Estimating Garment Patterns from Static Scan Data

Seungbae Bang,[†] Maria Korosteleva[‡] and Sung-Hee Lee[§]

Korea Advanced Institute of Science and Technology (KAIST), Korea

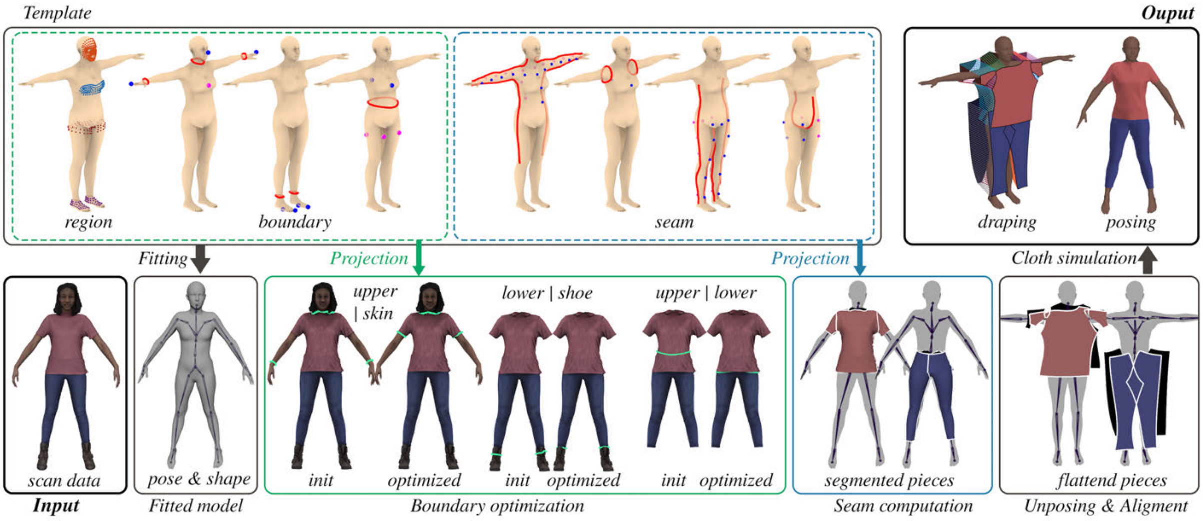

Figure 1: *Given a 3D scan mesh of a clothed person as input, our method fits a body template with predefined boundaries and seam curves to the input to obtain boundaries and seam curves of each garment. Then the mesh is segmented into pieces, which are flattened and placed to be ready for simulation.*

Abstract

The acquisition of highly detailed static 3D scan data for people in clothing is becoming widely available. Since 3D scan data is given as a single mesh without semantic separation, in order to animate the data, it is necessary to model shape and deformation behavior of individual body and garment parts. This paper presents a new method for generating simulation-ready garment models from 3D static scan data of clothed humans. A key contribution of our method is a novel approach to segmenting garments by finding optimal boundaries between the skin and garment. Our boundary-based garment segmentation method allows for stable and smooth separation of garments by using an implicit representation of the boundary and its optimization strategy. In addition, we present a novel framework to construct a 2D pattern from the segmented garment and place it around the body for a draping simulation. The effectiveness of our method is validated by generating garment patterns for a number of scan data.

CCS Concepts

• ***Computing methodologies*** → ***Shape modeling; Parametric curve and surface models;***

1. Introduction

Rapidly advancing 3D scanning techniques allow for acquiring an accurate, static 3D shape of a user with ease. In general, the 3D scan data is given as a single mesh showing only the visible part of the surfaces of the subject: the scan data is lacking in division of different parts, such as skin and clothes, let alone semantic labelling of these parts. This simple 3D model is still useful for 3D printing or populating static virtual scenes, but must be converted to an animatable form to be used in wider applications, such as movies and computer games. For this, one needs to estimate and rig the body shape of the subject including occluded parts of the body, and reconstruct individual parts of the worn garment.

This remains as a challenging task, and many researchers have

[†] jjapagati@gmail.com
[‡] mariako@kaist.ac.kr
[§] sunghee.lee@kaist.ac.kr

approached this problem from a number of perspectives. In this paper, we present a novel method to segment garment parts from input 3D scan data of triangulated mesh and obtain a body model with a simulatable garment by constructing the 2D pattern for each garment part.

A popular method for segmenting a garment part from a 3D scan or image data is to use graph-based (e.g., Markov Random Field) segmentation approach, in which the label of each node is determined individually based on its own priors and the relation with its neighbors. As a per-node classification approach, it often results in garment segmentation that contains miss-classified nodes inside or on the boundary of the garment. Recent deep learning-based parsing methods achieve greatly improved accuracy but are not free from this limitation.

To overcome this limitation, we developed a novel approach for the garment segmentation. Instead of the per-node classification, we sought to explicitly find the boundaries of garment parts. This boundary-based segmentation scheme can prevent artifacts such as holes and jagged boundaries in the garment segmentation. However, it is highly complex to explicitly define controllable curves on the surface. Instead, inspired by [SC18b], we represent the curve with an implicit function and developed a novel method to update the implicit function to drive the curve towards the boundary.

In general, the garment modeling process involves making 2D patterns, which are then stitched and simulated to create a 3D garment model. To estimate the 2D pattern for a given 3D garment model, existing approaches uses only a small number of pattern templates that are parameterized to adjust the overall shape to match the input 3D shape. Another approach inputs user-drawn seam lines over the 3D surface, then divides the 3D garment mesh into pieces and flattens them to make 2D patterns.

In this paper, we present a framework to create 2D patterns for the garments and place them in appropriate locations around the body for simulation. Regarding the creation of 2D patterns, we take the approach of cutting a 3D garment into pieces along some estimated seam lines and flatten the sub-divided pieces. To this end, we developed a stable method to estimate seam lines. Compared with previous work, our framework obtains 2D patterns that are not restricted by templates, without user intervention.

Our system requires the input scan mesh to be 2-manifold possibly with boundaries and free from topological merging with other parts. Other than this requirement, our system is robust to noisy input data (e.g., the head in Fig. 18(j)), complex color patterns in cloth (e.g., dot pattern in Fig. 18(f)), or even with additional objects (e.g., the clutch bag in Fig. 18(a)). Note that various types of garments (e.g., pants or skirts) are automatically computed to their corresponding pattern shapes without manual specification of their types from the user. We validate the effectiveness of the proposed method by constructing simulatable garment models from a number of scan data obtained from public repositories.

2. Related Work

There has been a wide variety of studies on modeling and simulating cloth and garments in computer graphics research. Among them, our discussion on previous studies focuses on methods for garment segmentation and modeling that are closely related to our research.

2.1. Garment segmentation on 3D human scans

Segmentation is one of the critical components of the garment model estimation. A number of studies, e.g., [GLL*18, LGSL19, RLH*19], showcase notable progress in cloth segmentation (also called "human parsing") in the image domain. On the other hand, the task of cloth segmentation in 3D lacks solutions of a similar quality level, with only a few studies exploring the problem. The work of [SGDA*10] determines a non-rigidly deforming surface region as a garment part. The work of [PMPHB17, ZPBPM17] performs direct vertex-level segmentation using texture maps. This method defines weak priors on the surface that are likely to belong to a certain class, and then solves the Markov Random Field (MRF) to perform the segmentation. The work of [BTTPM19] builds upon this approach by additionally incorporating image-based semantic segmentation [GLL*18] into the pipeline. The techniques mentioned above show overall robust segmentation in their applicable scopes; however, since they use independent per-vertex prediction, segmentation is oftentimes noisy, especially on the boundaries between classes. In our work, we shift focus to accurately separating the areas belonging to different classes by incorporating boundary priors alongside the color-based segmentation.

2.2. Estimation of Garment Models

Research on estimating garment models is very diverse reflecting the complexity of the task. Researchers have explored various input modalities and representations for garment models. Modeling garment shape in terms of displacement from the body surface [NH14, YFHWW18, SOC19] is simple and intuitive but cannot represent the complex shape of garments. Sizer [TBTPM20] uses neural networks to deform pre-defined garment templates. Deep Fashion3D [ZCJ*20] builds a large scale data set of various garments by deforming adaptive templates. Chen et al. [CZL*15] represented a garment as a hierarchy of 3D components (e.g. torso, sleeve, and collar) and collected a library of various garment components. A garment model is then created from the input RGBD by matching the components from the database to the input and constructing the hierarchy with the matched components.

Another way to represent a garment is with 2D pattern templates with a small number of parameters to control the garment size which can be easily modified and simulated to make 3D garment models. Some reconstruction methods [JHK15, YPA*18] use libraries of such parametrized 2D patterns to restore garment models from images. Jeong et al. [JHK15] estimates template parameters from key landmarks detected on the input image. A recent study [XYS*19] improved landmark prediction and parts segmentation by using deep neural networks. Obtained information acts as a target for deforming the 3D garment template and recovering a texture map with a predefined 2D pattern template as a reference. Yang et al. [YPA*18] focuses on images of people wearing garments and optimizes 2D pattern parameters so that the simulated garment matches the corresponding garment silhouette from the input image. The method of [VSGC20] learns data-driven wrinkle deformation as a function of 2D pattern parameters. 2D pattern templates play an important role in [WCPM18], which presents the concept of a learned shared latent shape, a latent vector that unifies 2D sketches, 3D geometries, body shape parameters, material properties, and 2D pattern template parameters of the same garment. This latent representation allows going from one explicit representation to another but needs to be trained separately for each garment template.

Our approach is close to the line of work that obtains a garment 2D pattern by flattening a 3D garment surface. The process usually involves defining the cut lines (seams, darts, etc.) on the 3D surface and flattening the resulting 3D patches. To find the cut lines, the majority of work relies on the user to define them on the 3D surface [KK00, CKI*06, WWY03, WWY05, DJW*06, JHK06, WLL*09, TB13b, TB13a, KJLH10, LZB*18]. Other approaches [BSBC12,MWJ12] have a 3D garment geometry as a simulation result of an input 2D pattern, and refer to this initial pattern to identify appropriate cuts on the target surface. In contrast, our method relies on seam line priors defined on the parametric body model that generalize across clothing types, e.g., skirts and pants. Our approach does not require user intervention or a specific reference 2D pattern, and thus has the potential for better generalization properties.

3. System Overview

The input to our method is static scan data of a clothed person with a triangulated mesh and an associated texture map. We assume that the subject's clothing is either two pieces of upper and lower garments or just one piece. Our method automatically determines the number of garment pieces. The input mesh is assumed not to have a topological merging between parts (e.g., the hand on the waist without boundary)[†]. Our method works for non-water-tight mesh with many holes, as can be seen in Fig.18(e, j).

Figure 1 describes an overview of our algorithm. First, a bare body template model is fitted to the input scan data and then predefined boundary curves and labeled regions are projected from the body model to the scan data. By using the color information of projected regions, these initially projected curves are optimized to find the boundaries between parts. Boundary optimization is performed in the order of (upper garment↔skin), (lower garment↔shoes), and (upper garment↔lower garment), from which garments are separated from the mesh. After that, the predefined seam curves are projected from the body model to the garment to segment into individual pieces for the garment pattern. Then, each segmented piece of garment is rigidly rigged to the nearest body joint and transformed to the unposed configuration. Finally, 2D parametrization is applied to flatten each segmented piece, which is subsequently aligned to its corresponding body part to be easily stitched by cloth simulators. Each step of our method is detailed in the following section.

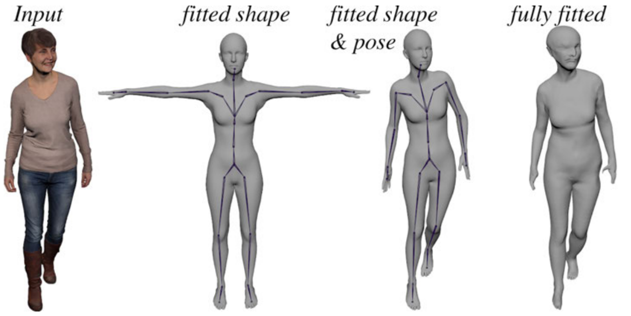

Figure 2: *From left to right, input scan data, SMPL model with fitted shape parameter, SMPL model with fitted shape and pose parameters, fully fitted mesh purposed for feature points projection (Sec. 5.1.2).*

4. Human Body Shape and Pose Estimation

The first step of our method is to estimate the naked body shape and pose under clothing from an input 3D scan. We adopt the SMPL [LMR*15] statistical body model as a representation of the bare body. The SMPL is a skinned vertex-based model $\bar{V}_{smpl}$ with a controllable shape parameter $\vec{\beta}$ ($\in \mathbb{R}^{10}$) and pose parameter $\vec{\theta}$ ($\in \mathbb{R}^{75}$). To estimate pose and shape parameters that would sufficiently match the input, following [ZPBPM17, YFHWW16], we minimize an objective function that pushes the skin vertices close to, but inside the scan mesh.

The success of an optimization-based approach relies heavily on the initial guess of parameters, and setting a default pose, e.g., A-pose, as the initial guess often leads to only local optima. An effective approach to estimating 3D human pose is to employ an image-based estimator, as shown in the recent work of [RRC*16,BKL*16,PZZD18,KBJM18,XCZ*18]. Similarly, to ensure that our system is applicable to various poses, we take an additional step of pose initialization that relies on an image-based pose estimator. Specifically, we choose OpenPose [CHS*19], which allows for 3D pose estimation from multi-view images of a human subject.

Figure 2 shows the result of the SMPL model fitting on scan data. Appendix A presents the detailed procedure of our pose and shape estimation.

5. Garment and Skin Segmentation

After obtaining a fitted SMPL body model, our method performs a clean segmentation between parts among garment pieces and skin regions from the input scan data.

5.1. Boundary Curve Optimization

Our strategy for the segmentation is to find optimal boundaries among clothing pieces and skin. We aim to find a total of three boundaries that divide between upper clothes and skin $B(U|S)$; lower clothes and shoes (or skin) $B(L|S)$; and upper and lower clothes $B(U|L)$. We achieve this by first projecting predefined boundary curves on our body model to the input scan mesh, and

[†] Input data with topological merging requires pre-processing to separate the merged parts. For example, we manually separated the hands in Fig.18(d) from the head and waist as they were merged in the original scan data.

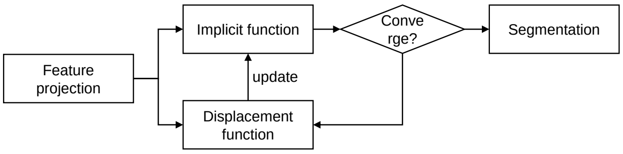

Figure 3: *Flowchart of boundary curve optimization.*

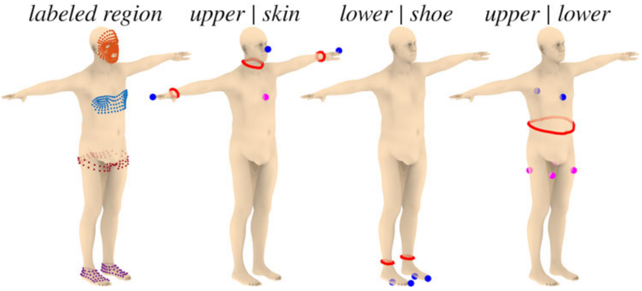

Figure 4: *From the left, pre-defined labeled region points, pre-defined spline and sign points for separating upper clothes and skin $B(U|S)$, lower clothes and skin $B(L|S)$, and upper and lower clothes $B(U|L)$. Blue and magenta points indicate positive and negative sign points, respectively.*

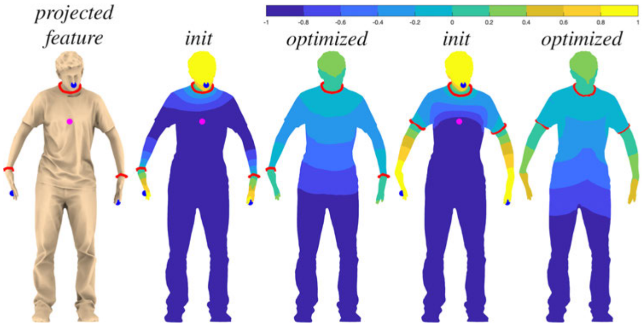

Figure 5: *Projected spline and sign points for $B(U|S)$ on scan data. The second and third columns show biharmonic function and signed geodesic function on the initial configuration of boundary curves. The fourth and fifth columns show the respective functions with optimized configuration of boundary curves. Note that the curve initially projected on the wrist are placed on the boundary of sleeves after optimization.*

then defining an implicit function on the mesh to represent the boundaries. We then adjust the implicit function by adding displacement function that is computed using color information on the input mesh iteratively until convergence (Fig. 3).

5.1.1. Predefined Feature Points on the SMPL Model

For boundary optimization, it is important to have a good initial boundary curve and displacement function. Since we fit the shape and pose of the SMPL body model, we pre-define the information necessary to define the boundary curve and displacement function on the SMPL model. To this end, a set of feature points is defined on the SMPL model (Fig. 4). The feature points include spline points, sign points, and labeled region points. Spline points define the initial boundary curves. Sign points are given either positive or negative value, which are later used to convert an explicit boundary curve to an implicit function that has positive and negative values across the boundary. Labeled region points are assigned with one of four labels, $l(u) = \{l_{skin}, l_{upper}, l_{lower}, l_{foot}\}$, and are used to collect color information for those regions. These points are defined only once for the average SMPL model and are used for every input scan mesh as the shape and pose parameters of the SMPL model adjust the locations of the points.

The spline and sign point sets are defined for each of the three boundaries to be optimized. For the boundary between the upper clothes and skin $B(U|S)$, we define 3 closed splines around the neck and each wrist. Three sign points are defined on the nose and the tip of hands with a positive value, and two additional sign points with a negative value are defined on the front and back sides of the chest. For the boundary between the lower clothes and skin $B(L|S)$, we define two closed splines around the ankle. Sign points are defined with four points with a positive value on each tip of the toe and bottom of the foot. For the boundary between the upper and lower clothes $B(U|L)$, we define a spline around the waist. For feature points, we define two points with a positive value on the front and back of the chest.

5.1.2. Feature Point Projection

To stably project pre-defined feature points on the SMPL model to scan data, we take an additional step to adjust the vertices of the fitted SMPL model to fully match the input scan data by using the nonrigid ICP method, for which the open source code of [Man20] is used. As this fully fitted mesh could involve excessive distortion to match garment parts, it only serves to stably deliver the feature points to the scan data and is not used as body mesh. Figure 2 (right) shows the result of a fully fitted SMPL model. Projecting the feature points to the scan data can now be simply done by finding the closest points on the scan surface from the fully fitted SMPL body model. For spline projection, we found projecting spline points directly could lead to noisy and jagged curves. We instead project the anchor points of the spline, and compute the spline on the scan surface with a weighted average on surface technique [PBDSH13] on the scan surface. In Fig. 7 (left), we show the projected spline and sign points on the input mesh.

This projection scheme is mostly stable, but there can be ill-projected points, which can lead to an undesirable initial spline and implicit function. Therefore, we find these ill-projected points with some simple heuristics and discard them. Removing a portion of the feature points is tolerated by our method. Appendix B details how we determine ill-projected points.

5.1.3. Implicit Function Computation

Performing curve optimization on a discrete surface domain is cumbersome, especially when a curve is constrained on a surface. Inspired by the idea of the variational surface cutting method [SC18b], instead of explicitly representing a curve on a surface,

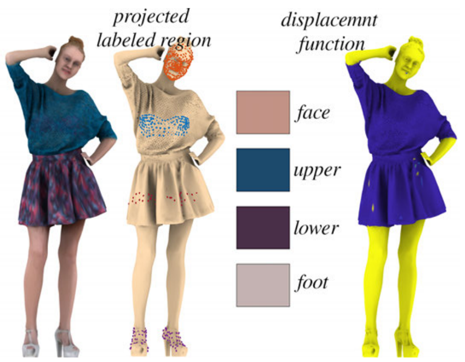

Figure 6: *From left to right, smoothed color on scan data, projected labeled region points, average color for each region, and displacement function.*

which is highly error-prone, we transform it to a smooth implicit function on a surface, in which a certain value of isoline represents a curve. A smooth implicit function on the surface makes curve optimization much easier.

Biharmonic Function With the projected spline and sign points, we solve a biharmonic function on the scan mesh with a constraint of zero value on the spline points and pre-assigned positive/negative values on the sign points. This can be expressed as

$$\begin{aligned} \phi_b &= \underset{u}{\operatorname{argmin}} \int_\Omega ||\Delta u||^2 dA \\ &s.t.\ u|_{\mathcal{C}} = 0\ ,\ u|_p = c, \end{aligned} \tag{1}$$

where $\mathcal{C}$ denotes a spline point on the surface, p is a sign point, and c is its pre-assigned value. Constraints on a curve can be expressed in terms of the barycentric coordinates. The resulting biharmonic function is a smooth function with zero values on the boundary spline points with different signs across the boundary. The zero value of isoline on this scalar function ϕ_b implicitly represents the spline curve.

However, using the biharmonic function directly as our implicit function for curve optimization can be problematic. The gradient of this biharmonic function can be different depending on the relative configuration between the spline and sign points. As we add a displacement function to our implicit function, this could lead to a biased configuration of the curve, e.g., an area with a steeper gradient can have less movement then an area with a flatter gradient, even with the same displacement function value. Therefore, we use the signed geodesic distance function ϕ_i for the boundary optimization.

Signed Geodesic Distance Function By using the biharmonic function and its extracted curve with a zero value of isoline, we compute geodesic distance from isoline using the fast marching method [KS98]. As this geodesic distance does not distinguish the sign between regions across the boundary, we determine the sign from the biharmonic function. By definition, the magnitude of the gradient of the signed geodesic distance ϕ_i is constant, $||\nabla\phi_i|| = 1$. Because the signed geodesic distance has a constant gradient from the isoline, it is much more stable than the biharmonic function when numerical optimization is performed. Figure 7 compares the biharmonic function and signed geodesic function. Note that the signed geodesic function has a consistent gradient compared to the biharmonic function.

Whenever there is an update on the implicit function, we recompute the biharmonic function and signed geodesic distance function.

5.1.4. Displacement Function Computation

Ultimately, we want our curve to be placed on the boundary between each region. To drive the curve towards the true boundary, we update the implicit function by adding a displacement function defined on the mesh surface. The displacement function needs to have zero value on and different signs across the true boundary.

We use the posterior probability function of the Gaussian Mixture Model (GMM) for our displacement function. We use CIEL*a*b* color space as it is good at distinguishing perceptually different colors. Before converting to CIEL*a*b* space, we perform a slight color smoothing in RGB space so that it can have a similar value around its local neighbour.

With projected labeled region points, we fit GMM on the color of these points. As we focus on the boundary between two regions for each step, the GMM is fit with two adjacent labeled region points. For example, to compute the boundary between the skin and upper boundary $B(U|S)$, we fit the GMM with two labeled regions: l_{skin} and l_{upper}. Then, the fitted GMM allows for calculating the posterior probability of any point on the surface u being classified between two labeled regions. We re-scale these probability values $[0,1]$ to the range $[-1,1]$ and use this as the displacement function. Given CIEL*a*b* color $c(u)$ on the vertex, the displacement function is defined as:

$$\phi_d(u) = 2(Pr(l(u) = l_{skin}|c(u)) - 0.5) \tag{2}$$

For the boundary $B(U|L)$, this will be $\phi_d(u) = 2(p(l(u) = l_{foot}|c(u)) - 0.5)$. For $B(L|S)$, the lower clothes can border either the shoe or leg skin. So we fit the GMM with three labeled regions of l_{foot}, l_{skin} and l_{lower}. We use the posterior probability of being either l_{foot} and l_{skin} as the displacement function for $B(L|S)$, i.e., $\phi_d(u) = 2(Pr(l(u) = l_{foot}$ or $l_{skin}|c(u)) - 0.5)$. Figure 8 shows an example of the displacement function computed for the boundary $B(U|L)$.

The displacement function is added with a small step size ($\alpha = 0.01$) as follows:

$$\tilde{\phi}_{i+1} = \phi_i + \alpha\phi_d \tag{3}$$

Length Regularization One may assume that this probability function could be applied directly to segment each region. However, this probability function has a high potential of misclassification. This was the approach of ClothCap [PMPHB17], which uses a sequence of data to fix this issue. Our boundary optimization approach is robust with a noisy probability function. The robustness is further improved by a length regularization step. For this, we apply

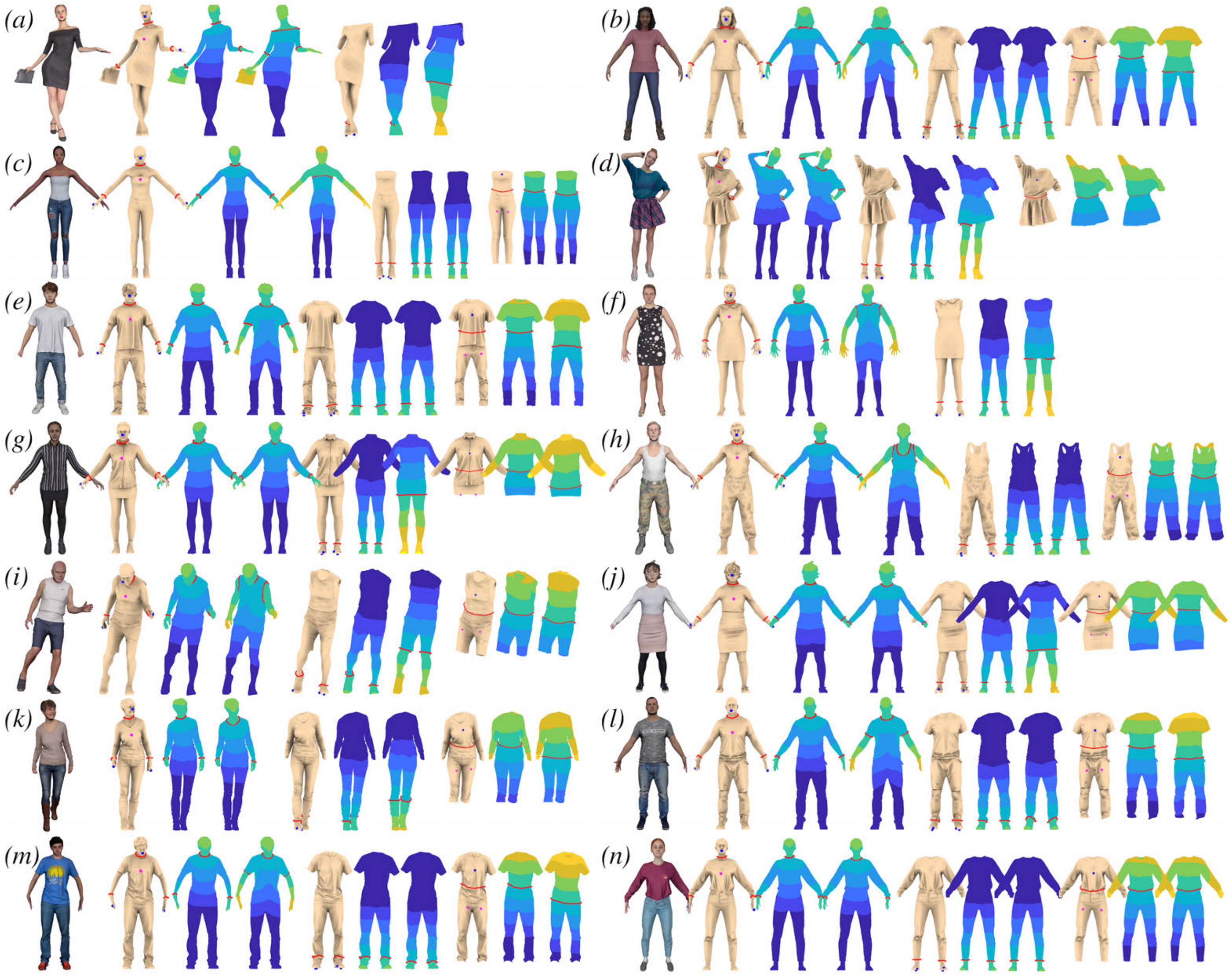

Figure 7: *Boundary optimization is performed in the order of $B(U|S)$, $B(L|S)$, and $B(U|L)$. For each optimization, this figure shows its initial boundary and implicit function followed by the optimized boundary and implicit function, for 14 subjects.*

the backward Euler scheme to update the implicit function similar to [SC18b].

$$(\mathrm{I}+\beta\Delta)\bar{\phi}_{i+1} = \tilde{\phi}_{i+1}, \tag{4}$$

where Δ denotes the Laplacian operator and β is the length regularization intensity parameter ($\beta = 0.05$). This will remove noisy local changes on the curve.

With the updated implicit function $\bar{\phi}_{i+1}$, we re-compute the signed geodesic distance function ϕ_{i+1}. The implicit function is iteratively updated until the difference from the previous implicit function becomes negligible.

5.1.5. Overall Algorithm for Boundary Optimization

Algorithm 1 summarizes the overall procedure for finding the optimal boundary.

Algorithm 1 Boundary Optimization Algorithm.

```
Project features points from SMPL model
Compute initial implicit function $\phi_0$ and displacement function $\phi_d$
repeat
    Add displacement function $\tilde{\phi}_{i+1}$
    Regularize implicit function $\bar{\phi}_{i+1}$
    Extract isoline & fit spline $\mathcal{C}_{i+1}$
    Compute implicit function $\phi_{i+1}$
until $\|\phi_{i+1} - \phi_i\|^2 <$ threshold
```

This same algorithm is applied for each step of finding boundaries $B(U|S), B(L|S)$ and $B(U|L)$ in sequence. We first find the optimal boundary curve $B(U|S)$, cut the mesh along the boundary curve and discard the skin part. With the remaining mesh, we find the op-

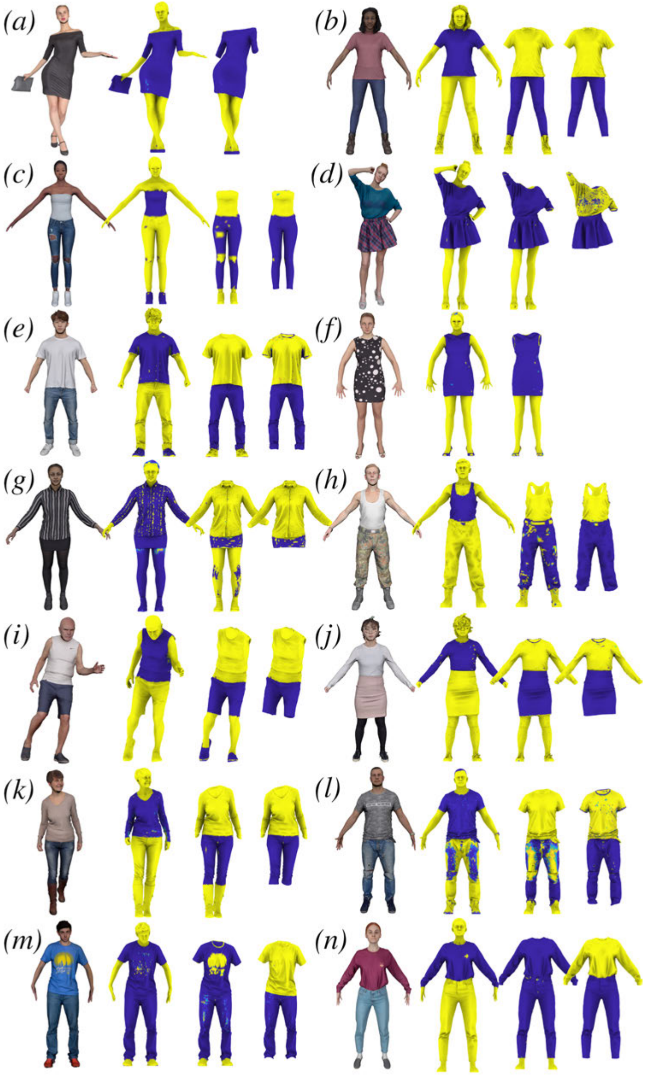

Figure 8: *Displacement functions for $B(U|S)$, $B(L|S)$, and $B(U|L)$.*

timal boundary curve $B(L|S)$ and discard the foot part. Lastly, we find the optimal boundary curve $B(U|L)$ then cut the cloth with the upper and lower part. For a one-piece garment, we skip the last part. Figure 7 shows the results of the implicit function applied on each stage applied, and Fig. 8 shows the results of the displacement function on each stage on 14 subjects. Note that different garment topologies (e.g., skirts and pants) can be generated by our method with the same initial boundary template. This is possible because our implicit function optimization enables topology change, which contrasts greatly with other approaches based on pattern or garment templates.

5.2. Handling Gap-Filling Triangles

In some cases, the scan mesh contains spurious triangles to fill the gap between the body and clothes (e.g., between the bottom of the skirt and leg). These triangles are typically assigned with meaningless colors, and they can prevent our method from finding the correct boundary. To fix this problem, we use the fact that these gap-filling triangles are oriented more or less orthogonal to the limb. Thus, when the input data contains such spurious triangles, we employ an additional displacement function that is determined by the normal direction of the vertices.

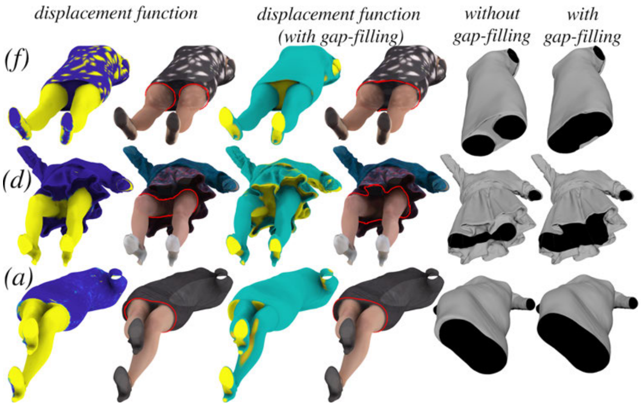

Figure 9: *From the left, displacement function by Eq. (2) and its resulting boundary, displacement function by Eq. (6) and its resulting boundary, and the comparison of segmented garment meshes.*

After boundary curve optimization performed with the displacement function in Eq. (2), we first find the best fitting plane on the boundary curve points. The normal vectors of the fitting plane are then compared with the vertices in the mesh. The displacement function has a larger value for a vertex of which normal $\vec{n}_u$ is closer to the plane normal $\vec{o}$, encouraging the boundary curve to pass the gap-filling triangles. Specifically, we define the displacement function as follows:

$$\phi_{dn}(u) = \max(2(|\vec{o} \cdot \vec{n}_u| - 0.5), 0). \tag{5}$$

Note that $\phi_{dn}(u) = 0$ for the vertices whose normal is far from $\vec{o}$, adding no effect to the displacement function. Boundary optimization is then performed once more with the normal displacement function added as below:

$$\tilde{\phi}_{i+1} = \phi_i + \alpha(\phi_d + \phi_{dn}). \tag{6}$$

Figure 9 compares the boundaries obtained before and after applying Eq. (6) on the skirt data on subjects a, d, and f.

Note that if a reasonable boundary is found before the gap-filling algorithm, the additional displacement function will not affect the original boundary position by converging to the solution quickly, as in the example of subject a.

5.3. Upper-Lower Clothes Segmentation

This step is taken to divide the upper and lower garments. If the difference between the mean color of the upper and lower body is small, we assume it is one-piece, skip this step and directly proceed to seam computation (e.g., subjects a and f). Otherwise, the following step is taken.

As the upper clothes often occlude the lower clothes or vice

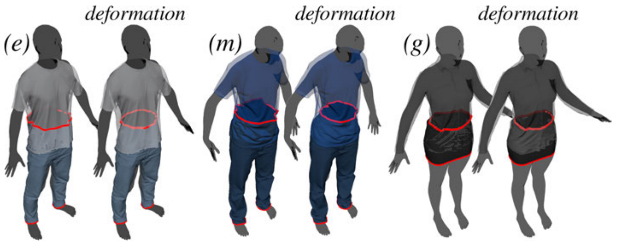

Figure 10: *Boundary constrained deformation applied to the lower garment. Initially, the top part of the pants coincides with the upper garment (left). After deformation, the same part has shrunk towards the body, naturally occluded by the upper garment (right).*

versa, finding only one visible boundary between the two and dividing them along the boundary results in an incomplete garment model for the occluded clothes. Therefore, it is necessary to infer the boundary of the hidden part of the clothes.

To this end, we first identify the occluded clothes. Our predefined spline for the upper-lower clothes boundary is located on the waist around the navel. If the optimized spline is located closely to the initial spline, we determine that the two clothes do not overlap. Otherwise, if it is positioned below the initial spline, we determine that the lower clothes are occluded, and vice versa. For non-occluded clothes, we use the optimized boundary for its segmentation. For occluded cloths, the initial boundary curve is used for the segmentation.

As the initial boundary curve encloses the outer clothes, we take an extra step to estimate the geometry of the hidden part of the occluded clothes. For this, we solve the biharmonic deformation [JTSZ10] with two constraints: that the segmented boundary shrinks to the projected positions on the SMPL body and that the positions corresponding to the optimized boundary keep their positions. This results in an unchanged visible region while the occluded region fits the body more tightly. Figure 10 shows the lower garment mesh before and after the boundary constrained deformation is applied.

6. Pattern Generation

The pattern serves as a low-dimensional space for creating various styles of a fabricatable garment. Because variation on the pattern space guarantees the construction of a reasonable clothes shape, estimating pattern shapes from 3D scan data is essentially finding a meaningful representation for the 3D garment model.

Until now, we have segmented each garment piece from scan data. To estimate a pattern for each garment piece, we first determine seam lines on the garment mesh, separate the mesh into several sub-pieces, and then flatten the sub-pieces to obtain a pattern. Subsequently, we remesh the pattern for quality triangulation. Finally, we place each panel of the pattern in 3D space near the corresponding body part to make it ready for simulation.

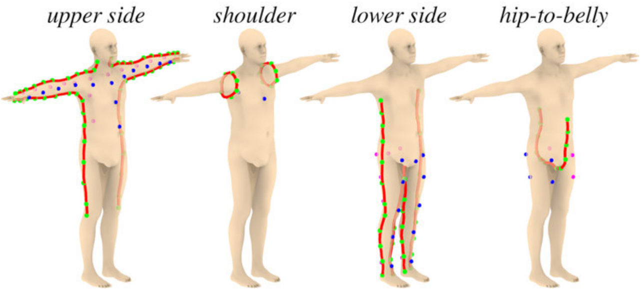

Figure 11: *Predefined spline and sign points for upper side seam, shoulder seam, lower side seam, and hip-to-belly seam*

6.1. Seam Line Estimation

Our goal is to estimate seam lines to divide a garment mesh into several pieces, each of which will make a panel of a pattern. Our method for estimating seam line shares many components with the boundary optimization method in Sec. 5.1. We pre-define splines corresponding to seam lines as well as some feature points on the body model. We project the spline to the scan mesh and convert it to an implicit function. Length regularization is performed to smooth the curve and then extract isoline to identify the seam line on the scan mesh.

6.1.1. Predefined Data on SMPL Model

We assume that our garment model is composed of several simple components commonly used in the garment industry. Figure 11 shows pre-defined spline and sign points for seam line computation. The upper clothes are assumed to consist of front, back and sleeve panels. To divide into each part, splines are pre-defined on the side of the body for front and back segmentation and on the shoulder for sleeve segmentation. For the front-back segmentation, feature points are pre-defined with 13 points on the front of the upper body with a positive value and 13 points on the back of the lower body with a negative value. For the shoulder seam segmentation, 2 points are defined with a positive value on the front and back of the chest. Green dots on the spline denote the anchor points.

We assume that the lower clothes are composed of four panels that divide the clothes from the front-back and left-right. To segment to each part, splines on the side of the body are pre-defined for front and back segmentation and from hip to belly to segment the left and right segmentation. For the side segmentation, sign points are pre-defined with 11 points on the front of the lower body with a positive value and paired with 11 points on the back. For the left-right segmentation, 4 points on left side of body with a positive value and 4 points on right side of body with a negative value are defined.

We do not have labeled region points for the seam line segmentation because we do not perform boundary curve optimization.

6.1.2. Seam Line Computation

To compute the seam line, we employ the spline projection technique introduced in Sec. 5.1.2 and convert the spline into an im-

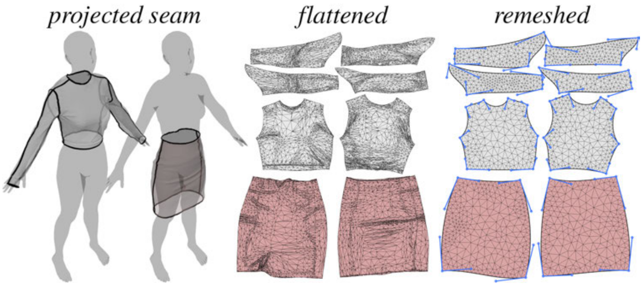

Figure 12: *Projected seam curve on scan data (left), flattened pieces from scan data (middle), and re-meshed pattern with Bezier curve fitting (right). Note that coarsely re-meshed result is shown for clear visualization. In practice, re-meshing is conducted with much denser triangulation.*

plicit function representation as in Sec. 5.1.3 along with length regularization. We then extract zero valued isoline as our seam line.

Seam Line Removal We employ the same heuristics in Appendix B to remove ill-projected anchor points. Moreover, unlike boundary spline curves, we allow the seam curves to be entirely removed if only a few (less than three in our experiment) anchor points remain. This enables the automatic removing of the hip-to-belly seam for skirts (d, g, j) and the shoulder seam for sleeveless shirts (c, f, h, i) in Figs. 8 and 18.

Paired Feature Point Projection A notable difference between seam line computation and garment segmentation is that here, we enforce that the projected feature points for positive and negative values exist in pairs: if one of paired feature point is discarded due to ill projection, we also discard its paired feature point. This is because, if feature points are not equally assigned, it can lead to the biased computation of the biharmonic function, which potentially leads to a undesirable configuration of the seam line.

Shoulder Seam Line Computation We expect the shoulder seam line to be placed around the armpit, which is sometimes not achieved in the above methods. To fix this problem, we take extra care on the shoulder seam line computation. When we compute an implicit function with the biharmonic function, we constrain zero values on projected anchor points instead of spline points. Then, we add extra constraint with the negative value at the boundary of the arm and the positive value at the boundary of the neck and waist. This helps our boundary curve to be placed around the armpit.

6.2. Flattening and Remeshing the Pattern

To minimize distortion from the 2D pattern to 3D garment part, finding 2D parameterization with zero area distortion for each 3D garment part gives us a reasonable pattern shape. Specifically, we use [SC18a] with no boundary constraint specified.

We perform remeshing for better triangulation, which is essential for quality simulation. When a garment piece is segmented along the seam line, as described in Sec. 6.1, we save the intersecting point between the seam line and boundary curve of the garment. Then each pattern shape can be represented with these points P and edges E. From the pattern shape, we fit each edge of E with a Bezier curve. Then, we resample the points along the edge, and perform Delaunay triangulation using the TRIANGLE of [She96]. As a result, a pattern is obtained with a clean and smooth boundary and evenly distributed triangles inside. Figure 12 shows the result of the procedure to create flattened pieces from the projected seam curve, and subsequently to obtain remeshed pattern pieces. Note the jagged boundaries and ill-shaped triangles are resolved after remeshing.

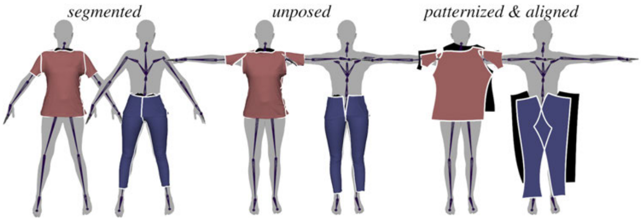

Figure 13: *Extracted and segmented upper and lower garments from scan data are rigidly attached to the estimated SMPL model (left). Unposing the SMPL model transforms each segmented piece to its rest configuration (middle). Flattened, remeshed pattern is aligned in correspondence with 3D configuration of body (right).*

6.3. Alignment of the Pattern

For the draping simulation, it is better for the body model to take a standard pose with arms and legs stretched and spread nicely to prevent tangling in the cloth simulation. In our case, this is simply achieved by unposing the SMPL body model, i.e., setting the pose parameter to zero. In addition, pattern panels must be placed near corresponding body parts, so that they are ready to be stitched with other panels and draped with the simulation. Figure 13 shows a result of an unposed and aligned pattern on a body model.

To determine the placement of the pattern, we use the spatial relation between 3D garment parts and their corresponding body parts. Specifically, each segmented garment part is rigidly bound to the skeleton of the SMPL model: the trunk parts and the sleeves are bound to the chest and shoulder joints, respectively, and the lower garment parts are bound to the hip joint. We then unpose the SMPL body model to its rest pose along with the bound garment parts. This results in a rest pose configuration of garment parts.

Since flattened garment pieces are obtained with parametrization, we have a correspondence between the 3D garment part and 2D pattern shape. We can also easily find the correspondence between patterns before and after re-meshing. With these consecutive correspondence relations, re-meshed pattern panels are aligned to the 3D rest pose configuration of its corresponding garment part. Specifically, we rotate the pattern panels about the frontal axis to align with their corresponding garment parts and then apply a short distance along the frontal axis between front and back patterns so that they do not collide with the body. The garment is finally ready to be simulated.

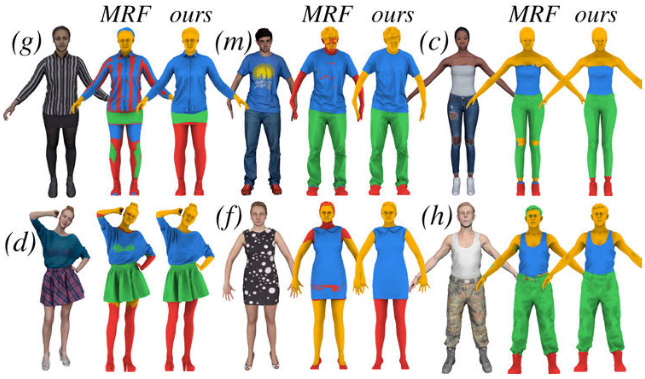

Figure 14: *Segmentation results with an MRF-based method (middle) and ours (right).*

7. Results

Our system is implemented in C++ and MATLAB. We used libIGL [JP*18] and GPTOOLBOX [J*18] as geometry processing libraries. The major bottleneck in the execution of our system is the boundary optimization step, of which computational performance is reported in Table 7. Note that our code is no optimized for performance. All the experiments were performed on a MacBook Pro with a 2.4 GHz Quad-Core Intel Core i5 CPU and 16GB of memory.

As our final output is pattern aligned on the body with a rest pose, we can import the result directly into commercial cloth simulation software. CLO software was used to run cloth simulation with our estimated pattern shape. Seam stitching information was manually set between garment panels. Physical parameters, such as stretch, bending, density, and friction coefficients, were also set manually to replicate the shape of the input scan data.

For testing, we collected scan data of 15 subjects from publicly available repositories such as Sketchfab and Turbosquid (subjects $a,b,c,d,f,g,h,i,k,l,m,n,r,u$, and v), and obtained data by scanning 2 subjects with a multi-camera 3D scanner (subjects e and j). In addition, we tested our method on an existing dataset from MGN (subjects o,p,q,s, and t). Figure 18 shows the results of our method from input scan data to final the draped simulation, as well as each intermediate step of the fitted body model, segmented garment, estimated 2D pattern, aligned and sewed pattern on the rest-shaped model. Note that various garment styles were successfully captured and reconstructed.

Comparison with MRF-based Segmentation We compare our segmentation with the MRF-based method in Fig. 14. Our implementation of MRF-based segmentation followed the procedure in ClothCap [PMPHB17]. Specifically, we used our displacement function from Eq. 2 as the data likelihood term, and our projected label region as the garment prior term. As can be seen in Fig. 14, the MRF-based method may exhibit mislabeled vertices inside the garment. In contrast, since our method finds the boundary of segmentation, it is inherently free from the mislabeled internal vertices. In addition, the implicit representation of the curve allows our method to generate a smooth boundary independently of mesh resolution, which cannot be done with a per-vertex labeling approach such MRF-based segmentation, as demonstrated by the jagged boundaries of subjects c and h. Figure 15 compares the segmentation results with ClothCap and ours. The results of ClothCap exhibit jagged boundaries because of the per-vertex labelling, as highlighted with red circles. In contrast, our results show a smooth boundary, even with a low resolution of mesh. Note that the ClothCap method uses a sequence of scan data to increase the robustness of labelling, while ours uses only single scan data. As shown in Fig. 16, the work of [BTTPM19] greatly improved the quality of the MRF-based segmentation by combining with semantic image segmentation. However, it still shows a few incomplete segmentations in the published dataset. For example, subjects p and q in Fig. 16 (purple circles) show the segmented boundary between the knee and pants placed slightly lower than the actual boundary. Geometric deformation of a garment may induce artifacts, as shown in Fig. 16 (red circles), whereas physical simulation of a 2D garment pattern is less prone to errors.

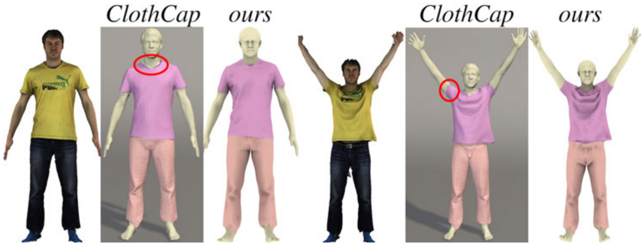

Figure 15: *Segmentation results comparison with data from ClothCap [PMPHB17] and ours.*

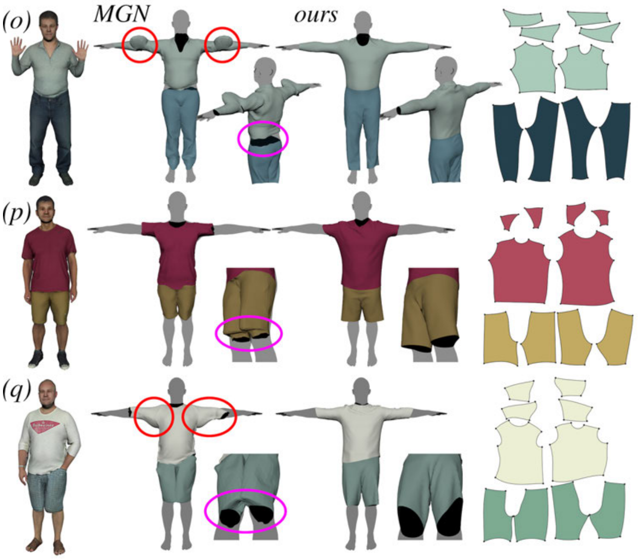

Figure 16: *From the left, input scan data from MGN dataset, geometric garment deformation with MRF-based segmentation and image-parsing prior [BTTPM19], and our method with draping pose with simulation. Purple circles: inexact segmentation. Red circles: artifact of geometric deformation of garment.*

subject	# verts	time (seconds)		
		boundary		
		B(U|S)	B(L|S)	B(U|L)
a	48000	71.4	84.5	N/A
b	50000	93.3	36.9	32.7
c	7277	11.1	1.6	1.5
d	22531	34.8	53.4	13.8
e	25017	43.5	13.6	19.4
f	13931	29.6	20.8	N/A
g	49956	40.8	109.0	39.1
h	31218	80.2	29.5	6.0
i	17363	36.9	26.9	11.5
j	24974	20.9	37.2	9.3
k	14999	24.2	20.1	13.1
l	37499	76.6	30.6	47.2
m	49483	92.7	42.7	49.3
n	31612	26.4	8.7	9.1

Table 1: *Compute time for boundary optimization of each subject.*

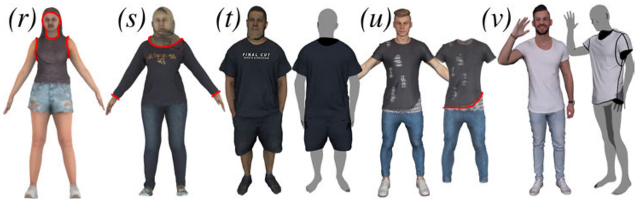

Figure 17: *Failure cases of our method.*

Comparison with Displacement-based Geometric Deformation Figure 16 compares our method with a geometric garment deformation scheme that represents the garment shape as a displacement from the body mesh and applies linear blend skinning to the garment vertices with the skinning weights of the associated body vertices as detailed in Sec. 3.1 in [BTTPM19]. Subjects *o* and *q* in Fig. 16 show unnatural garment deformation around the shoulder and elbow due to this geometric deformation scheme. By contrast, a physical simulation approach such as ours typically leads to more natural deformation.

8. Limitations and Future Work

Our method has several limitations that must be overcome with future research. First, our method can be applied only to a limited scope of scan data. It assumes that the subject is wearing rather simple two-piece or one-piece garments and cannot handle a subject wearing an additional garment, such as a coat. Subject *u* in Fig. 17 shows this issue where the subject is wearing a layered upper garment, and our algorithm distinguishes the inner layer to be a part of pants, placing a boundary between the outer layer and inner layer of the upper garment. Another limitation is that our method does not reconstruct garment parts occluded by hair or other objects. Subject *r* in Fig. 17 shows a failure case, where the neck-garment boundary is misplaced after optimization due to occlusion by hair. In the case of subject *s* from the MGN dataset, our method finds a boundary between hijab and shirts. To deal with such variations in input data, semantic recognition of body parts and garments will be necessary.

Our boundary optimization method uses a displacement function that only uses color information. Thus, it cannot distinguish between upper and lower garments with very similar color, such as subject *t* from the MGN dataset. Including geometric information in the optimization, such as curvature, is a promising way to improve the quality of the boundary. Another interesting future research topic is to improve this displacement, possibly by combining with deep learning approaches, such as semantic parsing.

Lastly, our seam computation step is rather simple compared with the boundary optimization step, and it may project the seam curve in a sub-optimal place. Subject *u* in Fig. 17 shows that the shoulder seam is projected too far toward the chest. To improve the quality of the seam computation, we have tested it with distortion minimization as in [SC18b], but it was not very effective in our problem domain. Still, it remains as interesting future work to develop an effective optimization framework for seam computation.

Acknowledgement

This work was supported by National Research Foundation, Korea (NRF-2020R1A2C2011541) and Korea Creative Content Agency, Korea (R2020040211).

References

[AMO] Agarwal S., Mierle K., Others: Ceres solver. `http://ceres-solver.org`. 14

[BA05] Bærentzen J. A., Aanæs H.: Signed distance computation using the angle weighted pseudonormal. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (5 2005), 243–253. 14

[BKL*16] Bogo F., Kanazawa A., Lassner C., Gehler P., Romero J., Black M. J.: Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision* (2016), Springer, pp. 561–578. 3

[BSBC12] Brouet R., Sheffer A., Boissieux L., Cani M.-P.: Design preserving garment transfer. *ACM Transactions on Graphics (TOG) 31*, 4 (7 2012), 1–11. 3

[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 International Conference on Computer Vision* (2019), pp. 5420–5430. 2, 10, 11

[CHS*19] Cao Z., Hidalgo G., Simon T., Wei S.-E., Sheikh Y.: Openpose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence 43*, 1 (2019), 172–186. 3, 13

[CKI*06] Cho Y., Komatsu T., Inui S., Takatera M., Shimizu Y., Park H.: Individual Pattern Making Using Computerized Draping Method for Clothing. *Textile Research Journal 76*, 8 (8 2006), 646–654. 3

[CZL*15] Chen X., Zhou B., Lu F., Wang L., Bi L., Tan P.: Garment modeling with a depth camera. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 1–12. 2

[DJW*06] Decaudin P., Julius D., Wither J., Boissieux L., Sheffer A., Cani M. P.: Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum 25*, 3 (2006), 625–634. 3

[GLL*18] Gong K., Liang X., Li Y., Chen Y., Yang M., Lin L.: Instance-level Human Parsing via Part Grouping Network. In *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 770–785. 2

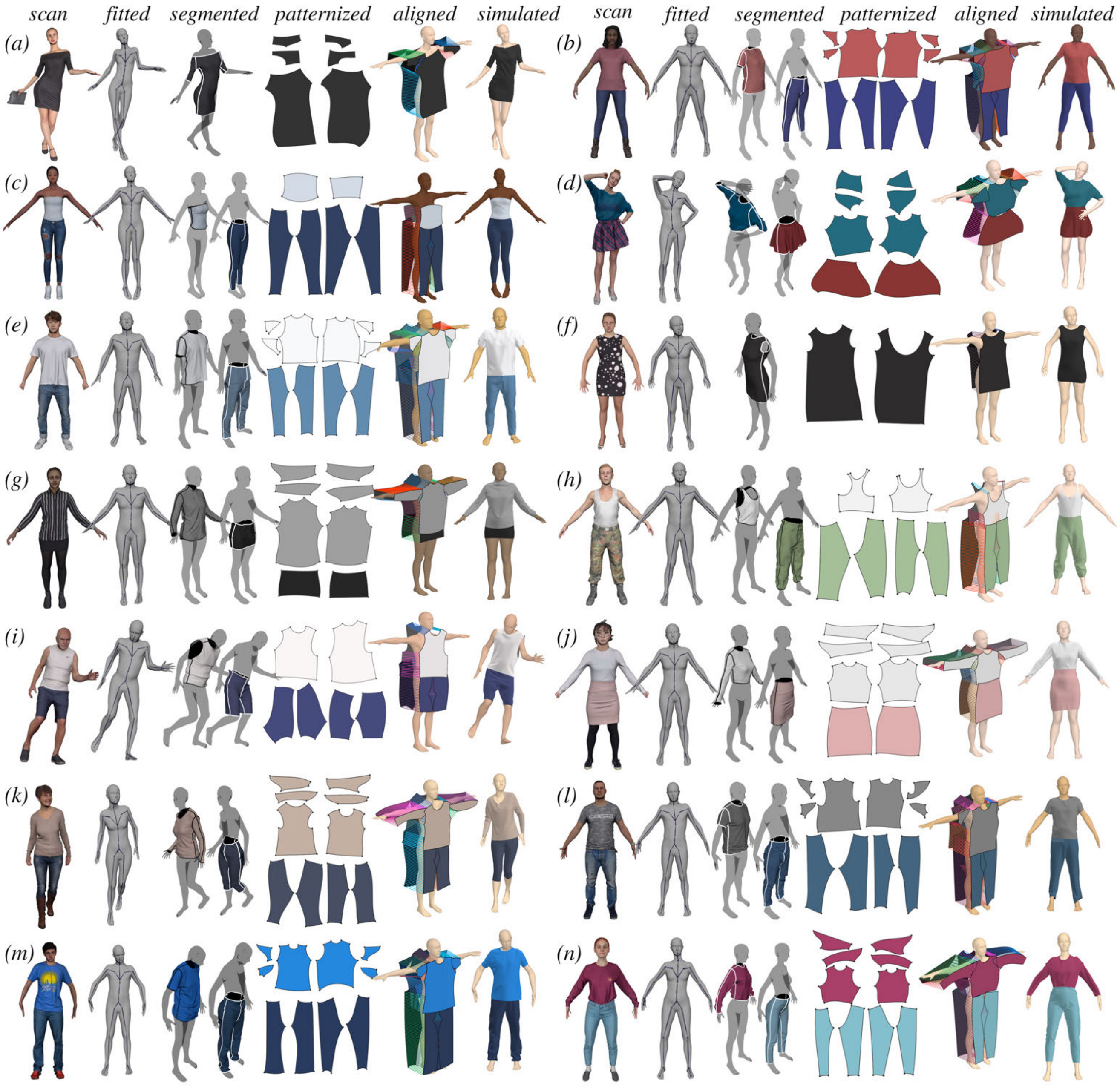

Figure 18: *Input scan data and simulated garment results.*

[J*18] Jacobson A., et al.: gptoolbox: Geometry processing toolbox. *ONLINE: http://github. com/alecjacobson/gptoolbox* (2018). 10

[JHK06] Jeong Y., Hong K., Kim S. J.: 3D pattern construction and its application to tight-fitting garments for comfortable pressure sensation. *Fibers and Polymers 7*, 2 (6 2006), 195–202. 3

[JHK15] Jeong M.-H., Han D.-H., Ko H.-S.: Garment capture from a photograph. *Computer Animation and Virtual Worlds*, 26 (2015), 291–300. 2

[JP*18] Jacobson A., Panozzo D., et al.: libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 10, 14

[JTSZ10] Jacobson A., Tosun E., Sorkine O., Zorin D.: Mixed finite elements for variational surface modeling. In *Computer graphics forum* (2010), vol. 29, Wiley Online Library, pp. 1565–1574. 8

[KBJM18] Kanazawa A., Black M. J., Jacobs D. W., Malik J.: End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT, USA, 2018), pp. 7122–7131. 3

[KJLH10] Kim S., Jeong Y., Lee Y., Hong K.: 3D pattern development of tight-fitting dress for an asymmetrical female manikin. *Fibers and Polymers 11*, 1 (3 2010), 142–146. 3

[KK00] Kang T. J., Kim S. M.: Optimized garment pattern generation based on three-dimensional anthropometric measurement. *International Journal of Clothing Science and Technology 12*, 4 (2000), 240–254. 3

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences 95*, 15 (1998), 8431–8435. 5

[LGSL19] LIANG X., GONG K., SHEN X., LIN L.: Look into Person: Joint Body Parsing & Pose Estimation Network and a New Benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*, 4 (4 2019), 871–885. 2

[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), 248. 3, 14

[LZB*18] LIU K., ZENG X., BRUNIAUX P., TAO X., YAO X., LI V., WANG J.: 3D interactive garment pattern-making technology. *Computer Aided Design 104* (11 2018), 113–124. 3

[Man20] MANU:. https://www.mathworks.com/matlabcentral/fileexchange/41396-nonrigidicp, 2020. 4

[MWJ12] MENG Y., WANG C. C., JIN X.: Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design 44*, 1 (1 2012), 68–76. 3

[NH14] NEOPHYTOU A., HILTON A.: A layered model of human body and garment deformation. In *2014 2nd International Conference on 3D Vision* (Tokyo, Japan, 2014), vol. 1, IEEE, pp. 171–178. 2

[PBDSH13] PANOZZO D., BARAN I., DIAMANTI O., SORKINE-HORNUNG O.: Weighted averages on surfaces. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 60. 4

[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), 73. 2, 5, 10

[PZZD18] PAVLAKOS G., ZHU L., ZHOU X., DANIILIDIS K.: Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, Utah, USA, 2018), pp. 459–468. 3

[RLH*19] RUAN T., LIU T., HUANG Z., WEI Y., WEI S., ZHAO Y.: Devil in the Details: Towards Accurate Single and Multiple Human Parsing. *Proceedings of the AAAI Conference on Artificial Intelligence 33* (7 2019), 4814–4821. 2

[RRC*16] RHODIN H., ROBERTINI N., CASAS D., RICHARDT C., SEIDEL H.-P., THEOBALT C.: General automatic human shape and motion capture using volumetric contour cues. In *European Conference on Computer Vision* (Amsterdam, The Netherlands, 2016), Springer, pp. 509–526. 3

[SC18a] SAWHNEY R., CRANE K.: Boundary first flattening. *ACM Transactions on Graphics (TOG) 37*, 1 (2018), 5. 9

[SC18b] SHARP N., CRANE K.: Variational surface cutting. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 156. 2, 4, 6, 11

[SGDA*10] STOLL C., GALL J., DE AGUIAR E., THRUN S., THEOBALT C.: Video-based reconstruction of animatable human characters. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 1–10. 2

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *Workshop on Applied Computational Geometry* (1996), Springer, pp. 203–222. 9

[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. 2

[TB13a] TAO X., BRUNIAUX P.: Toward advanced three-dimensional modeling of garment prototype from draping technique. *International Journal of Clothing Science and Technology 25*, 4 (7 2013), 266–283. 3

[TB13b] THOMASSEY S., BRUNIAUX P.: A template of ease allowance for garments based on a 3D reverse methodology. *International Journal of Industrial Ergonomics 43*, 5 (2013), 406–416. 3

[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 *European Conference on Computer Vision (ECCV)* (August 2020), Springer. 2

[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

[WCPM18] WANG T. Y., CEYLAN D., POPOVIĆ J., MITRA N. J.: Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics (TOG) 37*, 6 (12 2018), 1–13. 3

[WLL*09] WANG J., LU G., LI W., CHEN L., SAKAGUTI Y.: Interactive 3D garment design with constrained contour curves and style curves. *Computer Aided Design 41*, 9 (9 2009), 614–625. 3

[WWY03] WANG C. C., WANG Y., YUEN M. M.: Feature based 3D garment design through 2D sketches. *Computer Aided Design 35*, 7 (6 2003), 659–672. 3

[WWY05] WANG C. C., WANG Y., YUEN M. M.: Design automation for customized apparel products. *Computer Aided Design 37*, 7 (6 2005), 675–691. 3

[XCZ*18] XU W., CHATTERJEE A., ZOLLHÖFER M., RHODIN H., MEHTA D., SEIDEL H.-P., THEOBALT C.: Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG) 37*, 2 (2018), 1–15. 3

[XYS*19] XU Y., YANG S., SUN W., TAN L., LI K., ZHOU H.: 3D virtual garment modeling from RGB images. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Beijing, China, 2019), IEEE, pp. 37–45. 2

[YFHWW16] YANG J., FRANCO J.-S., HÉTROY-WHEELER F., WUHRER S.: Estimation of human body shape in motion with wide clothing. In *European Conference on Computer Vision* (Amsterdam, The Netherlands, 2016), Springer, pp. 439–454. 3

[YFHWW18] YANG J., FRANCO J.-S., HÉTROY-WHEELER F., WUHRER S.: Analyzing clothing layer deformation statistics of 3D human motions. In *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 237–253. 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 Transactions on Graphics (TOG) 37*, 5 (11 2018), 1–14. 2

[ZCJ*20] ZHU H., CAO Y., JIN H., CHEN W., DU D., WANG Z., CUI S., HAN X.: Deep fashion3d: A dataset and benchmark for 3D garment reconstruction from single images. In *European Conference on Computer Vision* (2020), Springer, pp. 512–530. 2

[ZPBPM17] ZHANG C., PUJADES S., BLACK M., PONS-MOLL G.: Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Hawaii, USA, 2017). 2, 3, 14

Appendix A: Body Shape and Pose Estimation

Pose Initialization. OpenPose [CHS*19] is used to estimate the pose of the input 3D scan. For this, we obtain multi-view images by rendering the 3D scan input from 7 equally spread directions. The renderings and the camera parameters are fed to OpenPose, and the estimated pose is obtained. As a result, we obtain the initial pose $\vec{\theta}_{op}$ for the following SMPL paramter estimation step.

Model Fitting. To obtain SMPL model parameters $\vec{\beta}, \vec{\theta}$ that approximate the body shape under clothing for the input 3D scan, we perform energy minimization with the following objective:

$$E(\vec{\beta},\vec{\theta}) = E_{fit}(\vec{\beta},\vec{\theta}) + \lambda_{\beta} E_{\beta}(\vec{\beta}) + \lambda_{\theta} E_{\theta}(\vec{\theta}), \tag{7}$$

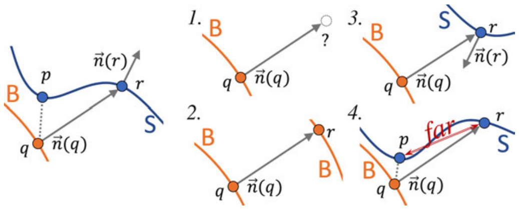

Figure 19: *Conditions to determine ill-projected feature point.*

where E_{fit} is a fitting term, and $\lambda_\beta E_\beta$ and $\lambda_\theta E_\theta$ are regularization terms for the shape and pose, respectively. Fitting starts with $\vec{\beta} = \vec{0}$, i.e., an average shape of SMPL model, and $\vec{\theta} = \vec{\theta}_{op}$. Fitting term follows the idea of the cloth term in *single-frame objective* of [ZPBPM17]. Fitting objective is the sum of two terms:

$$E_{fit}(\vec{\beta},\vec{\theta}) = \lambda_{out} E_{out}(\vec{\beta},\vec{\theta}) + \lambda_{in} E_{in}(\vec{\beta},\vec{\theta}), \quad (8)$$

where E_{out} penalizes model vertices that penetrate the input scan mesh M_{scan}, while E_{in} encourages the model vertices that are inside M_{scan} to remain close to the surface.

$$E_{out}(\vec{\beta},\vec{\theta}) = \sum_{v_i \in V_{smpl}} (1-\delta_i) dist(v_i, M_{scan})^2 \quad (9)$$

$$E_{in}(\vec{\beta},\vec{\theta}) = \sum_{v_i \in V_{smpl}} \delta_i \, GM(dist(v_i, M_{scan})^2) \quad (10)$$

where $dist(point, surface)$ calculates the distance for a point to the closest primitive on the surface, δ_i is an indicator function that is 1 when a vertex is inside M_{scan} and 0 otherwise, and $GM(x)$ is a Geman-McClure smoothing function used here to handle outliers on noisy scans. To calculate δ_i, we utilize the method of [BA05] implemented in [JP*18]. We use the $L2$-norm regularization on shape parameters $E_\beta(\vec{\beta}) = ||\beta||^2$. In practice, shape parameter regularization prevents compensation of pose estimation errors by extreme shape adjustments, e.g., shortening limbs. We regularize pose to be close to the initial guess $\vec{\theta}_{op}$ using a Mahalanobis distance $E_\theta(\vec{\theta}) = (\vec{\theta} - \vec{\theta}_{op})^T \Sigma_\theta^{-1} (\vec{\theta} - \vec{\theta}_{op})$, where Σ_θ^{-1} is a covariance matrix computed from the pose training set of [LMR*15] as part of the work of [ZPBPM17].

Implementation. We additionally increase robustness by applying classical ICP heuristics. We ignore far-distance vertices, and those vertices whose normal point to the opposite direction from the normal of the closest primitive on M_{scan}. For optimization, we use Levenberg–Marquardt algorithm implemented in [AMO], with analytically computed derivatives. To prevent issues with parameter scaling, optimization is performed individually for each group of parameters: translation, shape, and joints rotation, in this order.

Appendix B: Removal of Ill-projected feature points

We use the following simple heuristics to determine ill-projected points to be discarded. Let q and p denote a feature point on SMPL body mesh B and its closest distance projected point on scan mesh S, respectively. We also shoot a ray from q on its positive and negative normal $\vec{n}(q)$ directions and find the closest intersection point r with either B or S. We discard the projected point p if one of following condition is met (Fig. 19):

1. r does not exist.
2. r is found on B.
3. The normal directions $\vec{n}(q)$ and $\vec{n}(r)$ from q and r are almost opposite: $\vec{n}(p) \cdot \vec{n}(r) < -0.5$.
4. p and r are too far apart: $|p - r| > 0.4L$, where L is the length of body.

Note that we only use the closest distance projected point p. The ray-projected point r is only for validating p.