# Automatic Pose and Wrinkle Transfer for Aesthetic Garment Display


## Abstract

We present an automatic and semantic pose and wrinkle transfer method from one garment onto another for aesthetic display, which is previously performed by professional artists using a knowledge-intensive and time-consuming process. Given a source garment model with fine wrinkle details in a specific pose and another target garment model with a similar style in a neutral pose but without fine wrinkle details, our approach can automatically transfer the pose and wrinkle details faithfully from the source to the target using a two-stage process. In the semantic correspondence establishment stage, we construct a dense correspondence between the source and the target by utilizing their semantic information in 2D patterns. Specifically, we first obtain the initial correspondence points on the paired 2D patterns by leveraging their semantic information. These marker points, which act as constraints, are mapped to their corresponding 3D models. We then establish their per-triangle correspondence using a non-rigid Iterative Closest Point (ICP) algorithm. In the deformation transfer stage, we transfer the pose and wrinkle details from the source to the target by solving an optimization problem. Extensive experiments validate that our method is able to generate better results compared to state-of-the-art methods, and it can lead to significant time savings for fashion designers.



*Keywords:* Garment wrinkles, deformation transfer, aesthetic garment display, semantic information


## 1. Introduction

Online shopping for apparel has been the fastest growing sales channel in the last decade. In recent years, lots of clothing retailers are beginning to sell their apparel products using digital samples for fast fashion. These digital garments are aesthetically displayed with rich wrinkle details in some specific poses in order to attract more customers. Creating such high-quality digital garments is a laborious and knowledge intensive task. This process involves 2D pattern design, pattern arrangement on the mannequin, cloth simulation (Browzwear, 2000-2020; CLO, 2020; Optitex, 1988-2020), and fine-tuned sculpting (ZBrush, 2020). For interactive design, a simulated cloth model is usually created with a limited resolution. To achieve a more visually appealing appearance, the simulated cloth model is often sculpted to add fine high-frequency details to the garment using a digital sculpting tool like ZBrush (ZBrush, 2020). Such sculpting tools use dynamic levels of resolution to allow sculptors to make local changes to their models. However, such a sculpting process may take an experienced designer hours to create a satisfactory result. For a garment with the same or a similar design, the time-consuming sculpting process has to be repeated practically from scratch, even the only difference is the size. This motivates us to develop an automatic method to eliminate the laborious sculpting process.

The above problem can be considered as the deformation transfer from a source triangle mesh onto a target triangle mesh (Sumner and Popovic, 2004), which has received considerable attention in the past decades. In (Sumner and Popovic, 2004), the user needs to manually build a correspondence map between the triangles of the source and those of the target by specifying some vertex markers. Coating transfer can also be employed to switch geometric details between meshes with different topologies (Sorkine et al., 2004). However, this method may result in a certain degree of "blurring" artifacts. GeoBrush (Takayama et al., 2011) first selected ROIs on the source and target models interactively, and then cloned the arbitrary high-resolution surface features on the source model continuously to the specified area of the target model in real time. However, this approach cannot handle large pose changes. Using displacement maps, the details of high-quality meshes can be transferred to low-resolution meshes using metric learning (Berkiten et al., 2017). Nevertheless, their method ignored pose transfer and the generated details were not identical to the source, especially for the characteristics of wrinkles in specific parts of the models. The digital garments for online display are usually quite dense, ranging from 10k triangles to 100k triangles. Specifying vertex markers for these kind of garment models accurately is both time consuming and error prone. To eliminate the laborious sculpting process for garments, the key challenge is to establish the correspondence map for deformation transfer automatically while preserving the pose and wrinkle details faithfully. State-of-the-art methods are either not automatic (Sumner and Popovic, 2004; Takayama et al., 2011) or unable to generate high-quality transfer results (Sorkine et al., 2004; Berkiten et al., 2017).

For two garments with similar styles, or different sizes of the same style, their 2D patterns are usually similar, as shown in Figure 2. Based on this observation, we try to develop an automatic method to transfer the pose and wrinkles from one garment onto another faithfully to reduce the time consuming sculpting process by making full use of the semantic information of the 2D patterns of garments. We introduce a novel strategy for building semantic correspondences on garments automatically to replace the manual marking process in traditional deformation transfer methods. Unlike other types of models, 3D garment models have special semantics in their corresponding 2D patterns, which are utilized in our paper to establish dense correspondences between the source and target models. Since each 2D pattern corresponds to a particular body part, the semantic information is implied in its attributes and geometric contour features. Thus, a novel framework is presented to quickly generate 3D high-quality garment models with specific pose and wrinkle information transferred from high-quality source models.

In summary, our work makes the following contributions:

- A novel automatic and semantic deformation transfer framework designed for 3D garments, which can faithfully transfer the fine-grained pose and wrinkle details of high-resolution source models to target models in a neutral pose with similar styles, or different sizes of the same style, for aesthetic online garment display in e-commerce.
- A semantic correspondence establishment strategy to automatically align the source and target models by leveraging the semantic information of 2D patterns unique to garments.
- A 3D high-quality garment dataset consisting of 18 paired models of various types. Each pair consists of a garment and its fine-tuned sculpted counterpart as well as their patterns for further research.

## 2. Related Work

*Garment Modeling with Wrinkles.* For the past decades, great efforts have been made to enhance the realism of garment models. Researchers use either dynamic or static methods to obtain geometric details such as fine wrinkles on garments. Traditional physics-based simulation (PBS) methods (Jiang et al., 2017; Selle et al., 2009) can simulate garments with rich geometric details but at the cost of high-resolution mesh and much computation time. Other works compromise accuracy and physical correctness for speed (Bouaziz et al., 2014; Müller et al., 2007; Müller, 2008). Besides, many approaches try to find a balance between speed and accuracy by adding wrinkle details on low-resolution coarse meshes (Gillette et al., 2015; Goldenthal et al., 2007; Kavan et al., 2011; Kim et al., 2013; Rohmer et al., 2010). Data-driven methods (de Aguiar et al., 2010; Guan et al., 2012; Kim et al., 2013; Wang et al., 2010; Xu et al., 2014) can provide fast simulation by trading space for time or using some kind of approximation. This method in (Wang et al., 2010) leveraged a precomputed database to enhance the low-resolution clothing simulation based on joint proximity locally. However, the synthesized mesh may have more wrinkles compared to the ground truth. To bridge the gap between skinning and physical simulation, de Aguiar et al. (2010) learned and preserved essential dynamic properties of cloth motions with corresponding folding details. Near-exhaustive precomputation method can generate realistic secondary motion of clothing deformations by exhaustively searching a motion graph (Kim et al., 2013). However, the method relies on a large database which requires a high memory space and computational cost. Dressing arbitrary body shapes in any poses without physical simulations can also be simulated with a learned model (Guan et al., 2012). In recent years, deep learning-based garment modeling receives considerable attention. Lähner et al. (2018) presented a method to generate accurate and realistic clothing deformation from real data capture with a conditional Generative Adversarial Network. However, artifacts may arise because of the occlusion of the body from the camera view.

*Sketch-based Garment Modeling.* Another way to construct garment models is based on sketches. Folds were modeled by sweeping a cylindrical profile along a user sketched path in (Turquin et al., 2007). Jung et al. (2015) presented a method to generate smooth developable surfaces with pre-designed wrinkles from multi-view sketches, and this method focused on the garments with stiff fabric or leather products. More recently, BendSketch (Li et al., 2017) translated user input into surface detail geometry but cannot guarantee that the resulting folds are physically plausible. Besides, Li et al. (2018) presented an interactive system to support intuitive fold and pleat design, and generated physically reproducible fold-enhanced garments . These sketch-based methods can generate fold-enhanced garment models, but the results lack fine wrinkle details that are equally essential for clothing display. In (Wang et al., 2018), the data-driven learning framework could generate 3D draped garment shapes similar to the input 2D sketches. However, it may not satisfy the fold and wrinkle details induced by external forces applied to the garment. Although the aforementioned works can produce garment models with plausible folds and wrinkles, compared to the manually fine-sculpted garment models, they cannot guarantee the aesthetics and faithfulness of the geometric details for online apparel display in e-commerce.

*Garment Modeling Using Commercial Software.* Existing popular 3D garment design software packages (Browzwear, 2000-2020; CLO, 2020; Optitex, 1988-2020) adopt a 2D-to-3D design pipeline. Designers manually edit 2D patterns, sew them, and simulate the draping effects of the garment on the mannequin with some interactive methods. By specifying different cloth parameters, designers can obtain realistic wrinkles and folds of diverse cloth materials. However, the simulated result is usually not pleasing enough for online display. A time-consuming sculpting process is frequently employed to add fine details to the designed garments using professional software such as (ZBrush, 2020) in order to generate high-quality garment models. Our objective is to automate this laborious process for garments with the same or similar styles.

*Shape Correspondence.* A key step in our algorithm is to establish a semantically correct dense correspondence between the source and target. Shape correspondence and its applications have been extensively investigated in the past decades (Biasotti et al., 2016; van Kaick et al., 2011; Tam et al., 2013). Early shape matching methods focused on establishing pointwise correspondences between two non-rigid shapes (Huang et al., 2008; Ovsjanikov et al., 2010; Tevs et al., 2009). Later methods (Kezurer et al., 2015; Maron et al., 2016; Solomon et al., 2016) managed to deal with the computational complexity of quadratic assignment problems to reduce the computation cost. Recently, functional maps (Kovnatsky et al., 2013; Ovsjanikov et al., 2012; Rodolà et al., 2017) have been employed for non-rigid shape matching. Even orientation-preserving and bijective pointwise correspondences between non-rigid shapes can also be dealt with this kind of methods (Ren et al., 2018). For deformation transfer, a non-rigid iterated closest point algorithm with regularization was employed using the user-selected marker points to establish a per-triangle correspondence between the source and tagert models (Sumner and Popovic, 2004). In our scenario, the topology and semantics of garments are quite complex, and manually selecting marker points is time-consuming and error-prone. Different from the aforementioned methods, our garment models have their unique semantic information, and we make full use of these information stored in 2D patterns to automatically establish a dense correspondence.

*Mesh Deformation Transfer.* Deformation transfer (Sumner and Popovic, 2004) presented an efficient way to copy the deformations exhibited by pose and subtle geometric changes from a source mesh onto a different target mesh by manually specifying some maker points. It can also be extended to accommodate complex models consisting of multiple arbitrary components (Zhou et al., 2010). To reduce the laborious process of mesh animation, a semantic method was developed to transfer existing mesh deformation from one character to another by inferring a correspondence between the shape spaces of two characters (Baran et al., 2009). However, this method is designed to preserve the semantic characteristics of the motion. To reduce the substantial user effort to label the correspondences, a VAE-Cycle GAN (VC-GAN) network-based method is developed to automate the deformation transfer between two unpaired shape sets (Gao et al., 2018). Our approach differs from these methods in that we make use of the semantic information of 2D patterns associated with their corresponding garments.

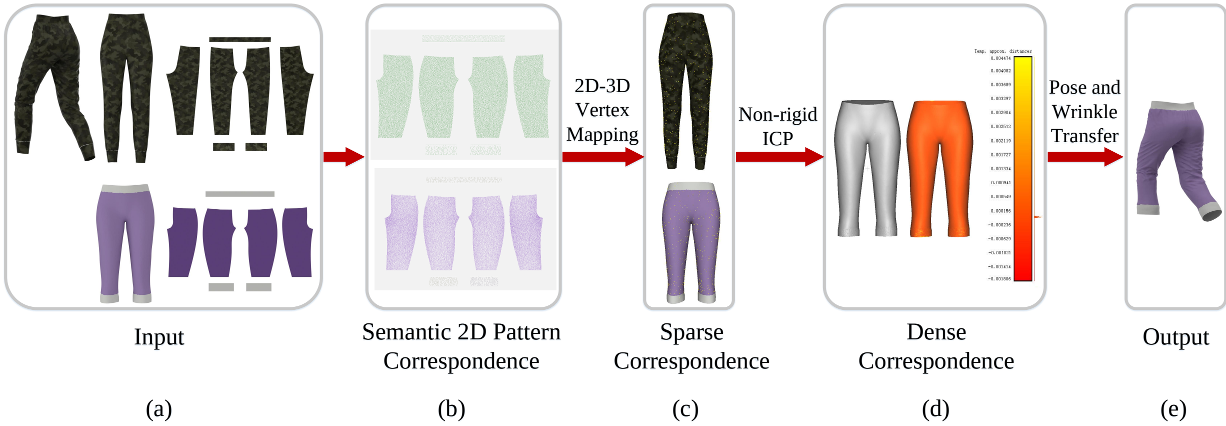


Figure 1: Algorithm overview. Our input is a source garment model with fine wrinkle details (the leftmost image) and its 2D patterns (the rightmost image) as well as its neutral pose without sculpting (the middle image)) (See the upper part in (a)), and another target garment model of a similar style in a neutral pose without fine wrinkle details as well as its 2D patterns (See the lower part in (a)). Note that the necessary 3D high-quality information for (a) is readily available in our database, which contains garments designed by professional garment modeling software. We first obtain a set of pointwise correspondence by utilizing the semantic information of 2D patterns (b). Then, we map these point pairs to their corresponding 3D models as the patterns and its corresponding garment model share the same UV coordinates (c). After that, we perform a non-rigid ICP algorithm with regularization, using the sparse point-to-point correspondence as constraints, to establish a per-triangle dense correspondence between the source and target (d). Deformation transfer is finally employed to generate the posed output with wrinkle details similar to the high-quality source model (e).

## 3. Methodology

The input of our method consists of source and target garment models as well as their 2D patterns, which are designed by professional designers using commercial garment modeling software (See Figure 1). The user specifies a garment without pose and detailed wrinkles as the target $T$, and then our system searches for another garment with the same or similar style from the database as the source $S$, both of which requires a draping garment model simulated by wearing on an A-pose mannequin. Moreover, the input source garment consists of an additional model, denoted as deformed source $S'$, which contains the pose and detailed wrinkle information of the source garment designed for online display. The fine wrinkles of $S'$ are carefully sculpted by professional artists for aesthetic display. For each garment, such a sculpting process takes about three hours in average using Zbrush. Our objective is to transfer the pose and wrinkle information from the source to the target so that the target may have a visually similar appearance with fine wrinkle details.

As shown in Figure 1, we first establish an initial vertex correspondence between the 2D patterns of the source and the target according to their semantic information. Then, we map the paired points onto their corresponding 3D models because the 2D patterns and their 3D garment counterpart share the same set of UV coordinates. The mapped points on the reference source and target models are enforced as constraints of the non-rigid iterated closest point (ICP) algorithm for establishing a per-triangle source-to-target correspondence. Finally, we generate the output by performing a deformation transfer method similar to the one in Sumner and Popovic (2004).

*Semantic Correspondence.* We utilize the semantic information of 2D patterns to establish a dense per-triangle correspondence between the source and target. A complete garment is usually sewn from multiple patterns, each of which has its specific attribute (Figure 2, *e.g., body_front, sleeves_left, etc.*). Garments of the same or similar styles generally have similar 2D patterns (Figure 2). Such a similarity of patterns is implied in the corners on the outline. The attributes and the corresponding geometric contour features constitute the semantic information of the patterns. We generate an initial pointwise correspondence automatically with the help of semantic information associated with two given patterns, which is effective in solving the correspondence problem.

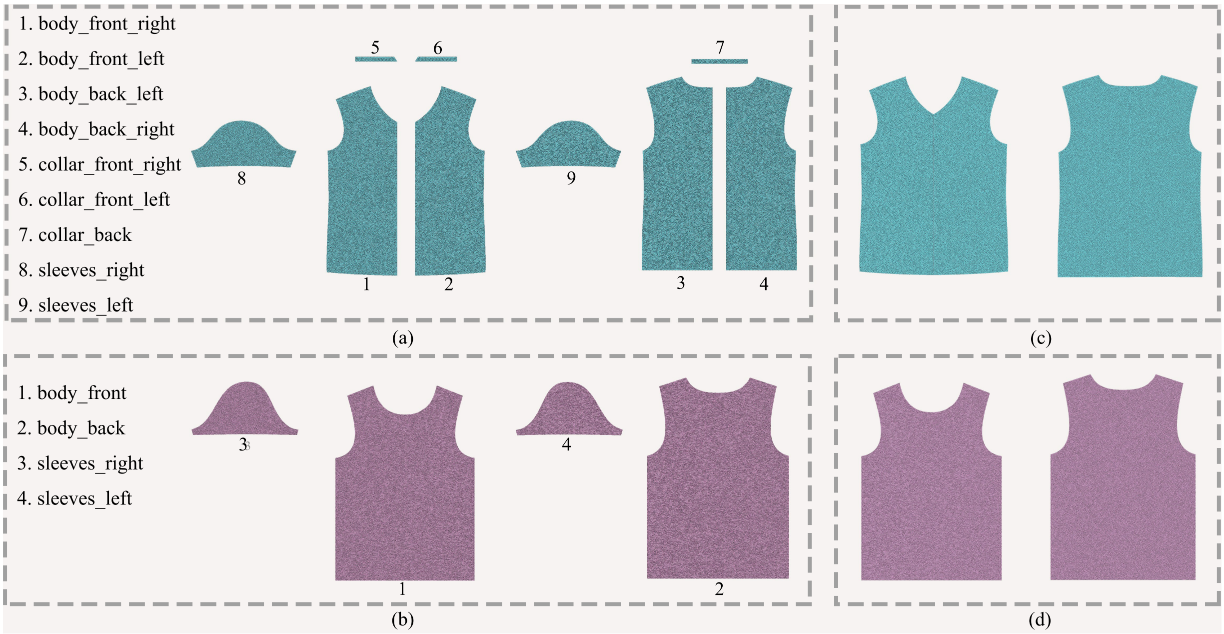


Figure 2: 2D patterns of a V-neck T-shirt and a round-neck T-shirt of similar style, along with the merge result of part of the V-neck T-shirt patterns for correspondence-building between two sets of patterns. (a) 2D patterns of a V-neck T-shirt. (b) 2D patterns of a round-neck T-shirt. (c) Merged pattern results of pattern 1 with 2, and pattern 3 with 4 of the V-neck T-shirt. (d) The body_front and body_back patterns of the round-neck T-shirt correspond to the two merged patterns in (c), respectively.

*2D-3D Mapping.* The mesh topologies of a simulated model and its patterns are usually different in garment modeling software (*e.g., (CLO, 2020)*). Therefore, we cannot establish the correspondence naively according to the vertex indices of the 2D patterns and their corresponding 3D model. Fortunately, the simulated 3D model and its 2D patterns share the same UV set, and therefore we can utilize this information as a bridge to determine the initial vertex constraints between 2D patterns and its corresponding 3D model by searching for vertices that have the same UV coordinates.

*Pose and Wrinkle Transfer.* Before deformation transfer, we should establish the per-triangle dense correspondence between the source and target. Here, we use the corresponding vertices obtained from our semantic correspondence algorithm as the constraints, and perform a non-rigid ICP algorithm with regularization to infer the per-triangle dense correspondence. After that, we employ the method described in (Sumner and Popovic, 2004) to perform pose and wrinkle transfer on garments to generate our final output garment with the same pose and wrinkle details as the one in $S'$.

## 4. Preprocessing

To perform our experiments, we build a high-quality garment database with fine wrinkle details. This database consists of 18 garments with various styles, and all of them are designed by professional fashion designers from fashion industry. Taking the lantern sleeve dress as an example (Figure 3(a)), we first arrange 2D patterns around an A-pose character (Figure 3(b)), and then perform physics-based simulation to obtain an original draping model. We denote the draping source model and its patterns as $S$ and $S_{patterns}$, respectively. After that, we apply the same patterns on a posed-character (Figure 3(c)) and obtain another garment $S_{pose}$ with a visually pleasing appearance by interactively applying some external forces on certain parts of the garment. An additional manually sculpting process is further employed to generate the garment $S'$ with special fine wrinkles that cannot be simulated by up-to-date professional software. This sculpting process is operated by professional artists using Zbrush. Note that $S$, $S_{pose}$, $S'$, and $S_{patterns}$ make up the

Figure 3: Part of the garment modeling steps. A lantern sleeve dress (a), its draping model on an A-pose female character (b), and the garment with a visually pleasing appearance on the posed character (c).

source part of our input, and they share the same UV set. Similarly, for a given target garment, we denote the simulated 3D model draping around the A-pose character and its corresponding set of 2D patterns as $T$ and $T_{patterns}$, respectively. To achieve a better wrinkle transfer effect, we smooth garments $S$ and $T$ using the open-source implementation of (Vollmer et al., 1999) in (Trimesh, 2020) to filter out unnecessary wrinkle details generated by the simulation phase, and finally get the source reference $\tilde{S}$ and target reference $\tilde{T}$.

## 5. Semantic Correspondence

### *5.1. Semantic Information of 2D patterns*

Each pattern is represented by an attribute and its contour as mentioned above. A shirt example is shown in Figure 2. For each pattern that makes up the corresponding 3D garment, its attribute is defined according to the body parts it covers (Figure 2 (a)(b)). We can thus quickly locate similar source and target patterns using such attributes. Furthermore, for paired source and target patterns, we can use the geometric features of the corners of their contours to obtain accurate point correspondences. However, two garments with a similar style cannot guarantee that they have the same number or attributes of patterns. To this end, we introduce *Pattern Merge* and *Pattern Split* operations to tackle this problem. From the garments' semantic information, we can find the pattern pairs that need to be merged and the corresponding edges with the same number of vertices. We use the corresponding edges as boundary constraints and perform a non-rigid ICP algorithm (Sumner and Popovic, 2004) to align one pattern to another in the same pairs. Finally, we merge the vertices in the corresponding edges to connect the two patterns.

### *5.2. Pattern Preprocessing for Special Cases*

*Pattern Merge.* A pattern merge example is shown in Figure 2, where (a) and (b) display a V-neck T-shirt and a round-neck T-shirt without one-to-one pattern correspondence. In this case, patterns 8 and 9 in (a) have the same attributes with patterns 3 and 4 in (b), respectively. Therefore, they can directly correspond to each other according to the attributes. However, even for patterns with the same attributes, there are no corresponding patterns in (b) for patterns 1, 2, 3 and 4 in (a). Fortunately, according to the semantic information of the garment, patterns 1 and 2 in (a) can be considered equivalent to pattern 1 in (b) by dividing it into two parts. Therefore, we can merge patterns 1 and 2 in (a) to get a new merged pattern and assign the attribute body_front (Figure 2 (c), left) to it. Similarly, we can perform the same operation on patterns 3 and 4 in (a) to get another new pattern body_back (Figure 2 (c), right). In this way, patterns 1 and 2 in (a) and patterns 3 and 4 in (a) can find their corresponding patterns in (b) to obtain

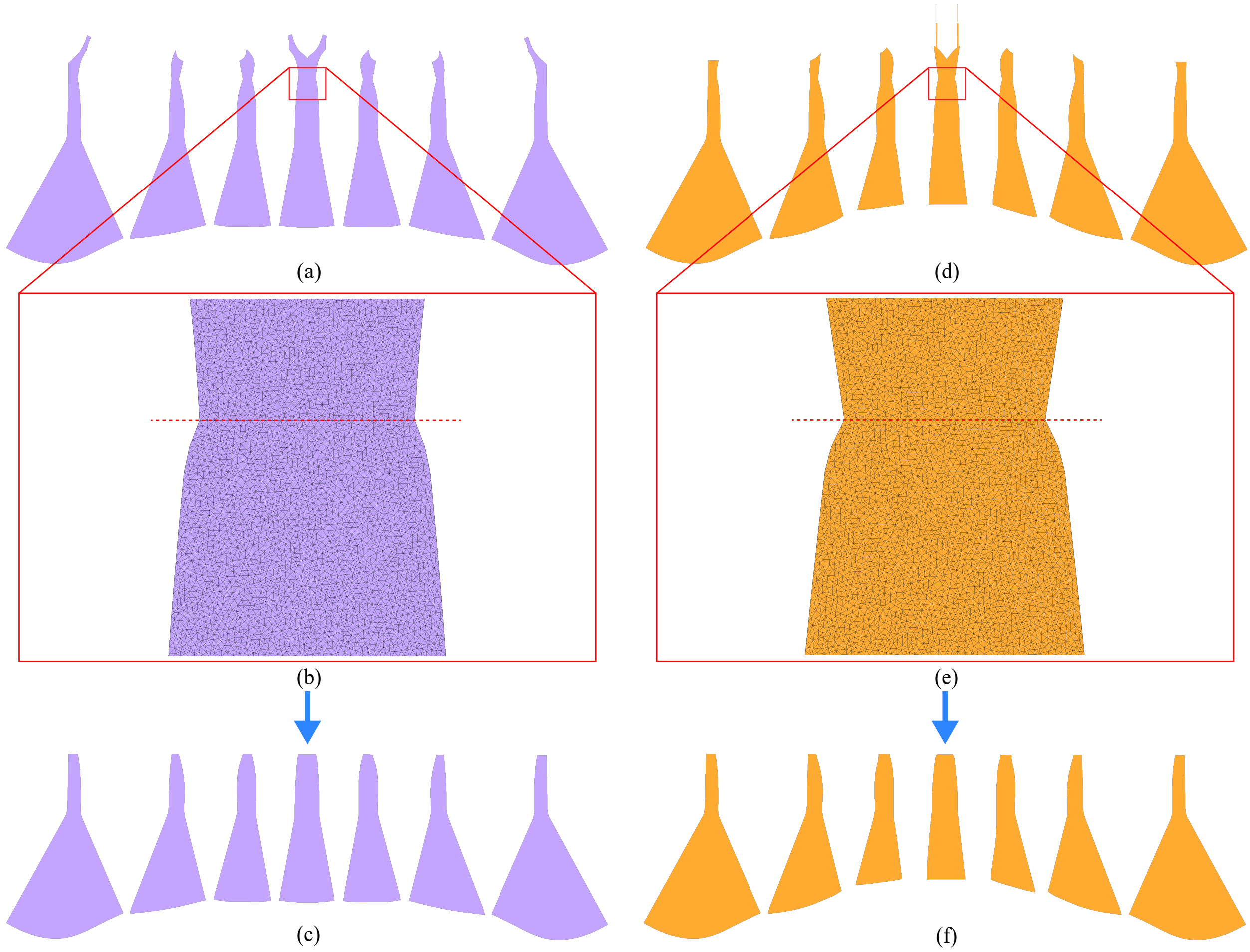


Figure 4: Pattern split example. Patterns of the source gown are shown in purple (left) and patterns of the target gown are shown in orange (right). We split all patterns ((a) and (d)) from the position semantically corresponding to the waistline of the human body (the red lines shown in (b) and (e)), so as to obtain new patterns corresponding to the dress part of the gowns ((c) and (f)). Note that the outlines of the two patterns in (c) and (f) (from left to right) are very similar, which ensures that we can obtain accurate correspondences between the two split sets of patterns in the subsequent steps.

the initial semantic correspondence. Patterns 5, 6 and 7, which make up the neckline in (a), can be ignored in the initial correspondence establishment process since there is no corresponding pattern in (b).

*Pattern Split.* For the case of two gowns with similar styles (See Figure 4), the contours of 2D patterns of the source and target are different in the upper part but quite similar in the lower dress part. In such a scenario, we divide the model into the upper and lower parts along the waistline and process them separately, since the changes of pose and wrinkle details usually happen in the lower parts.

### *5.3. Pattern Corner Determination*

Our algorithm ensures that each pattern in the source patterns can find a corresponding pattern in the target patterns with the employment of the semantic pattern attribute. Given one pair of the source and target patterns denoted as $(\boldsymbol{P}_i^S, \boldsymbol{P}_j^T)$, we can establish an initial correspondence using the rigid ICP algorithm (Jubran et al., 2021). We note that the artists set the same orientations for $(\boldsymbol{P}_i^S, \boldsymbol{P}_j^T)$ when creating them. Thus we can simply align $\boldsymbol{P}_i^S$ and $\boldsymbol{P}_j^T$ with scale and translation variations. We traverse the whole mesh of each pattern of $(\boldsymbol{P}_i^S, \boldsymbol{P}_j^T)$ to find the source contour $\boldsymbol{V}_C^S = \{\boldsymbol{v}_1^S, \boldsymbol{v}_2^S, ..., \boldsymbol{v}_m^S\}$ and the target contour $\boldsymbol{V}_C^T = \{\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, ..., \boldsymbol{v}_n^T\}$, respectively. For a contour vertex, we calculate the cosine value of the angle $\gamma$ between its two adjacent contour edges. When $\cos\gamma > -0.8$, we take the angle as a corner of the pattern. By traversing all the contour vertices, we can get the corner set of the patterns $(\boldsymbol{P}_i^S, \boldsymbol{P}_j^T)$:

$$\boldsymbol{C}^S = \{\boldsymbol{v}_{c_1^S}, \boldsymbol{v}_{c_2^S}, ..., \boldsymbol{v}_{c_p^S}\} \subset \boldsymbol{V}_C^S, \tag{1}$$

$$\boldsymbol{C}^T = \{\boldsymbol{v}_{c_1^T}, \boldsymbol{v}_{c_2^T}, ..., \boldsymbol{v}_{c_q^T}\} \subset \boldsymbol{V}_C^T, \tag{2}$$

where $p, q$ are the corner numbers of the source pattern and the target pattern, respectively. Although $(\boldsymbol{P}_i^S, \boldsymbol{P}_j^T)$ are similar in their contour shape, we note that their numbers of corners are not necessary to be the same.

### 5.4. Corner Vertex Correspondence

Before establishing the correspondence between the corners of the source pattern $P_i^S$ and the target pattern $P_j^T$, we first align them in the three-dimensional coordinate system by scaling and translation. Then we calculate the Laplacian coordinates (Sorkine, 2005) of all vertices of the source pattern $P_i^S$, which is denoted as $\boldsymbol{\Delta}_S = \{\delta_1^S, \delta_2^S, ...\}$. Similarly, we calculate the Laplacian coordinates of all vertices of the target pattern $P_j^T$, and denote them as $\boldsymbol{\Delta}_T = \{\delta_1^T, \delta_2^T, ...\}$. Let $\{\boldsymbol{v}_{c_k^S} \in \boldsymbol{C}^S, 1 \leqslant k \leqslant p\}$ be a corner vertex of the source pattern $P_i^S$, and its Laplacian coordinate $\boldsymbol{\delta}_{c_k^S}$. We establish the following energy function:

$$E = -(\boldsymbol{\delta}_{c_k^S} \cdot \boldsymbol{\delta}_{c_l^T} - \|\boldsymbol{v}_{c_k^S} - \boldsymbol{v}_{c_l^T}\|^2). \tag{3}$$

The first term of the function is the dot product of the Laplacian coordinates of $\boldsymbol{\delta}_{c_k^S}$ and $\boldsymbol{\delta}_{c_l^T}$, and the second term is the square distance between the two vertices. The smaller the value of the energy function in Equation 3 is, the higher the semantic similarity of the two corner vertices compared by the function. With the source pattern corner vertex $\boldsymbol{v}_{c_k^S}$, we suppose that the energy function in Equation 3 is minimized when the target pattern corner vertex $\boldsymbol{v}_{c_l^T}$ is taken. Similarly, for the corner vertex $\boldsymbol{v}_{c_l^T}$ of the target pattern, we assume that the energy function gets the minimum when the corner vertex of the source pattern takes $\boldsymbol{v}_{c_k^S}$. Then $\boldsymbol{v}_{c_k^S}$ and $\boldsymbol{v}_{c_l^T}$ are considered as a mutually corresponding vertex pair. By traversing all the corner vertices of the paired source pattern $P_i^S$ and target pattern $P_j^T$ using the method described above, we can build a one-to-one mapping between part of the source and target corner vertices and get an initial sparse vertex correspondence index set:

$$\boldsymbol{M}_C = \{(c_{r_0}^S, c_{r_0}^T), (c_{r_1}^S, c_{r_1}^T), ..., (c_{r_{nc}}^S, c_{r_{nc}}^T)\}. \tag{4}$$

### 5.5. Vertex Correspondence Generation

Taking the obtained vertex pairs in $\boldsymbol{M}_C$ as constraints, we perform Laplacian deformation on the vertices $\boldsymbol{V}_C^S = \{\boldsymbol{v}_1^S, \boldsymbol{v}_2^S, ..., \boldsymbol{v}_m^S\}$ that constitute the contour line of the source pattern $P_i^S$ by iterative optimization:

$$E(\boldsymbol{V}_C'^S) = \sum_{\alpha=1}^{m} \|\boldsymbol{L}(\boldsymbol{v}_\alpha'^S) - \boldsymbol{\delta}_\alpha^S\|^2 + \sum_{(c_{r_\beta}^S, c_{r_\beta}^T) \in \boldsymbol{M}_C} \|\boldsymbol{v}'_{c_{r_\beta}^S} - \boldsymbol{v}_{c_{r_\beta}^T}\|^2, \tag{5}$$

where $\boldsymbol{V}_C'^S = \{\boldsymbol{v}_1'^S, \boldsymbol{v}_2'^S, ..., \boldsymbol{v}_m'^S\}$ is the deformed source pattern contour vertices set to be optimized, and $\boldsymbol{L}$ is the Laplace operator. The first term of Equation 5 indicates that the geometric features of deformed source pattern contour vertices should be as close as possible to those before the deformation, in which $\{\boldsymbol{\delta}_1^S, ..., \boldsymbol{\delta}_m^S\}$ is the Laplacian coordinate set of the original source pattern contour vertices. The second term indicates that the new position of each paired corner vertex on the source pattern should be equal to the corresponding point on the target pattern.

Based on the Laplacian deformation results, we can calculate the one-to-one correspondence of the non-corner vertex of the source pattern $P_i^S$ and target pattern $P_j^T$ by finding the aforementioned mutually closest point, and get the new correspondence vertex pairs on the contour. We add these new pairs to $\boldsymbol{M}_C$ and denote the updated vertex pair set as $\boldsymbol{M}'_C$.

By setting the updated $\boldsymbol{M}'_C$ as constraints, we perform another Laplacian deformation on all the vertices of the source pattern $P_i^S$. A deformed source pattern $P_i'^S$ is obtained through an iterative solution, and $P_i'^S$ and $P_j^T$ are well-aligned at the same time. We then traverse all vertices on $P_i'^S$ and $P_j^T$. When a pair of vertices is closest to each other, it is considered that these two vertices are corresponded. Through the above-described method, we can finally generate a dense vertex correspondence between the paired patterns.

### *5.6. 2D-3D Vertex Mapping*

Since 2D patterns are the unfolded cloth of 3D garment models, an edge stitched on the 3D garment model corresponds to at least two edges of different patterns, which causes different topologies between the simulated garment and its corresponding patterns in the clothing modeling software (e.g., (CLO, 2020)). Nevertheless, for each vertex $\boldsymbol{v}_{i_p}$ in the pattern, there exists a corresponding vertex $\boldsymbol{v}_{i_s}$ in the simulated model. As discussed in Section 4, patterns and the simulated model share the same UV set, and therefore $\boldsymbol{v}_{i_s}$ can be located easily according to the UV coordinates of its $\boldsymbol{v}_{i_p}$. As we leverage the semantic information of 2D patterns to establish a one-to-one vertex correspondence between the 3D source and target garment models, our method can deal with models with different topologies. By now, we can obtain a vertex constraint set $M_{cons}$ between $\tilde{S}$ and $\tilde{T}$:

$$M_{cons} = \left\{(\boldsymbol{v}_{i_1}^S, \boldsymbol{v}_{i_1}^T), (\boldsymbol{v}_{i_2}^S, \boldsymbol{v}_{i_2}^T), ..., (\boldsymbol{v}_{i_m}^S, \boldsymbol{v}_{i_m}^T)\right\}. \tag{6}$$

By setting $M_{cons}$ as constraints, we perform the non-rigid iterated closest point with regularization to obtain per-triangle correspondences between $\tilde{S}$ and $\tilde{T}$. Finally, we employ the deformation transfer method to transfer the pose and wrinkles from $S_{sculpted}$ to the target $\tilde{T}$. For more details about the non-rigid ICP and deformation transfer algorithm, please refer to (Sumner and Popovic, 2004).

## 6. Experiments

We have implemented our presented framework in C++ on a desktop computer equipped with Intel® Core™ i7-7700K CPU at 4.2GHz, 32.0GB of RAM, and NVIDIA® GeForce GTX 1060 GPU. The run-time statistics are presented in Table 1, which demonstrates that our proposed method can efficiently generate new high-quality garment models on an off-the-shelf computer. As shown in Table 1, the time for semantic 2D pattern correspondence step and the non-rigid ICP step vary from a few seconds to a few minutes, depending on the number of vertices of the input meshes and the complexity of garment patterns. The deformation transfer process takes seconds on average without optimization. The total running time of our framework for generating a target model displayed in our paper is 1-2 minutes on average. The experimental group of gowns has the largest number of faces. The source model has 403.8k faces and the target has 364.5k faces, but the total running time is still less than 9 minutes.

Table 1: Runtime statistics for each pair of models.

| | # Source faces | # Target faces | CB (2D)[1] | CB (ICP)[2] | DT[3] | Total |
|---|---|---|---|---|---|---|
| Gowns | 403.8k | 364.5k | 329.148s | 135.608s | 42.605s | 507.361s |
| Women's casual trousers | 116.9k | 86k | 44.405s | 9.98s | 6.149s | 60.534s |
| Lantern sleeve dresses | 18.4k | 23.3k | 136.274s | 9.512s | 7.856s | 153.642s |
| Qipao and polo dress | 127.4k | 107.6k | 63.87s | 57.175s | 5.813s | 126.858s |
| Men's sports suit top | 150k | 122.9k | 60.017s | 71.62s | 13.786s | 145.423s |
| Men's sports suit bottom | 83.5k | 78k | 24.8s | 34.923s | 3.918s | 63.641s |

[1] 2D pattern correspondence building.
[2] Dense correspondence building via ICP mehod.
[3] Deformation transfer phase.

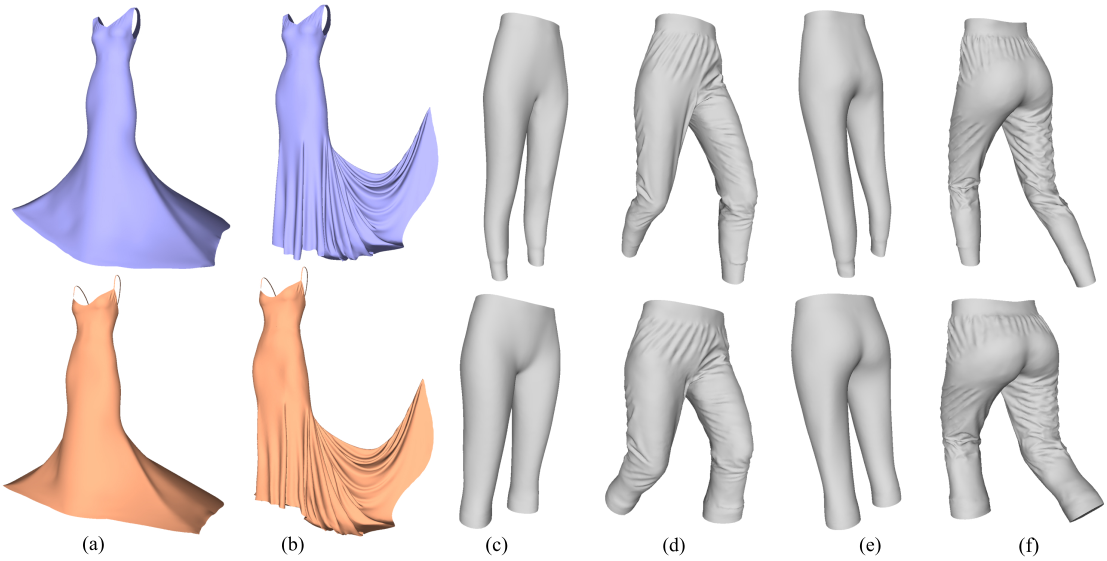


Figure 5: Pose and wrinkle transfer results of "gowns" and "women's casual trousers". For garment "gowns", we show the source models in purple and the target models in orange. Meanwhile, the two models in column (a) correspond to their smoothed reference source models, respectively. The purple model in column (b) (upper) represents the sculpted deformed source model, and the orange model in column (b) (bottom) represents the transfer result of our method. For garment "women's casual trousers", column (c) and (e) represent the smoothed reference source (upper) and target (bottom) models, and column (d) and (f) represent the sculpted deformed source model (upper) and our transfer results (bottom) from two perspective views.

In Figure 5, generated high-quality garment models with poses and detailed wrinkle information are semantically identical to the given source garment model. In addition, we also display our transfer results of four sets of garments with textures (See Figure 12 and Figure 13). Figure 13 illustrates that our method can deal with surfaces of different topologies. Results show that our algorithm can transfer pose and wrinkle characteristics from the source garment to the target faithfully.

### *6.1. Ablation Studies*

We present smoothing preprocessing and automatically-built correspondence as the two ablation studies to validate the effectiveness of our algorithm.

*Smoothing Preprocessing.* As shown in Figure 6, without smoothing preprocessing, the effect of deformation transfer is severely affected by the original geometric details of the models, and the fine-grained details of $S_{sculpted}$ is challenging to reproduce.

*Automatically-built Correspondence.* As shown in Figure 7, our method can automatically generate semantically correct dense correspondences between the source and the target, and obtain high-quality result in one minute (See Table 1, the third row), which competes against the original deformation transfer method (Sumner and Popovic, 2004) that takes about 20 minutes for manually selecting 50 pairs of marker points on $\tilde{S}$ and $\tilde{T}$. Moreover, it can release designers from making mistakes for manual operations.

### *6.2. Comparisons to Other Methods*

*Comparison to (Berkiten et al., 2017).* We compare our method to the algorithm proposed in (Berkiten et al., 2017), which learns the relationships between the geometric features and the displacement map (expressed as a texture) by deploying metric learning on the source model instead of establishing dense correspondences between the source and target meshes. As illustrated in Figure 8, the method of Berkiten *et al.* fails to transfer most of the wrinkle details of the sculpted source model compared to our method. In addition, our method can transfer the poses of $S_{sculpted}$ to the target garments, which cannot be represented by their displacement map.

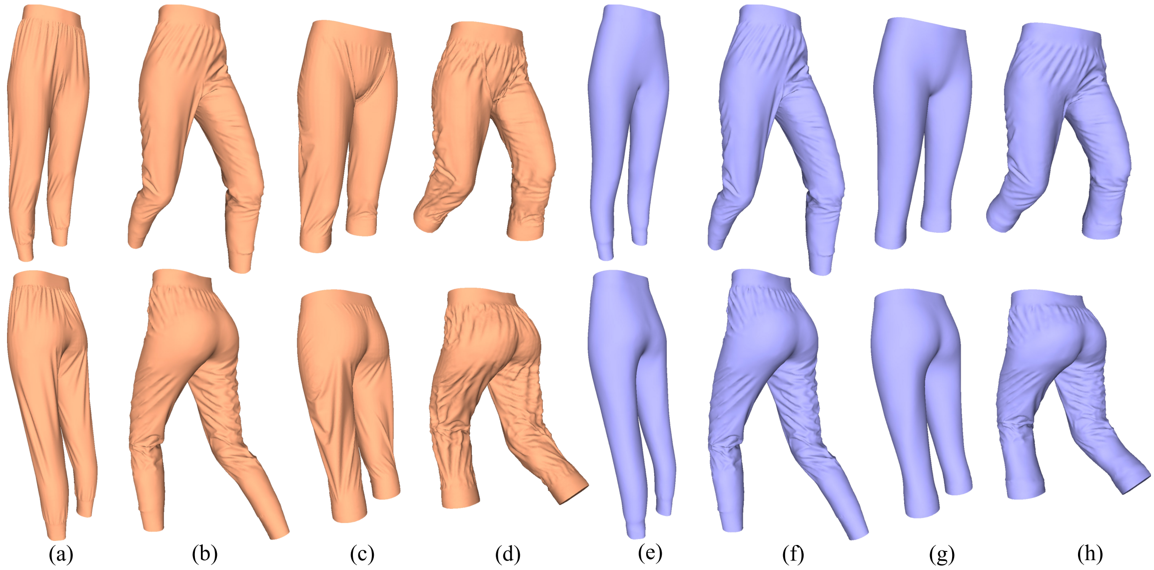


Figure 6: Ablation study for smoothing preprocessing. We use two rows to render the same models from two different perspective views. The trousers shown in orange in the left half are the result without smoothing processing, where (a) is the original simulated source model $S$, (b) is the same fine-grained sculpting result $S_sculpted$ as (f), (c) is the original simulated target model $T$, and (d) is the result without smoothing preprocessing. The trousers shown in purple in the right half are the result of our method with smoothing preprocessing, where (e) is the reference source model $\tilde{S}$, (f) is the fine-grained sculpting result $S_sculpted$ of $\tilde{S}$, (g) is the reference target model $\tilde{T}$, and (h) is the final result. As shown in (d), although the pose is successfully transferred to the target model, it can be observed that the wrinkles details are not smooth and lack aesthetics.

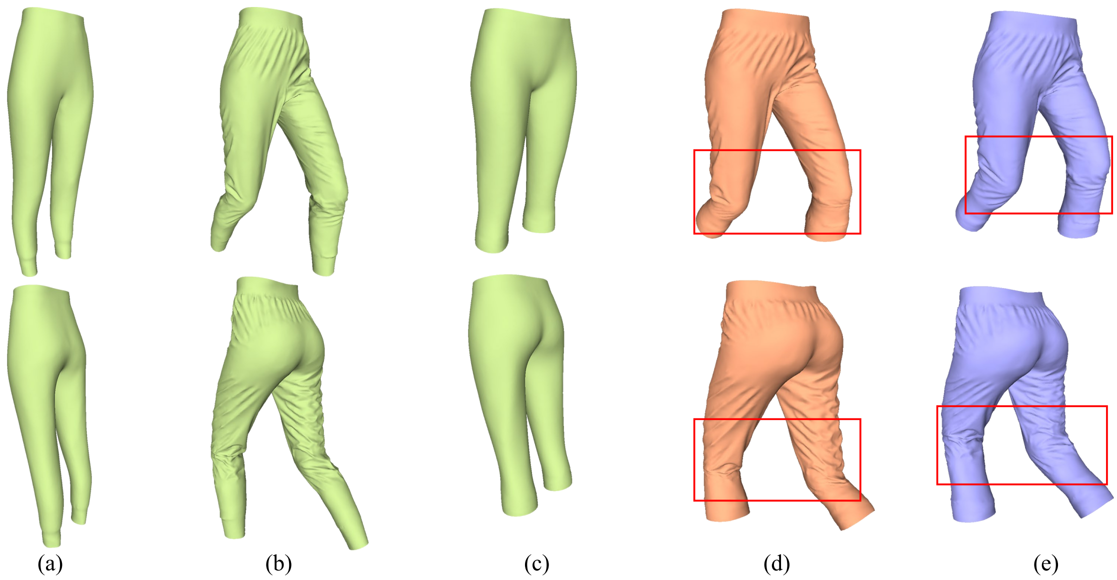


Figure 7: Ablation study for the correspondence establishment. The figure consists of two rows, each row shows the same model from two different perspective views. The women's casual trousers shown in green are the reference source $\tilde{S}$ (a), the sculpted source $S_sculpted$ (b), and the reference target $\tilde{T}$ (c). (d) is the result of deformation transfer (Sumner and Popovic, 2004) whose correspondence is established manually. (e) is the result of our presented framework whose correspondence is established automatically based on the semantic information of patterns.


Figure 8: Comparison to the algorithm trousers in (Berkiten et al., 2017). We use two sets of casual trousers models with similar styles for testing. As shown in the figure, the models shown in gray are the reference source (a), the sculpted source (b), and the reference target (c), respectively. The result generated by the method of (Berkiten et al., 2017) is shown in orange and the result produced by our algorithm is shown in blue.

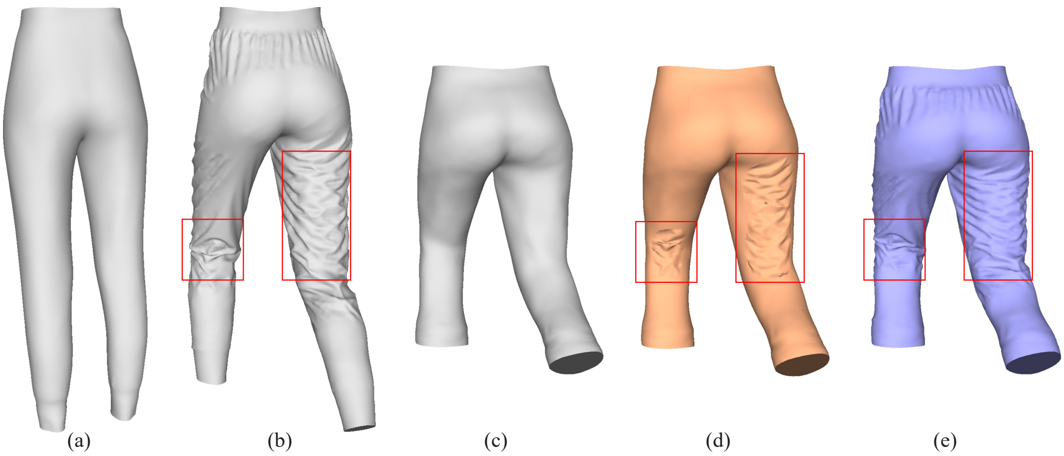


Figure 9: Comparisons to the method of (Takayama et al., 2011). The input of our algorithm is the smoothed reference source (a), the deformed source with a specific pose and fine wrinkle details (b), and the reference target with a similar style (c). The input of (Takayama et al., 2011) is the deformed source (b) and the reference target (c). The result of (Takayama et al., 2011) is shown in orange (d), and the result of our algorithm is shown in purple (e). The red boxes in (b) represent the ROI, and the red boxes in (d) and (e) represent the transfer results of GeoBrush (Takayama et al., 2011) and our algorithm, respectively.

*Comparison to ([Takayama et al., 2011](#)).* Due to the limitations of the canvas and ROI selection of the method in (Takayama et al., 2011), we can only select one ROI on the deformed source model (Figure 9 (b)) each time to clone the details in this area onto the specified area of the target model. As a result, wrinkle details of the source deformed model cannot be continuously cloned onto the target (Figure 9 (c)). For the above reasons, we select three regions on the trouser legs of the deformed source as ROI, and clone the details of these regions onto the target reference in three times respectively. As shown in Figure 9, the red box in (b) represents the ROI, and the red boxes in (d) and (e) represent the transfer results of GeoBrush (Takayama et al., 2011) and our algorithm, respectively. Although GeoBrush can maintain the features of the details of the cloned ROI to a large extent, there is still a certain degree of distortion and discontinuity for wrinkles between two regions.

*Comparisons to (Sorkine et al., 2004).* When the source and the target models have different topologies, Sorkine et al. (2004) find a dense mapping between their parametric domains using a few manually marked feature points for initial correspondences to perform the coating transfer. Since coating transfer cannot handle large shape changes, we first transfer the pose information from the source to the target model using deformation transfer (Sumner and Popovic (2004)). As shown in Figure 10, the coating information comes from the deformed source model and its filtered model without surface details (Figure 10 (c) and (b)), and we transfer the coating information onto the target model (Figure 10 (d)) using the dense mapping between the parametric domains of the filtered source model (Figure 10 (b)) and the target model (Figure 10 (d)). It can be seen from the figure that the result of coating transfer cannot preserve the wrinkle details (Figure 10 (e)) and has visual interpenetration artifacts between triangles, while our method can generate plausible results with aesthetic appearance. This may due to the fact that the parametric domains of garments have large distortions and are independent of each other. The dense mapping found in the parametric domains using sparse correspondences cannot encode the strong semantic correspondences demanded in our method.

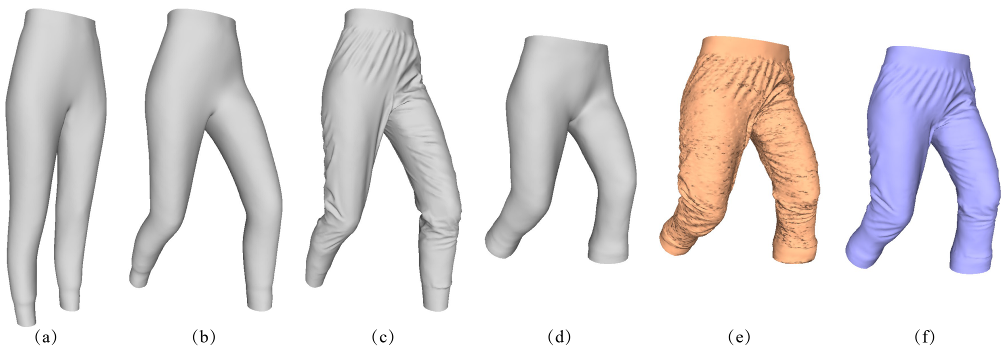


Figure 10: Comparisons to the method of (Sorkine et al., 2004). The input to coating transfer is the deformed source model (c) along with its filtered model without surface details (b), and the reference target (d). The input to our algorithm is the reference source (a), the deformed source (c) and the reference target (d). The coating transfer result is shown in orange (e), and our result is shown in purple (f). Comparisons show that our method can generate better results.

### *6.3. User Study*

We present five types of garments to 21 participants in sequence, where the manually sculpted source model and the automatically generated target model via our method in each type are randomly displayed for similarity comparison. A five-point scale is adopted, where 5 stands for the most similar score, and 1 represents the most dissimilar score. Table 2 shows the score distributions from the 21 participants across the five types of garments. Most of the scores are greater than 3, which validates the effectiveness of our pose and wrinkle transfer method.

## 7. Conclusions and Limitations

We have presented an automatic method to transfer pose and wrinkle details of the reference garment models to target models with the same or similar styles faithfully. Our key contribution is establishing the

Table 2: Distribution of similarity comparison score.

| | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|
| Gowns | 10 (47.62%) | 11 (52.38%) | 0 | 0 | 0 |
| Lantern sleeve dresses | 4 (19.05%) | 14 (66.67%) | 3 (14.29%) | 0 | 0 |
| Men's sports suits | 9 (42.86%) | 7 (33.33%) | 5 (23.81%) | 0 | 0 |
| Qipao and polo dress | 9 (42.86%) | 7 (33.33%) | 3 (14.29%) | 2 (9.52%) | 0 |
| Women's casual trousers | 9 (42.86%) | 9 (42.96%) | 3 (14.29%) | 0 | 0 |

dense correspondences between source and target models via the latent semantic information of 2D patterns, which releases the time and expertise requirements for manually selecting marker points. Experiments validate that our method can perform garment mesh deformation transfer efficiently and generate garment models for online display aesthetically.

Our method has some limitations. First, when the source and the target garment models have significantly different shapes, our method may fail because of their obvious differences of patterns. This limitation can be easily eliminated by augmenting the sculpted garment database. Second, our method does not deal with the collisions induced by the deformation transfer explicitly. As a result, self-penetration artifacts may arise in some garment models with complex rich wrinkle details (e.g., polo dress shown in Figure 11 (a), and lantern sleeve dress shown in Figure 11 (b)). Moreover, as we do not consider the underlying mannequin during deformation transfer, penetrations between the garment and the mannequin may arise, as shown in Figure 11 (c). We can eliminate both the self-penetration artifacts and penetrations between the garment and the mannequin by adding collision handling (Wu et al., 2018) at the cost of wrinkle detail changes to a certain extent. Third, for different trouser sizes, our automatic approach may lead to transfer position deviations, as shown in Figure 7 (e) marked with red boxes. In such scenarios, semantic information such as joints should also be taken into consideration.

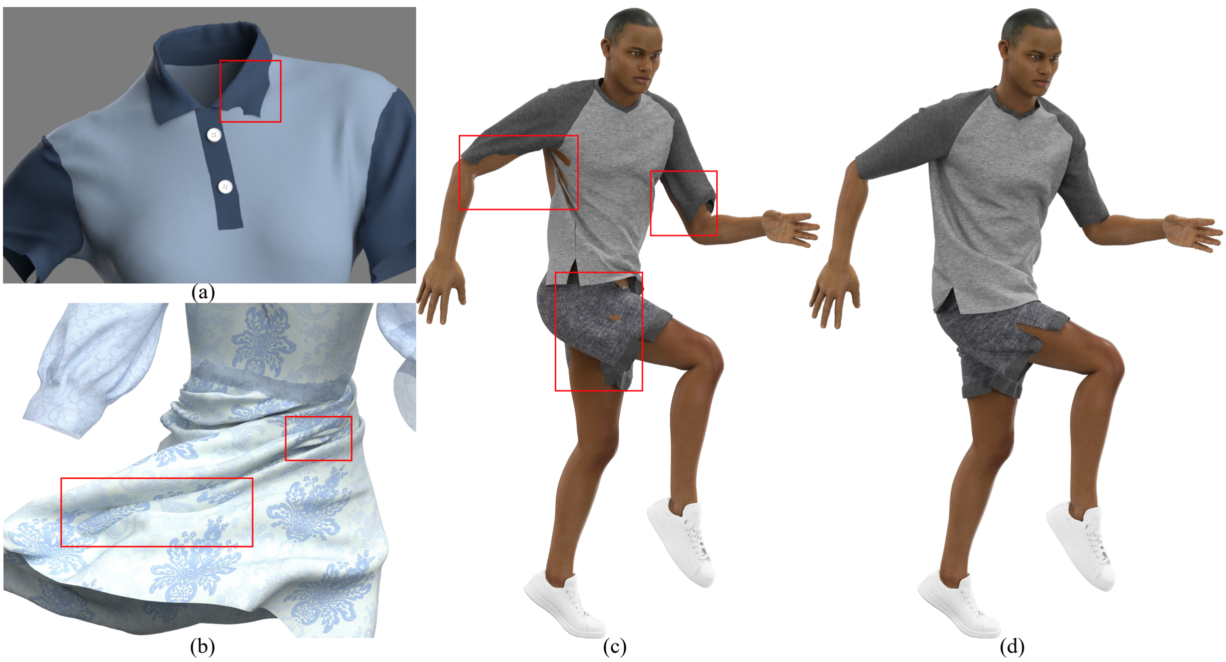


Figure 11: Penetration artifacts. Self-penetration artifacts (marked with red boxes) may arise for the resulting garments, such as the polo dress (a) and lantern sleeve dress (b). Penetrations may also exist (marked with red boxes) between the garment and the mannequin when we dress the transferred garment back to the mannequin (c), and such artifacts can be eliminated by collision handing methods (such as (Wu et al., 2018)) to generate plausible results (d).


Figure 12: Transfer results rendered with textures. From top to bottom in each row: Qipao and polo dress, lantern sleeve dress, men's sports suits, top garment of men's sports suits, and bottom of men's sports suits. For men's sports suits, we show the top and bottom garment separately in two rows. The three columns on the left represent source garments, and the three columns on the right represent target garments. In columns (a) and (d), we show the reference model of each group of garments. In columns (b), (c) and (e), (f), we show the same models from two different perspective views. Columns (b) and (c) are the manually-sculpted deformed garments, and columns (e) and (f) are the outputs of our method.

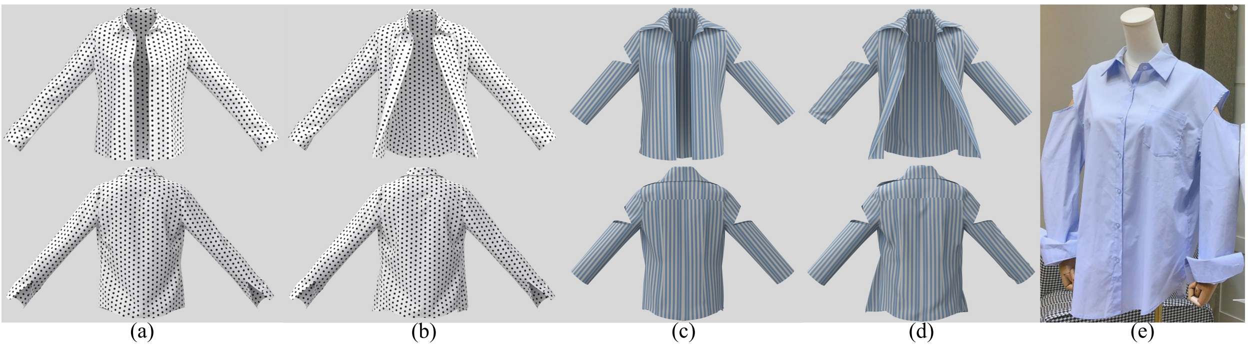


Figure 13: Transfer results between surfaces of different topologies. As shown in the figure, we showcase each model from two perspective views (front and back). Column (a) is the reference source model, column (b) is the deformed source model, column (c) is the reference target model, and column (d) is our transfer result. Compared to the source models (a, b), the target models (c, d) have two holes in the sleeves. Column (e) shows a real women's shirt with notched sleeves, which inspires the design of the target model (c, d).

## References


de Aguiar, E., Sigal, L., Treuille, A., Hodgins, J.K., 2010. Stable spaces for real-time clothing. ACM Trans. Graph. 29, article 106. https://doi.org/10.1145/1778765.1778843.

Baran, I., Vlasic, D., Grinspun, E., Popovic, J., 2009. Semantic deformation transfer. ACM Trans. Graph. 28, article 36. https://doi.org/10.1145/1531326.1531342.

Berkiten, S., Halber, M., Solomon, J., Ma, C., Li, H., Rusinkiewicz, S., 2017. Learning detail transfer based on geometric features. Comput. Graph. Forum 36, 361–373. https://doi.org/10.1111/cgf.13132.

Biasotti, S., Cerri, A., Bronstein, A.M., Bronstein, M.M., 2016. Recent trends, applications, and perspectives in 3d shape similarity assessment. Comput. Graph. Forum 35, 87–119. https://doi.org/10.1111/cgf.12734.

Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M., 2014. Projective dynamics: Fusing constraint projections for fast simulation. ACM Trans. Graph. 33, article 154. https://doi.org/10.1145/2601097.2601116.

Browzwear, 2000-2020. Browzwear solutions pte ltd. Website. browzwear.com.

CLO, 2020. Clo virtual fashion llc. Website. clo3d.com.

Gao, L., Yang, J., Qiao, Y.L., Lai, Y.K., Rosin, P.L., Xu, W., Xia, S., 2018. Automatic unpaired shape deformation transfer. ACM Trans. Graph. 37, article 237. https://doi.org/10.1145/3272127.3275028.

Gillette, R., Peters, C., Vining, N., Edwards, E., Sheffer, A., 2015. Real-time dynamic wrinkling of coarse animated cloth, in: Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Association for Computing Machinery, New York, NY, USA. p. 17–26. https://doi.org/10.1145/2786784.2786789.

Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., Grinspun, E., 2007. Efficient simulation of inextensible cloth. ACM Trans. Graph. 26, article 49. https://doi.org/10.1145/1276377.1276438.

Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J., 2012. Drape: Dressing any person. ACM Trans. Graph. 31, article 35. https://doi.org/10.1145/2185520.2185531.

Huang, Q., Adams, B., Wicke, M., Guibas, L.J., 2008. Non-rigid registration under isometric deformations. Comput. Graph. Forum 27, 1449–1457. https://doi.org/10.1111/j.1467-8659.2008.01285.x.

Jiang, C., Gast, T., Teran, J., 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. ACM Trans. Graph. 36, article 152. https://doi.org/10.1145/3072959.3073623.

Jubran, I., Maalouf, A., Kimmel, R., Feldman, D., 2021. Provably Approximated ICP. arXiv e-prints arXiv:2101.03588. https://arxiv.org/abs/2101.03588.

Jung, A., Hahmann, S., Rohmer, D., Bégault, A., Boissieux, L., Cani, M., 2015. Sketching folds: Developable surfaces from non-planar silhouettes. ACM Trans. Graph. 34, article 155. https://doi.org/10.1145/2749458.

van Kaick, O., Zhang, H., Hamarneh, G., Cohen-Or, D., 2011. A survey on shape correspondence. Comput. Graph. Forum 30, 1681–1707. https://doi.org/10.1111/j.1467-8659.2011.01884.x.

Kavan, L., Gerszewski, D., Bargteil, A.W., Sloan, P.J., 2011. Physics-inspired upsampling for cloth simulation in games. ACM Trans. Graph. 30, article 93. https://doi.org/10.1145/2010324.1964988.

Kezurer, I., Kovalsky, S.Z., Basri, R., Lipman, Y., 2015. Tight relaxation of quadratic matching. Comput. Graph. Forum 34, 115–128. https://doi.org/10.1111/cgf.12701.

Kim, D., Koh, W., Narain, R., Fatahalian, K., Treuille, A., O'Brien, J.F., 2013. Near-exhaustive precomputation of secondary cloth effects. ACM Trans. Graph. 32, article 87. https://doi.org/10.1145/2461912.2462020.

Kovnatsky, A., Bronstein, M.M., Bronstein, A.M., Glashoff, K., Kimmel, R., 2013. Coupled quasi-harmonic bases. Comput. Graph. Forum 32, 439–448. https://doi.org/10.1111/cgf.12064.

Lähner, Z., Cremers, D., Tung, T., 2018. Deepwrinkles: Accurate and realistic clothing modeling, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV, Springer. pp. 698–715. https://doi.org/10.1007/978-3-030-01225-0_41.

Li, C., Pan, H., Liu, Y., Tong, X., Sheffer, A., Wang, W., 2017. Bendsketch: modeling freeform surfaces through 2d sketching. ACM Trans. Graph. 36, article 125. https://doi.org/10.1145/3072959.3073632.

Li, M., Sheffer, A., Grinspun, E., Vining, N., 2018. Foldsketch: enriching garments with physically reproducible folds. ACM Trans. Graph. 37, article 133. https://doi.org/10.1145/3197517.3201310.

Maron, H., Dym, N., Kezurer, I., Kovalsky, S.Z., Lipman, Y., 2016. Point registration via efficient convex relaxation. ACM Trans. Graph. 35, article 73. https://doi.org/10.1145/2897824.2925913.

Müller, M., 2008. Hierarchical position based dynamics, in: Faure, F., Teschner, M. (Eds.), Proceedings of the Fifth Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2008, Grenoble, France, 2008, Eurographics Association. pp. 1–10. https://doi.org/10.2312/PE/vriphys/vriphys08/001-010.

Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J., 2007. Position based dynamics. J. Vis. Commun. Image Represent. 18, 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005.

Optitex, 1988-2020. Optitex. Website. optitex.com.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.J., 2012. Functional maps: a flexible representation of maps between shapes. ACM Trans. Graph. 31, article 30. https://doi.org/10.1145/2185520.2185526.

Ovsjanikov, M., Mérigot, Q., Mémoli, F., Guibas, L.J., 2010. One point isometric matching with the heat kernel. Comput. Graph. Forum 29, 1555–1564. https://doi.org/10.1111/j.1467-8659.2010.01764.x.

Ren, J., Poulenard, A., Wonka, P., Ovsjanikov, M., 2018. Continuous and orientation-preserving correspondences via functional maps. ACM Trans. Graph. 37, article 248. https://doi.org/10.1145/3272127.3275040.

Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D., 2017. Partial functional correspondence. Comput. Graph. Forum 36, 222–236. https://doi.org/10.1111/cgf.12797.

Rohmer, D., Popa, T., Cani, M., Hahmann, S., Sheffer, A., 2010. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. ACM Trans. Graph. 29, article 157. https://doi.org/10.1145/1882261.1866183.

Selle, A., Su, J., Irving, G., Fedkiw, R., 2009. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. IEEE Trans. Vis. Comput. Graph. 15, 339–350. https://doi.org/10.1109/TVCG.2008.79.

Solomon, J., Peyré, G., Kim, V.G., Sra, S., 2016. Entropic metric alignment for correspondence problems. ACM Trans. Graph. 35, article 72. https://doi.org/10.1145/2897824.2925903.

Sorkine, O., 2005. Laplacian mesh processing, in: Chrysanthou, Y., Magnor, M.A. (Eds.), 26th Annual Conference of the European Association for Computer Graphics, Eurographics 2005 - State of the Art Reports, Dublin, Ireland, August 29 - September 2, 2005, Eurographics Association. pp. 53–70. https://doi.org/10.2312/egst.20051044.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H., 2004. Laplacian surface editing, in: Boissonnat, J., Alliez, P. (Eds.), Second Eurographics Symposium on Geometry Processing, Nice, France, July 8-10, 2004, Eurographics Association. pp. 175–184. https://doi.org/10.2312/SGP/SGP04/179-188.

Sumner, R.W., Popovic, J., 2004. Deformation transfer for triangle meshes. ACM Trans. Graph. 23, 399–405. https://doi.org/10.1145/1015706.1015736.

Takayama, K., Schmidt, R.M., Singh, K., Igarashi, T., Boubekeur, T., Sorkine, O., 2011. Geobrush: Interactive mesh geometry cloning. Comput. Graph. Forum 30, 613–622. https://doi.org/10.1111/j.1467-8659.2011.01883.x.

Tam, G.K.L., Cheng, Z., Lai, Y., Langbein, F.C., Liu, Y., Marshall, A.D., Martin, R.R., Sun, X., Rosin, P.L., 2013. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. IEEE Trans. Vis. Comput. Graph. 19, 1199–1217. https://doi.org/10.1109/TVCG.2012.310.

Tevs, A., Bokeloh, M., Wand, M., Schilling, A., Seidel, H., 2009. Isometric registration of ambiguous and partial data, in: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, IEEE Computer Society. pp. 1185–1192. https://doi.org/10.1109/CVPR.2009.5206775.

Trimesh, 2020. Python library for loading and using triangular meshes. Website. https://github.com/mikedh/trimesh.

Turquin, E., Wither, J., Boissieux, L., Cani, M., Hughes, J.F., 2007. A sketch-based interface for clothing virtual characters. IEEE Comput. Graph. 27, 72–81. https://doi.org/10.1109/MCG.2007.1.

Vollmer, J., Mencl, R., Müller, H., 1999. Improved laplacian smoothing of noisy surface meshes. Comput. Graph. Forum 18, 131–138. https://doi.org/10.1111/1467-8659.00334.

Wang, H., Hecht, F., Ramamoorthi, R., O'Brien, J.F., 2010. Example-based wrinkle synthesis for clothing animation. ACM Trans. Graph. 29, article 107. https://doi.org/10.1145/1778765.1778844.

Wang, T.Y., Ceylan, D., Popovic, J., Mitra, N.J., 2018. Learning a shared shape space for multimodal garment design. ACM Trans. Graph. 37, article 203. https://doi.org/10.1145/3272127.3275074.

Wu, N., Zhang, D., Deng, Z., Jin, X., 2018. Variational mannequin approximation using spheres and capsules. IEEE Access 6, 25921–25929. https://doi.org/10.1109/ACCESS.2018.2837013.

Xu, W., Umetani, N., Chao, Q., Mao, J., Jin, X., Tong, X., 2014. Sensitivity-optimized rigging for example-based real-time clothing synthesis. ACM Trans. Graph. 33, article 107. https://doi.org/10.1145/2601097.2601136.

ZBrush, 2020. Pixologic, inc. Website. pixologic.com.

Zhou, K., Xu, W., Tong, Y., Desbrun, M., 2010. Deformation transfer to multi-component objects. Comput. Graph. Forum 29, 319–325. https://doi.org/10.1111/j.1467-8659.2009.01601.x.