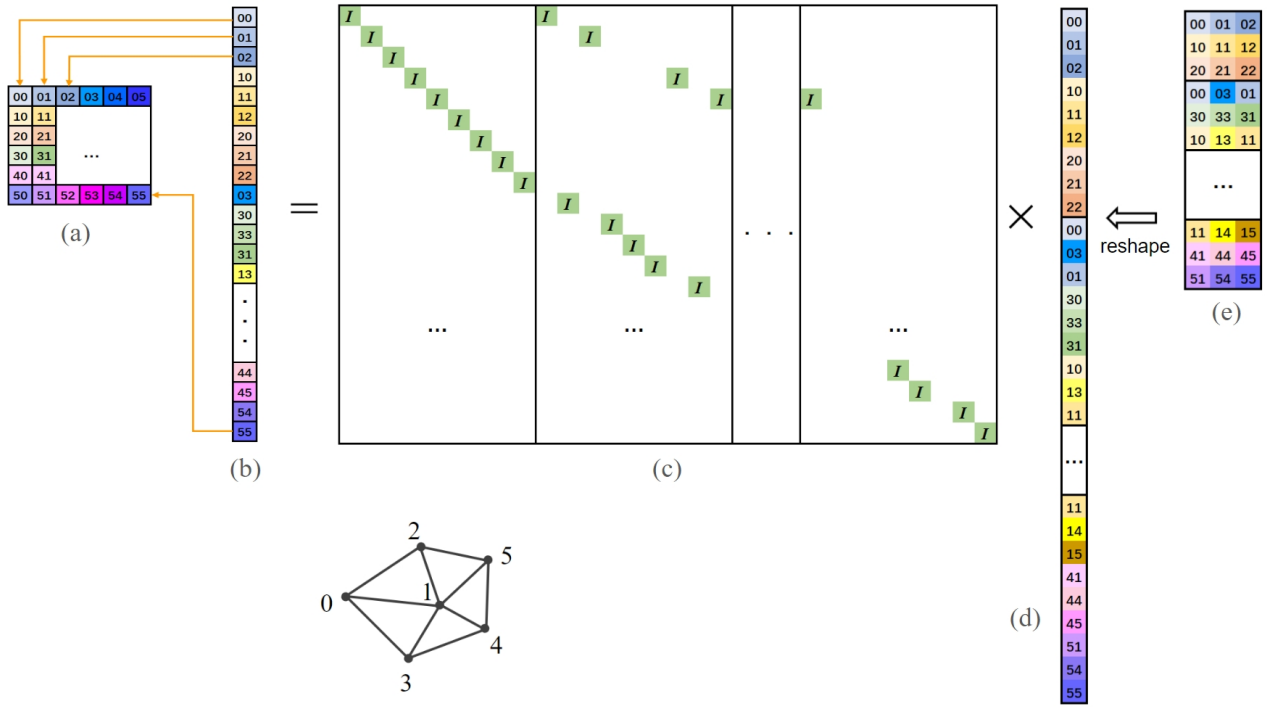

Figure 5: *The assembly of a compressed Jacobian vector (b) is obtained by a static matrix multiplication, encoding the mapping from the per-element Jacobian (e) to the per force-node Jacobian (b). Compressed Jacobians are then transferred to the extended mass matrix (a) .*

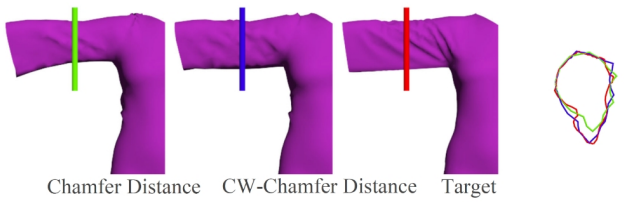

Figure 6: *The cross-sectional curves on the right, generated from three meshes on the left: two simulated meshes, and the target. The colors of the cutting planes are used to draw the cross-sections. The curvature-weighted Chamfer (blue) leads to a sleeve draping silhouette closer to the ground truth (red), compared to the standard Chamfer distance (green).*

$$L_{seam}(U) = \sum_{i}^{|S|} \sum_{j}^{|E_i|=|E_i^{corr}|} \left(|e_i^j| - |e_i^{corr,j}| \right)^2, \tag{8}$$

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 H and k by adding them into the variable set:

$$\Gamma := (H_{00}, H_{01}, H_{11}, H_{22}, k). \tag{9}$$

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.

4. Experiments

4.1. Implementation details

We now describe the main implementation details. Further information is provided in the supplementary material.

Simulation. We set one time step Δt 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 [MQK*21] with periodic activation functions [SMB*20] 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 [LLK19] allows it to run all computations on a GPU. Table 1 summarizes the computation time of linear solve by the baseline model [LLK19] 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 [LLK19].

Table 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.*

	Baseline [LLK19]		Ours		
iterations / # verts	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

Optimization. The variables are optimized by using the Adam optimizer [KB15], 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.

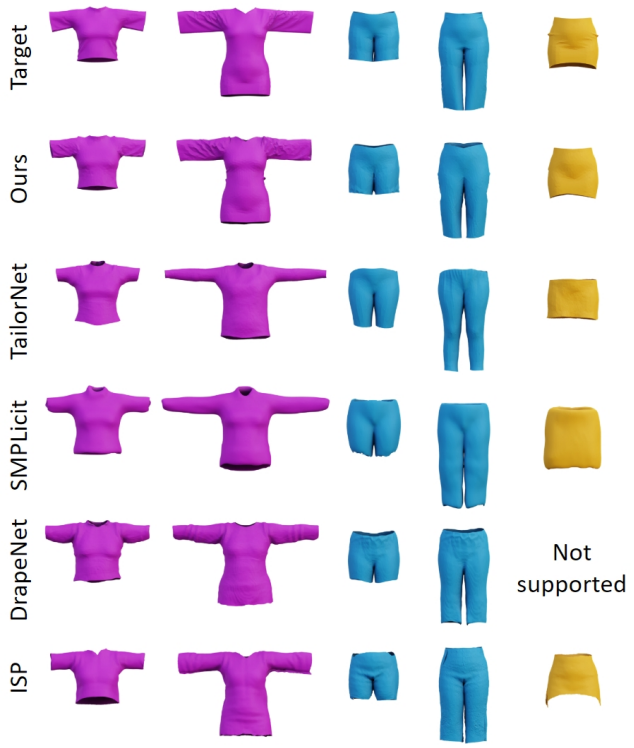

Figure 7: *Comparison of 3D garment reconstruction of our method with others [PLPM20, CPA*21, DLLG*23, LGF24]. Our method reproduces faithful garment shapes, even accounting for intricate geometry details like wrinkles on large sleeves. Best viewed on screen zoomed-in.*

4.2. 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 dataset [KL21] as targets (i.e. test data), which is an unseen dataset for all evaluated methods. It is also one of the rare 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 [KB15] 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 our method, the γ garment style parameter for TailorNet [PLPM20], the latent vector $z = [z_{cut}, z_{style}]$ describing garment cut and style for SMPLicit [CPA*21], and the latent codes z's encoding the garment characteristics in DrapeNet and ISP [DLLG*23, LGF24]. The results are quantitatively evaluated using two metrics: Chamfer distance to the ground truth mesh vertices, and the angular error to measure the similarity of the computed normal vectors, similar to [BPG*20]. As shown in Table 2, our method is consistently better than others, which is confirmed by qualitative results, shown in Figure 7. 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 the optimization.

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 [Wan18, BSK*16]. We compare our work with NeuralTailor [KL22] and SewFormer [LXL*23], 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

Table 2: *Quantitative evaluation in 3D (garment reconstruction) and 2D (pattern estimation). To measure the accuracy of 2D patterns, we evaluate the turning function metric for comparing polygonal shapes [ACH*91] and the surface error (the average of normalized surface difference error computed for each patch).*

	3D Reconstruction					2D Estimation		
	Chamfer distance / Normal similarity					Turning /Surface area		
Garments	SMPLicit	TailorNet	Drapenet	ISP	Ours	NeuralTailor	SewFormer	Ours
T-shirt	1.4 /-	0.331 / 0.081	0.689 / 0.129	0.297 / 0.094	**0.112 / 0.049**	10.50 / 0.13	10.46 / 0.15	**9.12 / 0.09**
Dress	3.2 / -	1.305 / 0.161	0.619 / 0.135	0.189 / 0.131	**0.110 / 0.075**	11.20 / 0.37	11.77 / 0.43	**10.96 / 0.10**
Shorts	1.3 / -	1.036 / 0.050	0.131 / 0.048	0.202 / 0.095	**0.126 / 0.043**	7.41 / 0.05	7.74 / 0.06	**7.33 / 0.04**
Pants	2.9 / -	2.587 / 0.104	0.485 / 0.085	0.185 / 0.077	**0.142 / 0.049**	**6.77 / 0.01**	6.79 / 0.20	7.00 / 0.08
Skirt	6.5 / -	1.300 / 0.063	- / -	0.435 / 0.093	**0.106 / 0.014**	4.31 / 1.11	4.42 / 0.62	**4.14 / 0.04**

used in NeuralTailor as the ground-truth ones. Then, we generate 3D drape shapes at a T-pose by using an independent simulator [Auto], 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 illustrated in Figure 8. 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 [ACH*91], and (2) the relative error in surface area, 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.

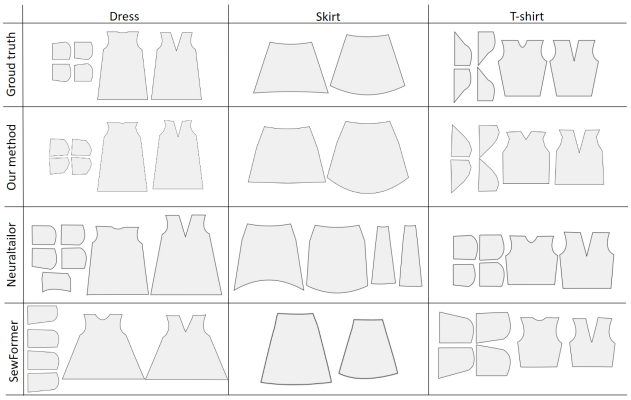

Figure 8: *2D pattern estimation from 3D garment meshes. From top to bottom: Ground truth, ours, NeuralTailor [KL22] and SewFormer [LXL*23].*

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 [LXL*23]. As shown in Figure 9, 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.

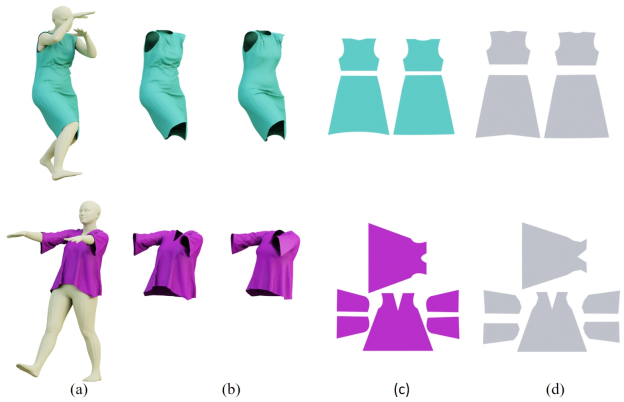

Figure 9: *Results of our method evaluated using varying poses. (a) Target clothed body; (b) target garment mesh (left) and the reconstructed mesh (right); (c) ground-truth pattern; and (d) our estimated pattern.*

4.3. 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 [KL22]. As shown in Figure 10, 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 a significantly different pattern for each target instance.

4.4. 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

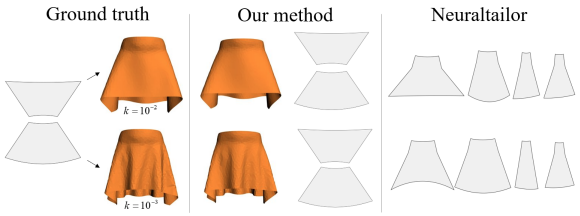

Figure 10: *Sewing patterns estimated from two input meshes, both simulated from an identical ground-truth pattern but with varying bending coefficients. By optimization over bending coefficient, our method correctly finds the panel shapes, compared to the alternative method.*

scanner [VIT], 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 11, 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 12. More results are provided in the accompanying video.

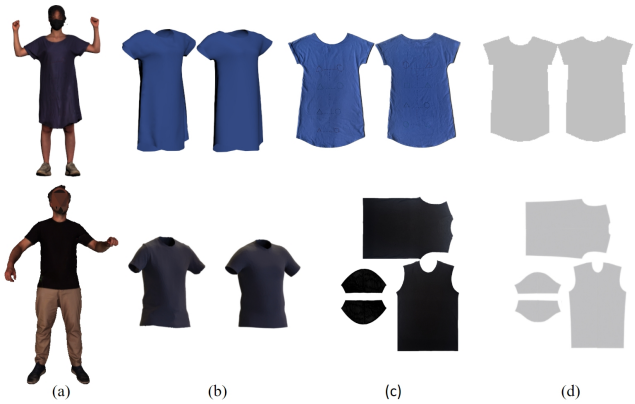

Figure 11: *Results of our method evaluated using 3D scan data. (a) Input 3D scan; (b) segmented target (left) and simulated garments (right); (c) ground-truth pattern; and (d) estimated pattern.*

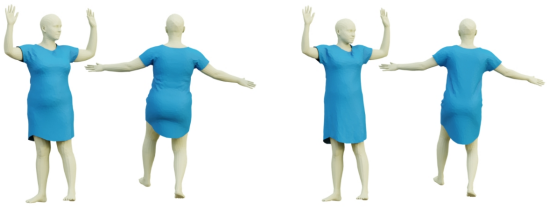

Figure 12: *The dress model obtained from the 3D scan (Figure 11) has been retargated to two new SMPL bodies.*

5. 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). Our model achieved a Chamfer distance precision (CF) of 0.1103 (in *mm*) 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), when the seam consistency loss term is removed (2nd row), or when the optimization of physical parameters was disabled (3rd row). These results confirm the importance of both loss terms and the integration of physical properties in the optimization process.

Figure 13 illustrates the reconstructed models obtained from the ablation study. We observe that our model (Figure 13(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 13(b)). The absence of seam loss leads to a puckered seam around the shoulder, resulting from the extra tension exerted on the the shorter seam (Figure 13(c)). The optimization of physical parameters helps to recover fine wrinkles, as well as more plausible pattern estimation. As we can see in Figure 13(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.

Table 3: *Ablation study.*

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**

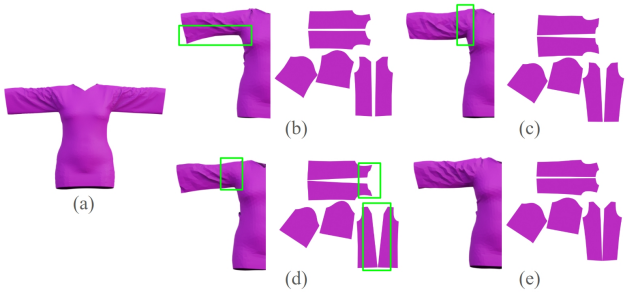

Figure 13: *Results of our ablation study on loss terms: target mesh (a); results with vanilla Chamfer loss (without curvature weights (b), without seam loss (c), and without physical parameters (d); our results (e).*

6. 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 [JYK*23]. 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. Finally, our method is unable to handle cloth self-collision and multi-layer garment inputs, which we leave as future work.

References

[ACH*91] ARKIN E., CHEW L., HUTTENLOCHER D., KEDEM K., MITCHELL J.: An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 3 (1991), 209–216. doi:10.1109/34.75509. 9

[Auto] AUTODESK: 3ds max, o. 3dsMax. URL: https://www.autodesk.com/products/3ds-max/. 9

[BKL21] BANG S., KOROSTELEVA M., LEE S.-H.: Estimating garment patterns from static scan data. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 273–287. 2

[BME20] BERTICHE H., MADADI M., ESCALERA S.: Pbns: physically based neural simulator for unsupervised garment pose space deformation. *arXiv preprint arXiv:2012.11310* (2020). 3

[BME22] BERTICHE H., MADADI M., ESCALERA S.: Neural cloth simulation. *ACM Trans. Graph. 41*, 6 (nov 2022). doi:10.1145/3550454.3555491. 3

[BMF05] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*. 2005, pp. 3–es. 5, 6

[BPG*20] BEDNARIK J., PARASHAR S., GUNDOGDU E., SALZMANN M., FUA P.: Shape reconstruction by learning differentiable surface representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 4716–4725. 8

[BRB*19] BUFFET T., ROHMER D., BARTHE L., BOISSIEUX L., CANI M.-P.: Implicit untangling: A robust solution for modeling layered clothing. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 2

[BRPW22] BONGRATZ F., RICKMANN A.-M., PÖLSTERL S., WACHINGER C.: 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* (2022), pp. 20773–20783. 6

[BSK*16] BARTLE A., SHEFFER A., KIM V. G., KAUFMAN D. M., VINING N., BERTHOUZOZ F.: Physics-driven pattern adjustment for direct 3d garment editing. *ACM Trans. Graph. 35*, 4 (2016), 50–1. 3, 8

[BSTPM20a] BHATNAGAR B. L., SMINCHISESCU C., THEOBALT C., PONS-MOLL G.: Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)* (aug 2020), Springer. 4

[BSTPM20b] BHATNAGAR B. L., SMINCHISESCU C., THEOBALT C., PONS-MOLL G.: Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)* (December 2020). 4

[BTTPM19] BHATNAGAR B. L., TIWARI G., THEOBALT C., PONS-MOLL G.: Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 5420–5430. 2

[CPA*21] CORONA E., PUMAROLA A., ALENYÀ G., PONS-MOLL G., MORENO-NOGUER F.: Smplicit: Topology-aware generative model for clothed people. In *CVPR* (2021). 2, 8

[CPY*21] CHEN X., PANG A., YANG W., WANG P., XU L., YU J.: Tightcap: 3d human shape capture with clothing tightness field. *ACM Trans. Graph. 41*, 1 (nov 2021). doi:10.1145/3478518. 2

[CSH*24] CHEN C.-H., SU J.-W., HU M.-C., YAO C.-Y., CHU H.-K.: Panelformer: Sewing pattern reconstruction from 2d garment images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2024), pp. 454–463. 3

[DLLG*23] DE LUIGI L., LI R., GUILLARD B., SALZMANN M., FUA P.: Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 1451–1460. 3, 8

[GLNP21] GUO J., LI J., NARAIN R., PARK H.: Inverse simulation: Reconstructing dynamic geometry of clothed humans via optimal control. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2021), IEEE Computer Society, pp. 14693–14702. 3

[GTBH23] GRIGOREV A., THOMASZEWSKI B., BLACK M. J., HILLIGES O.: HOOD: Hierarchical graphs for generalized modelling of clothing dynamics. 3

[HVTG08] HARMON D., VOUGA E., TAMSTORF R., GRINSPUN E.: Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*. 2008, pp. 1–4. 6

[HYZ*24] HE K., YAO K., ZHANG Q., YU J., LIU L., XU L.: Dresscode: Autoregressively sewing and generating garments from text guidance. *arXiv preprint arXiv:2401.16465* (2024). 3

[JYK*23] JANG J., YUN W.-H., KIM W. H., YOON Y., KIM J., LEE J., HAN B.: Learning to boost training by periodic nowcasting near future weights. In *Proceedings of the 40th International Conference on Machine Learning* (23–29 Jul 2023), Krause A., Brunskill E., Cho K., Engelhardt B., Sabato S., Scarlett J., (Eds.), vol. 202 of *Proceedings of Machine Learning Research*, PMLR, pp. 14730–14757. 11

[KB15] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)* (2015). URL: http://arxiv.org/abs/1412.6980. 8

[KL21] KOROSTELEVA M., LEE S.-H.: Generating datasets of 3d garments with sewing patterns. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* (2021), Vanschoren J., Yeung S., (Eds.), vol. 1. 8

[KL22] KOROSTELEVA M., LEE S.-H.: Neuraltailor: reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–16. 3, 8, 9

[LCL*23] LI Y., CHEN H.-Y., LARIONOV E., SARAFIANOS N., MATUSIK W., STUYCK T.: Diffavatar: Simulation-ready garment optimization with differentiable simulation. *arXiv preprint arXiv:2311.12194* (2023). 3

[LDW*22] LI Y., DU T., WU K., XU J., MATUSIK W.: Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics (TOG) 42*, 1 (2022), 1–20. 3

[LEWS22] LARIONOV E., ECKERT M.-L., WOLFF K., STUYCK T.: Estimating cloth elasticity parameters using position-based simulation of compliant constrained dynamics. *arXiv preprint arXiv:2212.08790* (2022). 3

[LGF24] LI R., GUILLARD B., FUA P.: Isp: Multi-layered garment draping with implicit sewing patterns. *Advances in Neural Information Processing Systems 36* (2024). 3, 8

[LLK19] LIANG J., LIN M., KOLTUN V.: Differentiable cloth simulation for inverse problems. *Advances in Neural Information Processing Systems 32* (2019). 1, 3, 5, 6, 7, 8

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG) 34*, 6 (2015), 1–16. 4

[LXL*23] LIU L., XU X., LIN Z., LIANG J., YAN S.: Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (TOG) 42*, 6 (2023), 1–15. 3, 8, 9

[MMJ12] MENG Y., MOK P. Y., JIN X.: Computer aided clothing pattern design with 3d editing and pattern alteration. *Computer-Aided Design 44*, 8 (2012), 721–734. 5

[MQK*21] MU J., QIU W., KORTYLEWSKI A., YUILLE A. L., VASCONCELOS N., WANG X.: A-sdf: Learning disentangled signed distance functions for articulated shape representation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 12981–12991. 7

[MSY*21] MA Q., SAITO S., YANG J., TANG S., BLACK M. J.: 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* (2021), pp. 16082–16093. 2

[MYR*20] MA Q., YANG J., RANJAN A., PUJADES S., PONS-MOLL G., TANG S., BLACK M. J.: Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 6469–6478. 2

[MYTB21] MA Q., YANG J., TANG S., BLACK M. J.: The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct. 2021), pp. 10974–10984. 2

[NQZS23] NAG S., QI A., ZHU X., SHAMIR A.: Personaltailor: Personalizing 2d pattern design from 3d garment point clouds. *arXiv preprint arXiv:2303.09695* (2023). 3

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG) 31*, 6 (2012), 1–10. 1, 2, 3, 5, 6

[NVI] NVIDIA: Kaolin library documentation. Kaolin. URL: https://kaolin.readthedocs.io/en/latest/index.html. 6

[PDF*22] PIETRONI N., DUMERY C., FALQUE R., LIU M., VIDAL-CALLEJA T., SORKINE-HORNUNG O.: Computational pattern making from 3d garment models. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–14. 2

[PLPM20] PATEL C., LIAO Z., PONS-MOLL G.: 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)* (jun 2020), IEEE. 2, 8

[PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M. J.: Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (ToG) 36*, 4 (2017), 1–15. 2

[SB12] SIFAKIS E., BARBIČ J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*. 2012, pp. 1–50. 5

[SC18] SHARP N., CRANE K.: Variational surface cutting. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–13. 2

[Sch66] SCHÖNEMANN P. H.: A generalized solution of the orthogonal procrustes problem. *Psychometrika 31*, 1 (1966), 1–10. 4

[SGC18] STEIN O., GRINSPUN E., CRANE K.: Developability of triangle meshes. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–14. 2

[SH*19] SAITO S., , HUANG Z., NATSUME R., MORISHIMA S., KANAZAWA A., LI H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *arXiv preprint arXiv:1905.05172* (2019). 2

[SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems 33* (2020), 7462–7473. 7

[SOC19] SANTESTEBAN I., OTADUY M. A., CASAS D.: Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 355–366. 3

[SOC22] SANTESTEBAN I., OTADUY M. A., CASAS D.: Snug: Self-supervised neural dynamic garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 8140–8150. 3

[SOTC22] SANTESTEBAN I., OTADUY M. A., THUEREY N., CASAS D.: ULNeF: Untangled layered neural fields for mix-and-match virtual try-on. In *Advances in Neural Information Processing Systems, (NeurIPS)* (2022). 2

[STOC21] SANTESTEBAN I., THUEREY N., OTADUY M. A., CASAS D.: 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* (2021), pp. 11763–11773. 2

[TBTPM20] TIWARI G., BHATNAGAR B. L., TUNG T., PONS-MOLL G.: 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* (2020), Springer, pp. 1–18. 2

[TMT10] TANG M., MANOCHA D., TONG R.: Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (2010), pp. 7–13. 6

[VCD*16] VAXMAN A., CAMPEN M., DIAMANTI O., PANOZZO D., BOMMES D., HILDEBRANDT K., BEN-CHEN M.: Directional field synthesis, design, and processing. In *Computer graphics forum* (2016), vol. 35, Wiley Online Library, pp. 545–572. 2

[VIT] VITRONIC: Vitronic vitus human solutions. URL: https://www.vitronic.com/en-us/3d-bodyscan. 10

[VSGC20] VIDAURRE R., SANTESTEBAN I., GARCES E., CASAS D.: Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 145–156. 3

[Wan18] WANG H.: Rule-free sewing pattern adjustment with precision and efficiency. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–13. 3, 8

[WHZ*21] WOLFF K., HERHOLZ P., ZIEGLER V., LINK F., BRÜGEL N., SORKINE-HORNUNG O.: 3d custom fit garment design with body movement. *arXiv preprint arXiv:2102.05462* (2021), 1–12. 2, 3

[WWX*21] WANG Z., WU S., XIE W., CHEN M., PRISACARIU V. A.: Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064* (2021). 2

[XAS21] XU H., ALLDIECK T., SMINCHISESCU C.: H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *Advances in Neural Information Processing Systems* (2021), Ranzato M., Beygelzimer A., Dauphin Y., Liang P., Vaughan J. W., (Eds.), vol. 34, Curran Associates, Inc., pp. 14955–14966. 2

[YPA*18] YANG S., PAN Z., AMERT T., WANG K., YU L., BERG T., LIN M. C.: Physics-inspired garment recovery from a single-view image. *ACM Trans. Graph. 37*, 5 (nov 2018). doi:10.1145/3026479. 2

[ZCM22] ZHANG M., CEYLAN D., MITRA N. J.: Motion guided deep dynamic 3d garments. *ACM Trans. Graph. 41*, 6 (nov 2022). doi:10.1145/3550454.3555485. 3

Algorithm 1 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 $c^i_p \in C_p$ corresponds to $c^i_{p'} \in C_{p'}$ 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 $(Q \times M) \simeq Q'$.

Procedure: SymmetryDetection(C)
  K, $InterSym \leftarrow$ InterSymmetry(C)
  $IntraSym \leftarrow \emptyset$
  for each $C_p \in K$ **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 if
  end for
return: $K, InterSym, IntraSym$

Procedure: InterSymmetry(C)
  $K \leftarrow C$, $InterSym \leftarrow \emptyset$
  for C_p **in** K **do**
    $error_p \leftarrow \infty$, $M_p \leftarrow I$, $p' \leftarrow -1$
    for C_q **in** K **do**
      1. $C^r_q \leftarrow$ ReversePointsOrder(C_q)
      2. $error_{tmp}, M_{tmp}, C_{tmp} \leftarrow$ RigidAlignment(C_p, C^r_q)
      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 if
    end for
    if $p' \neq -1$ **then**
      1. $InterSym \leftarrow InterSym \cup (C_p, C_{p'}, M_p)$
      2. $K \leftarrow K \setminus C_{p'}$
    end if
  end for
return: $K, InterSym$

Procedure: IntraSymmetry(C_p)
  $l \leftarrow |C_p|$
  for $i = 0...l/2$ **do**
    $U \leftarrow \{c^u_p\}$, $u = \{i...i + \lfloor l/2 \rfloor - 1\}$
    $V \leftarrow \{c^v_p\}$, $v = \{(i + \lceil l/2 \rceil)\%l ... (i + l - 1)\%l\}$
    $V^r \leftarrow$ ReversePointsOrder(V)
    $error$, M, $U' \leftarrow$ RigidAlignment(U, V^r)
    if $error < \epsilon$ **then**
      return (U, U', M)
    else
      return $(Null)$
    end if
  end for

Procedure: RigidAlignment(C_1, C_2)
  $l \leftarrow |C_2|$, $error_{min} \leftarrow \infty$, $M^* \leftarrow I$, $C^*_2 \leftarrow C_2$
  for $i = 0...l$ **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 if
  end for
return: $error_{min}, M^*, C^*_2$

Algorithm 2 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

Procedure: MeasureDistanceAlongBones($\{B_k\}$, $\{O^s\}$, $\{O^t\}$)
  for each O^s **do**
    $\{L_i^s\} \leftarrow$EncircledBones $(O^s, \{B_k\})$
  end for
  for each O^t **do**
    $\left\{L_j^t\right\} \leftarrow$ EncircledBones $(O^t, \{B_k\})$
  end for
  $MAP \leftarrow \emptyset$
  for each (O^s, O^t) pair **do**
    $MAP^{st} \leftarrow \emptyset$
    for each (L_i^s, L_j^t) pair **do**
      if bodypartname(B_i^s) == bodypartname(B_j^t) **then** //bodyparts: torso and four limbs
        $MAP^{st} \leftarrow MAP^{st} \cup \{(O^s, O^t, L_i^s, L_j^t)\}$
      end if
    end for
    $MAP \leftarrow MAP \cup \{MAP^{st}\}$
  end for
  for each MAP^{st} **in** MAP **do**
    1. take $\left\{(L_i^s, L_j^t)\right\}$ 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 for
return: $\{O^s, O^t, d^{st} = d(O^s, O^t)\}$

Procedure: EncircledBones($O, \{B_k\}$)
  Input:
  $\{B_k\}$: bone vectors from SMPL joints
  O: open contour on a mesh
  Output:
  $L = \{L_i\}$, $L_i = \{(B_i, u_i, P_i, h_i)\}$ where B_i is bone vector enclosed by O, and u_i and h_i are the axial and the perpendicular distances of the contour center in relation to B_i.

  $L \leftarrow \emptyset, O_{center} \leftarrow center(O)$
  for each B_i **in** $\{B_k\}$ **do**
    1. $J_i^{parent}, J_i^{child} \leftarrow$ Two joints of bone vector B_i
    2. $u_i \leftarrow \frac{\overrightarrow{J_i^{parent} J_i^{child}} \cdot \overrightarrow{J_i^{parent} O_{center}}}{\|\overrightarrow{J_i^{parent} J_i^{child}}\|^2}$ // axial distance from the parent
    3. $P_i \leftarrow J_i^{parent} + u * (\overrightarrow{J_i^{parent} J_i^{child}}))$ // 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 if
  end for
return: $L = \{L_i\}$