Data-driven Garment Pattern Estimation from 3D Geometries

Chihiro Goto and Nobuyuki Umetani

The University of Tokyo

Figure 1: *Estimation of garment patterns from a 3D geometry. The triangles in the input mesh (left) are classified using a neural network into different types of pattern e.g., front, back and sleeve. The flat pattern shapes (right) are computed by developing the three-dimensional mesh for each pattern type.*

Abstract

Three-dimensional scanning technology recently becomes widely available to the public. However, it is difficult to simulate clothing deformation from the scanned people because scanned data lacks information required for the clothing simulation. In this paper, we present a technique to estimate clothing patterns from a scanned person in cloth. Our technique uses image-based deep learning to estimate the type of pattern on the projected image. The key contribution is converting image-based inference into three-dimensional clothing pattern estimation. We evaluate our technique by applying our technique to an actual scan.

CCS Concepts

• ***Computing methodologies*** → ***Shape modeling;*** *Neural networks;*

1. Introduction

Scanning the three-dimensional geometries of people has become popular thanks to the development of photogrammetry technology, which does not require any special optical devices other than ordinary cameras. However, the scanned geometry is stationary and is difficult to naturally animate because the scan lacks the information required for clothing simulation such as segmentation between clothing and body, and clothing pattern information.

In this paper, we present a technique to reconstruct clothing pattern from a 3D geometry of a clothed person typically obtained from 3D scanning and subsequent mesh clean-up. Our technique leverages image-based deep learning to estimate the type of pattern on the projected images. Our key contribution is the technique to convert two-dimensional image-based inference into clothing pattern type estimation on the three-dimensional geometry. We present a way to combine several two-dimensional inferences that are taken from multiple viewpoints to produce a single estimation of the pattern type on three-dimensional geometry. We chose to estimate garment patterns from multi-view projected images rather than directly from 3D data because many highly successful deep learning classification models are based on image data. By flattening the estimated pattern on the three-dimensional geometry, shapes of two-dimensional patterns are computed. We evaluate our technique by estimating the clothing pattern from the actual scan.

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

DOI: 10.2312/egs.20211013

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

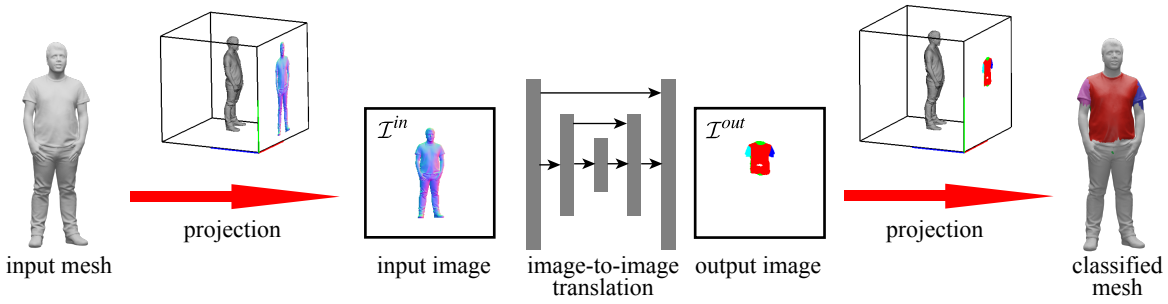

Figure 2: *Workflow of our method.*

2. Related Work

Garment reconstruction from scan Chen et al. [CZL*15] presents a technique to capture the three-dimensional shapes of clothing from depth scan. However, they do not estimate the pattern. So the captured clothing may not physically correct. Clothing geometry is estimated from the dynamic scan in [PMPHB17], however, the geometry lacks the pattern information. Raquel et al. [VSGC20] presented a learning-based approach to simulate garments with different mesh topology.

Pattern estimation from image Yang et al. [YPA*18] presented a framework to infer clothing patterns and their shapes. Wang et al. [WCPM18] predicts the patterns and their three-dimensional shapes from the user's sketch input. More recently Bhatnagar et al. [BTTPM19] presented a method to synthesize three-dimensional body and clothing shapes with texture from few frames of the video. However, these studies do not take a three-dimensional scan as an input.

Geometric flattening Geometric approaches are studied for developing three-dimensional surface to two-dimensional flat shape while optimized seam has been studied [PTH*17,SC18]. However, because there are multiple patterns to achieve a similar clothing shape, it is difficult to determine the patterns purely from the geometry. Instead, we estimate the shape of patterns leveraging the pre-existing way to construct patterns.

3. Method

The scanned triangle mesh typically does not have a consistent topology. The numbers of the vertices are different in each scan and there is no correspondence in their ordering. Thus, it is difficult to directly input the triangle mesh to the neural network. Hence, we render the triangle mesh into the image and predict the clothing pattern on the image. Figure 2 shows the overall workflow of our method.

3.1. Generation of Input and Output Images from a Mesh

View transformation Given an input geometry that is constructed from a set of triangles $\mathcal{T}$, we render its normal map image from multiple viewpoints. Let $\mathcal{V}$ the set of the viewpoints where the A_v, $v \in \mathcal{V}$ stands for the Affine transformation that maps the global coordinates to the local camera coordinates.

In this paper, the projection is orthogonal and we project the geometry into 2mx2m image plane with the resolution of 256x256 pixels. To make the input mesh fit into the view, we compute the bounding box of the input mesh and translate the center of the bounding box to the origin.

Input image The neural network takes normal map image (i.e. *input image* $\mathcal{I}^{in}$) as the input. The input image has three channels storing values of the surface normal computed in the camera coordinate $\mathbf{A}_v\mathbf{n}_t$ where the $\mathbf{n}_t \in \mathbb{R}^3$ is the normal of the triangle $t \in \mathcal{T}$. As for the background, we set the value as $(1,1,1)^T$. Because the normal map does not take this value, the neural network can easily differentiate the background and foreground. The normal map can be efficiently computed by rendering the mesh with a programmable shader. Specifically, we used GLSL in the OpenGL graphics library.

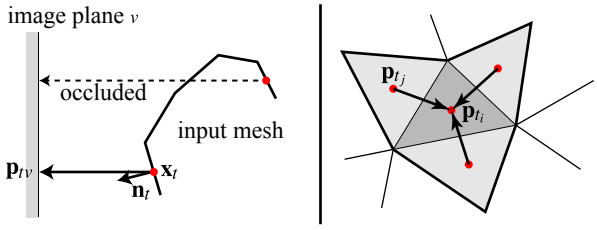

Figure 3: *Left: projection of the probability. Right: relaxation of the probability on the triangle mesh.*

Output image The input image is feed into a neural network that produce *output image* $\mathcal{I}^{out}$. Each pixel of the output image has the probability of the types of patterns (e.g., background, front, back, right sleeve, and left sleeve) of the corresponding triangle showing in the pixel. The output image has $M+1$ channels where the M is the number of the panels. The additional channel stores the probability that the pixel does not show the garment (i.e., showing body or background). Because each channel stands for the probability, the sum of them becomes one $\sum_{m=1}^{M+1} p_m = 1$, which is typically achieved by the *softmax* operation at the output layer of the neural network.

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

3.2. 3D Clothing Pattern Type Segmentation

Projection of probability Given the set of output images $\mathcal{I}_v^{out}$ from multiple different viewpoints and their corresponding Affine transformation A_v, we compute the pattern classification on the triangle mesh. As shown in Figure 3, given a triangle t in the mesh, we first compute the center of the gravity of the triangle $\mathbf{x}_t \in \mathbb{R}^3$. With the Affine transformation of the center position $\mathbf{A}_v\mathbf{x}_t$, we find the corresponding pixel in the image by looking at the horizontal and vertical coordinates on the camera. Let us denote the probabilities of the pixel for the triangle t and viewpoint v as $\mathbf{p}_{tv} \in \mathbb{R}^{M+1}$. We first compute the weighted sum of the probabilities

$$\mathbf{p}_t = \sum_{v\in\mathcal{V}} w_{tv}\mathbf{p}_{tv}, \qquad w_t = \sum_{v\in\mathcal{V}} w_{tv}. \tag{1}$$

Here, the weight $w_{tv} \in \mathbb{R}$ takes zero if the triangle is occluded and not shown in the pixel. The weight is scaled by the inner product of surface normal and the projection direction in the camera coordinate $\mathbf{e}_{depth}$ to prioritize the triangle facing the image

$$w_{tv} = \begin{cases} 0 & \text{if occluded} \\ (\mathbf{A}_v\mathbf{n}_t)\cdot\mathbf{e}_{depth} & \text{if not occluded.} \end{cases} \tag{2}$$

Note that the weight is equal or greater than zero $w_{tv} \geq 0$ as the non-occluded triangle is always facing the projection direction to some extent.

We check the occlusion efficiently without ray-triangle intersection computation on the CPU. When we render the normal map image, we obtain the depth image as the by-product of the hidden surface removal using the Z-buffer method. For a triangle t, if the depth of the pixel and the depth computed from the Affine transformation $\mathbf{A}_v\mathbf{x}_t$ are within a certain threshold, we conclude that the center of the triangle is not occluded.

Relaxation Due to the occlusion, certain triangles do not get much inference via projected probability i.e, the w_t is small. Hence, we diffuse the probability by computing the average among the neighboring three triangles to relax the triangle-wise prediction

$$\mathbf{p}_{t_i} = (1-\alpha)\mathbf{p}_{t_i} + \alpha \sum_{t_j\in Neighbor(t_i)} w_{t_j}\mathbf{p}_{t_j} \tag{3}$$

$$w_{t_i} = (1-\alpha)w_{t_i} + \alpha \sum_{t_j\in Neighbor(t_i)} w_{t_j}. \tag{4}$$

Here, we use the relaxation coefficient $\alpha = 0.2$ and we iterate this relaxation 20 times. Note that the number of the iteration is changed according to the mesh resolution – finer mesh requires more iteration. Finally, for each triangle, we set the pattern type that has the largest probability.

3.3. Developing Pattern

After we estimate the pattern type for each triangle in the mesh, we flatten the surface to compute the two-dimensional pattern shape. There are a lot of flattening approaches presented in the graphics community. Here, we used the approach based on the *exponential map* [SGW06] for the fast and robust mesh flattening.

As shown in Figure 4, for each pattern type, we first compute the triangle that is located at the center of the pattern. We find this center triangle by computing the maximum distance inward from the boundary using Dijkstra's algorithm [Dij59]. The boundary of the pattern is defined as the edges of triangles neighboring triangles with other types of patterns. Then, we compute the minimum distance for each triangle in the same pattern outward from the center triangle using again the Dijkstra's algorithm. For the Dijkstra's algorithm, we compute the distance between neighboring triangles as the geodesic distance between the centers of the triangles. Using the exponential map approach, we propagate the two dimensional coordinate from the center triangle by computing the weighted average of the two-dimensional coordinates of the neighboring triangles with smaller geodesic distance.

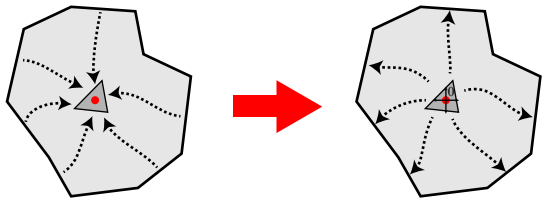

Figure 4: *Flattering classified triangle mesh. We first find out the center of the triangle mesh of the same pattern type by computing the maximum distance from the boundary using the Dijkstra's algorithm (left). Then, we compute two dimensional coordinate from the center*

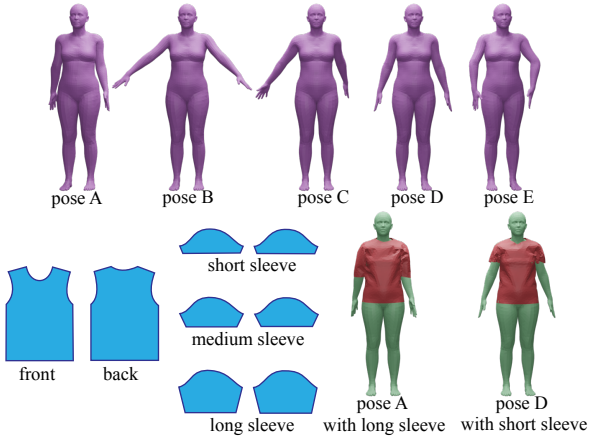

Figure 5: *Our training data is synthesized by simulating clothing on five different body shapes and three different clothing patterns.*

3.4. Training Dataset Generation

Training data for the neural network consists of a pair of input image $\mathcal{I}^{in}$ and the output image $\mathcal{I}^{out}$ We prepare many of these images via synthetic data generated via clothing simulation. We use SMPL model [LMR*15] to generate various body shapes with different poses. Then, we simulate clothing using our in-house clothing simulation code. We used the image-to-image translation network based on the U-net [RFB15] as the neural network architecture where the loss is the Cross entropy between the probabilities in the output image generated by the neural network and that of training data. Figure 5 shows an example of our training data. Each

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.

combination of body shape and clothing pattern produces 100 images with random scale, translation, and camera angle, which totally amount to 1,500 images. Then, we divided them into 1,400 images for training, 80 image for validation, and 20 images for test.

4. Results

Training In total, we prepared fifteen geometries of people in the cloth by simulating clothing on the combination of five different poses and three different clothing patterns (see Figure 5). For each geometry, we capture one hundred pairs of images for training. We trained the neural network for fifty epoch and it takes 1 hour for the training on the Google Colab. Figure 6 shows the convergence of the training of the neural network.

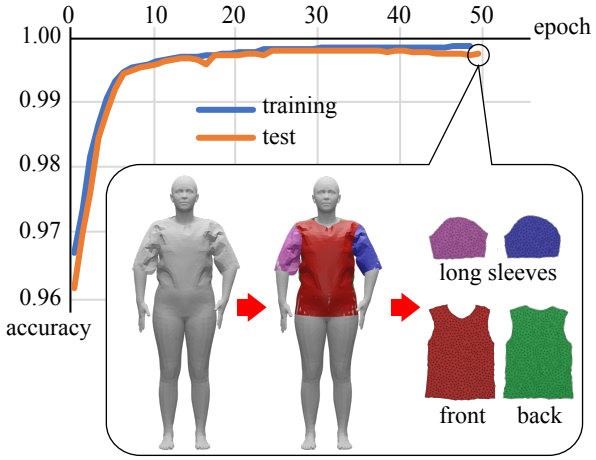

Figure 6: *Convergence of the accuracy in the network training. After training, the network estimates pattern shapes for the synthetic geometry ("pose E" with "long sleeve shirt" in Figure 5) from unseen view angles producing shapes similar to the original pattern.*

Figure 1 shows the input geometry and the predicted pattern for the three-dimensional model of a person wearing a T-shirt estimated from the four view points (front, back, right and left). Figure 7 shows the comparison between the pattern estimation result from two viewpoints (front and back) and the four viewpoints (front, back, right and left).

The estimation using the four viewpoints improved the quality of the pattern classification demonstrating that our projection and relaxation algorithm can successfully combine estimations from the multiple viewpoints. For each prediction, the image-based neural network takes 50 ms and other procedures including projection and flattening takes about 35 ms for the four viewpoint estimation (the time is measured with 13-inch Macbook Pro mid-2014 model).

5. Limitation and Future work

Our method predicts the pattern on the projected image, and hence it is difficult to predict the pattern type where it is heavily occluded. Another limitation is that the current flattening is not very accurate because the algorithm does not consider the stretch of the fabric. The estimation of the stretch from geometry might be possible through other neural network models but it is left as future work.

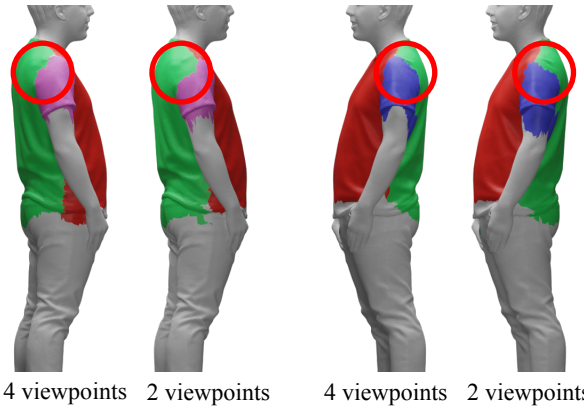

Figure 7: *Pattern estimation with different number of viewpoints.*

In the current pattern prediction, the boundaries of the pattern are jagged. For clothing simulation, another procedure to smoothing boundary is necessary. Enforcing the symmetry of the pattern in the prediction is important but it is left to be done as future work. So far, we only tested our method on the pattern of T-shirts. We are planning to test our algorithm for a wider variety of garments such as skirts and jackets.

References

[BTTPM19] BHATNAGAR B. L., TIWARI G., THEOBALT C., PONS-MOLL G.: Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)* (Oct 2019), IEEE. 2

[CZL*15] CHEN X., ZHOU B., LU F., WANG L., BI L., TAN P.: Garment modeling with a depth camera. 2

[Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische mathematik 1*, 1 (1959), 269–271. 3

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: Smpl: A skinned multi-person linear model. 3

[PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M. J.: Clothcap: Seamless 4d clothing capture and retargeting. 2

[PTH*17] PORANNE R., TARINI M., HUBER S., PANOZZO D., SORKINE-HORNUNG O.: Autocuts: Simultaneous distortion and cut optimization for uv mapping. 2

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241. 3

[SC18] SHARP N., CRANE K.: Variational surface cutting. 2

[SGW06] SCHMIDT R., GRIMM C., WYVILL B.: Interactive decal compositing with discrete exponential maps. 3

[VSGC20] VIDAURRE R., SANTESTEBAN I., GARCES E., CASAS D.: *Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On*. Eurographics Association, Goslar, DEU, 2020. 2

[WCPM18] WANG T. Y., CEYLAN D., POPOVIĆ J., MITRA N. J.: Learning a shared shape space for multimodal garment design. 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. 2

© 2021 The Author(s)
Eurographics Proceedings © 2021 The Eurographics Association.