FEATURE ARTICLE: Garment modelling, Sewing pattern design

PerfectTailor: Scale-Preserving 2D Pattern Adjustment Driven by 3D Garment Editing

Anran Qi and Takeo Igarashi, *The University of Tokyo, Tokyo, 113-0033, Japan*

Abstract—We address the problem of modifying a given well-designed 2D sewing pattern to accommodate garment edits in the 3D space. Existing methods usually adjust the sewing pattern by applying uniform flattening to the 3D garment. The problems are twofold: first, it ignores local scaling of the 2D sewing pattern such as shrinking ribs of cuffs; second, it does not respect the implicit design rules and conventions of the industry, such as the use of straight edges for simplicity and precision in sewing. To address those problems, we present a pattern adjustment method that considers the non-uniform local scaling of the 2D sewing pattern by utilizing the intrinsic scale matrix. In addition, we preserve the original boundary shape by an as-original-as-possible geometric constraint when desirable. We build a prototype with a set of commonly used alteration operations and showcase the capability of our method via a number of alteration examples throughout the paper.

Current garment alternation design is mostly centered around 2D sewing pattern space which involves numerous pattern editing operations to achieve the envisioned alterations of 3D garments. In practice, achieving the correct pattern adjustment not only necessitates specialized expertise in garment design but is also time-consuming. This is because designers need to justify both the envisioned 3D geometric changes of the garment and the embedded intrinsic design, such as smocking, elastic threading design, *etc*.

To speed up the design process and reduce the required expertise, researchers have proposed many powerful methods[1,2] to edit the garment directly in 3D space, and then automatically adjust the 2D pattern accordingly. Those methods usually assume the pre-existence of the sewing patterns. This matches the practice — designers often start with an existing pattern to create either real or virtual outfits. As a well-designed sewing pattern has already undergone design and development processes, it can significantly streamline the design process for designers, saving time and resources in comparison to developing a new pattern from scratch. However, the existing methods use uniform flattening (geometric surface parametrization) to update the sewing pattern after 3D editing. This is suboptimal for the following two reasons. Firstly, it is grounded in the hypothesis that the local sizes of both 2D and 3D triangles in the sewing pattern and 3D garment design remain mostly constant (no shrinkage or expansion). For certain secondary textile designs like smocking and elastic thread design, the 3D triangle on the garment shrinks or expands due to embedded stitching force during the simulation and draping, which makes the local sizes of 2D and 3D triangles non-uniform. Thus, uniformly flattening the 3D garment surface leads to misestimating the sewing panel size (see Figure 1 top row). Secondly, this technique usually pro-

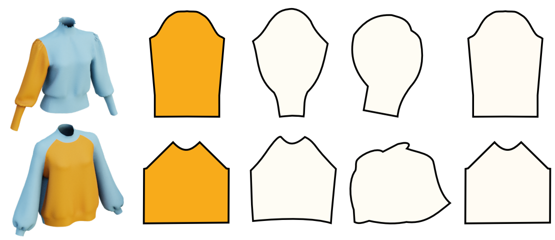


FIGURE 1. (a) Two 3D garments with hard yellow indicating the corresponding part to (b) the original panel designs. Panels flattened by (c) Sheffer *et al.*[11] (d) Igarashi *et al.*[4] and (e) our method. (c) and (d) generate smaller panels than the original panel (top row) and irregular boundaries (bottom row). In both cases, our method can produce the same panel shapes as that of the original panels.

Digital Object Identifier 10.1109/XXX.0000.0000000

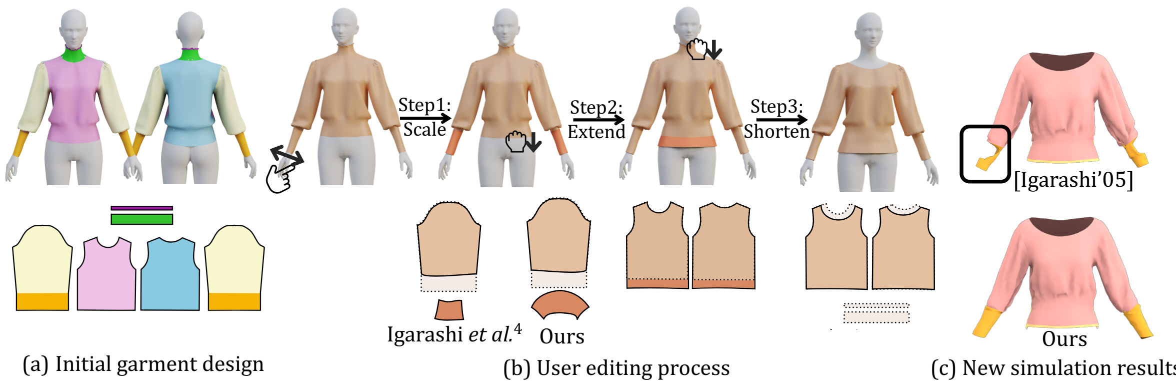


FIGURE 2. A user alteration example. (a) A 3D garment model from the front and back view (top row) and its corresponding sewing pattern (bottom row). The matching colour indicates the garment and sewing pattern correspondence. It is worth mentioning that the bottom of the sleeve (hard yellow) fits tighter than the upper part (soft yellow) due to its elastic string design, while their corresponding pattern has a similar width. (b) Our function sets allow the user to alter the garment directly in 3D space based on their preference and our pattern adjustment method updates the pattern accordingly. Top row: the 3D garment edits by the user; Bottom row: the pattern adjustment results (Dash lines illustrate the original panel shape). For Step 1: scale operation, we show the pattern adjustment results of Igarashi *et al.*[4] which is a naive geometric surface parametrization technique and ours. Igarashi *et al.*[4] generates a much smaller panel due to directly flattening the surface. Thus unlike us, it is not able to preserve the original embedded design. Our system incorporates the left-right symmetry in garment design literature, *i.e.,* the user only needs to edit on a single side of the garment and our system automatically mirrors those edits across the other side. (c) The garment is simulated by the pattern from Igarashi *et al.*[4] (top row) and ours (bottom row). The smaller panel produced by Igarashi *et al.*[4] causes the tearing of the sleeve highlighted in the black frame box.

duces panels with irregular boundaries. This neglects the implicit rules and conventions in the industry's design practices, such as straight edges, symmetric shapes and *etc.* (see Figure 1 bottom row).

To address this problem, we propose a pattern adjustment method specifically for synchronizing a well-designed 2D pattern according to the user's edits in 3D space. Our method considers the non-uniform local scaling of the 2D sewing pattern and respects the implicit rules and conventions of the industry. Specifically, our method memorizes the local intrinsic scale difference between the 2D pattern and the 3D drape of the initial design, which is used when updating the 2D pattern to compensate for the user's edit in 3D space. In addition, we propose an as-original-as-possible constraint to preserve the original panel boundary shape when the user modifies the entire panel. With our adjustment method, we develop a set of commonly-used editing operations to support garment alteration, such as scaling a specific part for fitting, extending and shortening the along the boundary, and cutting to achieve the desired shape (see Figure 2). We demonstrate alteration results on a number of garments, showing its usability and generalizability.

RELATED WORK

Garment Design

Various computational garment design techniques offer tools to automate the adjustment of underlying 2D patterns. These methods contribute to accelerating the design workflow and minimizing the need for extensive expertise in the field. Sensitive Couture[9] proposes a bidirectional interactive garment edit method by leveraging the fast simulation technique. It begins with well-designed sewing patterns and supports a subset of garment edits in 3D but is restricted to the dragging of vertices and edges.

Pietroni *et al.*[5] automatically generates the 2D sewing pattern from an input 3D shape by first creating the panel patch layout and then flattening the patch considering the anisotropic material property of the woven fabric. Wolff *et al.*[10] optimizes the 3D rest shape of the garment to maximise the fit and comfort under a range of poses and body shapes. The corresponding sewing pattern is generated by Sharp *et al.*[6] which directly optimizes the distortion in the cutting and flattening process. Liu *et al.*[17] adjusts the 2D sewing pattern by allowing the user to draw construction curves on the surface of the edited 3D garment model, then flatten the patches formed by the those curves. These

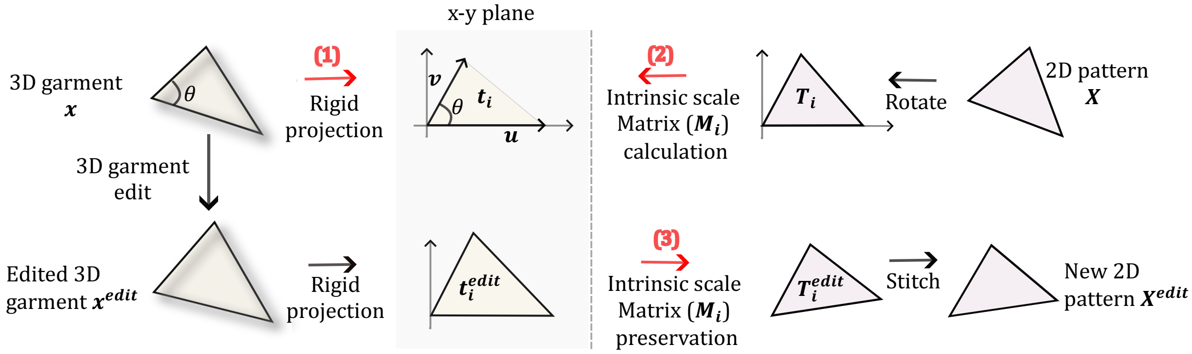


FIGURE 3. Computation flow of our flattening method. (1) (2) and (3) indicate the calculation of Equation 1, Equation 2 and Equation 3, respectively.

methods applies uniform flattening, thus cannot handle non-uniform local scaling of the sewing pattern and geometric constraints on boundary shapes.

Bartle *et al.*[1] proposes a fixed-point iteration method to optimize the pattern by minimizing the distortion between physical simulation results and the target draped garment so that it will generate the target garment geometry after simulation in the context of direct garment editing. The optimization accounts for pattern deformation caused by the physical forces during draping. However, physical simulation and optimization take time, requiring certain effort (*e.g.*, parameter tuning) to make it work. Our method bypasses physical simulation by taking a purely geometric approach, which is faster and more stable, but sacrifices physical correctness. Combining our method with theirs can be an interesting future work.

Surface Flattening

Surface flattening (*i.e.*, parametrization) methods take a surface with disk topology and aim to optimize its 2D mapping based on the defined distortion measurement. This is a fundamental and well-studied problem in computer graphics. We refer the reader to the survey[8] for a more complete background. Here we review a few works that are commonly used in the garment design literature for completeness. Those works can be roughly classified into two categories: geometry-based methods[4,11,12,13] and physics-based methods.[14,15]

Geometry-based methods formulate the surface flattening as a distortion minimization problem based on either vertex, edge, or face. Sheffer *et al.*[11] defines an angle preservation metric and a set of constraints on the angles to ensure the validity of the flat mesh. Then formulate it as a constrained minimization problem. As-rigid-as-possible methods use a local-global optimization approach to optimize an isometric distortion measurement defined based on a set of edges[12] or triangles.[4,13]

Physics-based methods use the elastic energy model to drive the deformation of each 2D triangle. McCartney *et al.*[14] proposes to use a relatively simple elastic model to calculate strain energy to measure the movement of vertices in each 2D triangle. Later, Wang *et al.*[15] converts the energy into force and utilizes it within the Lagrange equation to calculate the movement of each 2D vertex.

Different from the aforementioned works which flatten the 3D surface from scratch, we focus on flattening the surface with an initial condition to keep the original design intention.

METHOD

Our key observation is the uniform flattening of the garment surface will dissipate the embedded initial design in the pattern. To this end, we propose to memorize the local scale difference between the 2D pattern and the 3D drape of the initial design and use it when updating the 2D pattern to accommodate user editing on the 3D drape. Specifically, we transform the original 2D pattern X to new 2D pattern X^{edit} responding to user's edit from the original 3D garment x to new 3D garment x^{edit}. This adjustment process memories the local scale difference between X and x, and applies it to the local scale difference between X^{edit} and x^{edit} (see Figure 3).

Motivated by Bartle *et al.*,[1] we aim at modelling the initially embedded local scaling as an *intrinsic scale matrix* and keep the matrix during the updating process. Since this matrix is invariant to rigid transformations, we rigidly project a 3D triangle of the 3D garment x onto the 2D x–y plane, denoted as t_i. Then we rotate the corresponding 2D triangle of the 2D pattern X to align the t_i, denoted as T_i. So that a designated edge corresponded in t_i and T_i is aligned with the x-axis (see Figure 3). Now we can formulate each triangle as

$$\begin{pmatrix} |u| & |v|sin(\theta) \\ 0 & |v|cos(\theta) \end{pmatrix} \tag{1}$$

where u and v are edge vectors and θ is the angle between them. We model the intrinsic scale matrix as a 2D transformation M:

$$M_i = t_i T_i^{-1} \tag{2}$$

At each time, the user edits the 3D garment, we update the 3D garment geometry based on well-established geometric rules and get the edited 3D triangle s_i^{edit} with its 2D projection t_i^{edit}.

Recap that our goal is to keep the intrinsic scale matrix during the updating process. To do this, we seek to find the optimal 2D triangles T^{edit} that when multiplied with this matrix will produce the target triangles t^{edit}:

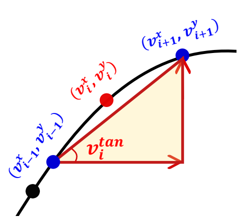
$$t_i^{edit} = M_i T_i^{edit} \tag{3}$$

Thus, we update the T_i^{edit} as

$$T_i^{edit} = M_i^{-1} t_i^{edit} = T_i t_i^{-1} t_i^{edit} \tag{4}$$

Stitching After the updating stage, the new triangle has the desired property. However, each triangle is processed independently, so it is unlikely that they will not form a continuous 2D mesh pattern. We, therefore, need to stitch all the triangles together to form a valid pattern S^{edit}. This is a well-studied mesh parametrization problem that can be solved using geometry-based surface flattening methods. We use the as-rigid-as-possible surface flattening technique[4] to get our final results.

After extensive deliberations with professional garment designers, we decided to preserve the original pattern shape, maintaining the integrity of the initial pattern when the user edit will affect the entire sewing panel equally. Therefore, we propose an as-original-as-possible constraint to preserve the discrete tangent of boundary vertices v_i^{tan} expressed as $\frac{v^x_{(i+1)} - v^x_{(i-1)}}{v^y_{(i+1)} - v^y_{(i-1)}}$ in the stitching process (inset).



We define the quadratic error function as

$$\begin{aligned} \arg\min_{v' \in V} & \sum_{(i,j)\in E} \|(v_j' - v_i') - (v_j - v_i)\|^2 \\ & + w_1 \sum_{i \in C} \|(v_i' - C_i)\|^2 + w_2 \sum_{i \in B} \|(v_i^{tan'} - B_i^{tan})\|^2 \end{aligned} \tag{5}$$

where v_i and $v_i^{'}$ are the vertex coordinate of the triangle in the original pattern (T_i) and the new targeted pattern (T_i^{edit}), respectively. E is a set of edges, C is a set of fixed vertices, C_i is the fixed vertex coordinate, B is the set of boundary vertices and B_i is the boundary vertex coordinate. We omit the vertices where $v^{tan'}$ is not defined. And fix the endpoints of an edge at the center of the original pattern and use $w_1 = w_2 = 1000$ for the examples in this paper.

SYSTEM OVERVIEW and IMPLEMENTATION

User Interface

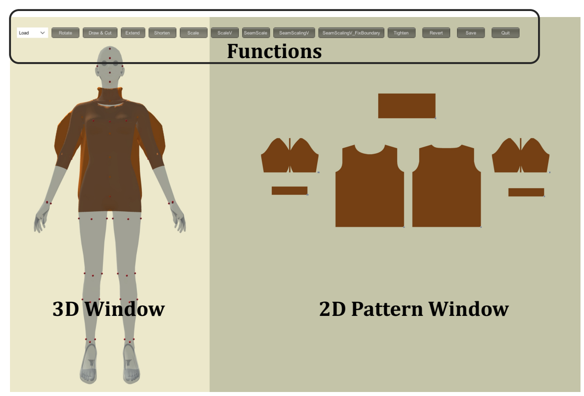


FIGURE 4. A screen snapshot of the system. The left (beige) is the 3D window displaying the 3D garment on a human body. Red dots indicate key feature points on the surface of the human body, *e.g.*, front neck point, and busty points. The right is the 2D pattern window showing the sewing pattern. The top shows the functions of the system.

The system has two windows: the 3D window and the 2D pattern window for displaying the 3D garment and 2D sewing pattern, respectively (see Figure 4). The user can edit the 3D garment using editing operations provided by the system in the 3D window. In the 2D pattern window, the user can freely move the panel to adjust the pattern's layout by clicking. The top shows the system's editing functions. Since interacting with these editing functions is straightforward and hence not described here. Please see the accompanying video for details. In the following section, we detail the supported operations and illustrate them with multiple examples. We developed our system on the Unity

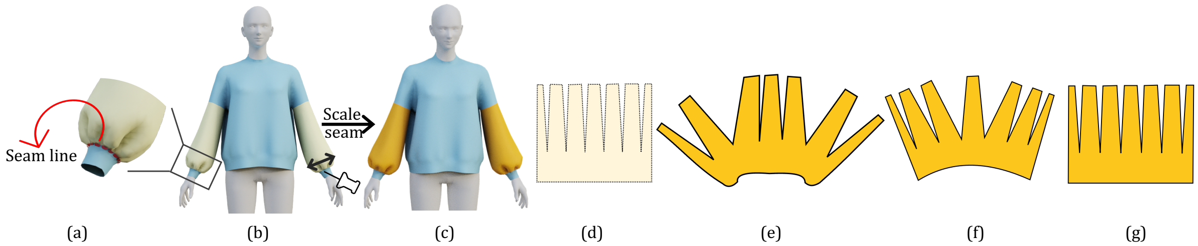


FIGURE 5. Scale the internal sleeve part by editing the seam line. (a) The red stitches visualize the seam line of the lantern sleeve. (b) The user edits the seam line to scale the internal sleeve (soft yellow) perpendicular to the human body whilst fixing the bottom part around the wrist. (c) The scaled internal sleeve (hard yellow). (d) The original sleeve panel. (e) The updated panel by Igarashi *et al.*[4] (f, g) The updated panel by our method without/with the as-original-as-possible constraint, respectively. Compared with (g), Igarashi *et al.*[4] (e) has a slightly larger pattern with the severely shrunk lower boundary. This is because it directly flattens the deformed 3D triangles, which are deformed due to stitching force, gravity, and *etc* in the simulation. (f) has the right size but its curved boundary damages the original design. With our as-original-as-possible constraint, (g) has the right size whilst maintaining the original panel's boundary design.

platform using C# and run our system on a desktop with Intel(R) Core(TM) i7-8700K 3.7GHz CPU.

Editing Operations

Starting with the input 3D garment and the corresponding pattern, we implement a set of simple and commonly used alteration operations that allow the user to explore the complex redesign space:

Scale: The user can select a part on the 3D garment and scale either along (see Figure 6 top row) or perpendicular (see Figure 6 bottom row) to the human skeleton direction by dragging. Furthermore, our system also supports the scaling of the internal garment part leveraging the seam line. In the garment literature, a seam line refers to the line or path created by joining two or more pieces of fabric together using stitches (see Figure 5 (a)). Our system utilizes the seam line to allow the user to scale the internal garment part either along (see Figure 5) or perpendicular (see Figure 7) to the human body. In such instances, after thorough discussions with the designers, we opt to update the whole sewing panel using our proposed flattening method with the as-original-as-possible constraint to preserve the original panel's boundary shape. The rationale behind this is the user edit will affect the entire sewing panel equally, thus it is optimal to maintain the original panel's boundary design as closely as possible.

Cut: We allow the user to sketch on the 3D garment to cut the garment into the desired shape. By default the user sketches from one viewpoint, and we will automatically form a closed loop as the cutting line on the 3D garment to cut both sides. We also enable the user to only cut one side if the user specifies. Then we re-triangulate the meshes affected by the new cutting line and update the sewing pattern by directly transferring the barycentric coordinate of the new vertex into the local coordinate system of the corresponding 2D triangle. For the detailed algorithm, we refer the reader to the Teddy system.[3]

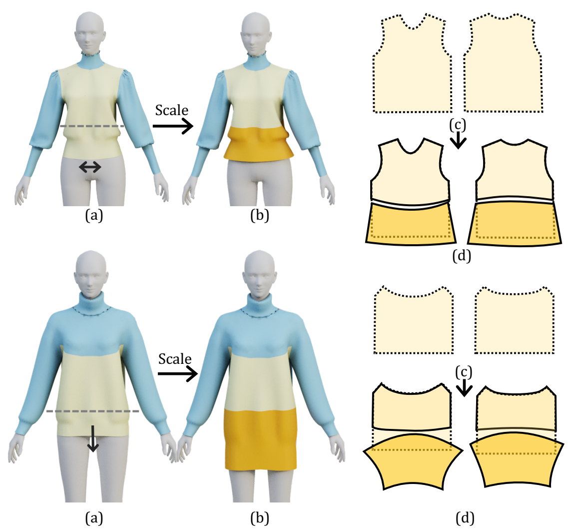


FIGURE 6. The user scales the bottom part of the garment perpendicular to the human skeleton making it looser (top row), along with the human skeleton making it longer (bottom row). (a) The original garment. Soft yellow indicates the panels affected by the user edits. (b) The updated garment geometry. Dark yellow indicates the parts selected by the user, being customised. (c) The original panels. (d) The updated panels (Dash lines illustrate the original panel shape).

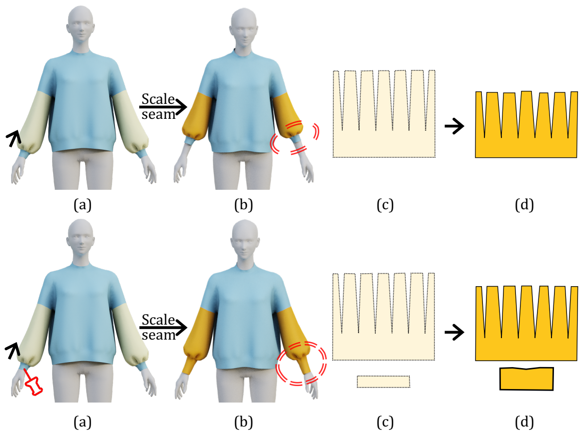


FIGURE 7. Scale the internal sleeve part along the body by editing the seam line. Top row: The user moves the seam line upwards making the internal sleeve shorter. Bottom row: The user moves the seam line upwards whilst fixing the lower boundary of the bottom sleeve. This leads to a shorter internal sleeve but a longer bottom sleeve. The red circle in (b) highlights the difference. (a) The original garment. Soft yellow indicates the panels affected by the user edits. (b) The updated garment geometry. Dark yellow indicates the parts selected by the user and being customised. (c) The original panels. (d) The updated panels.

Shorten: The user can drag the boundary to the desired position to shorten the garment (see Figure 2 (b) Step 3 and Figure 8). In detail, we compute the iso-line on the garment surface mesh where the distance to the boundary is the user-specified shortened distance. We take the computed iso-line as the cutting line on the 3D garment and utilize the Cut function to shorten the garment.

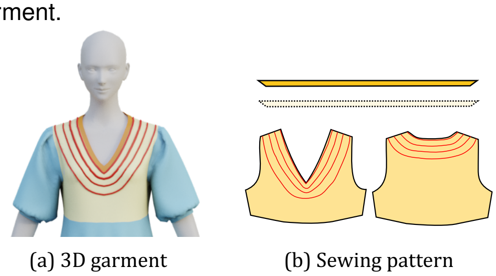


FIGURE 8. Shorten example. The user iteratively shortens the garment four times to explore various collar designs. Red curves on (a) and (b) indicate the cutting lines on the 3D garment and sewing pattern, respectively.

Extend: The user can also extend the garment by dragging the boundary to the desired position (see Figure 2 (b) Step 2 and Figure 9). Following the observation from Brouet *et al.*[2], people tend to preserve the slope and tangent plane orientations across the garment surface when transferring the garment. We follow the same principle by appending the triangle faces which share the same surface normal with the connected triangle faces. Body-garment collisions sometimes occur when extending. We resolve it by pushing the vertex towards the normal direction of the nearest triangle on the body surface. Finally, we update the sewing pattern in the same way as that of the Cut function.

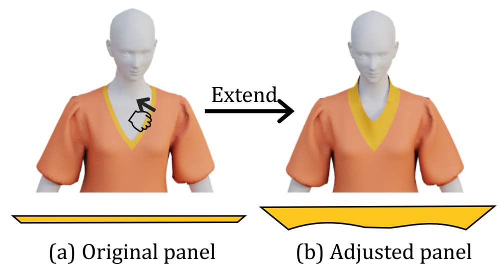


FIGURE 9. The user extends the V-shape neck collar.

F5 Tighten/Loosen: We also allow the user to tighten/loosen the garment by directly over-sketching the silhouette of the 3D garment. The system deforms the garment to meet the silhouette and updates the pattern with our proposed flattening method. For the detailed 3D deformation algorithm, we refer the reader to the Nealen *et al.*[16]

Those editing operations allow the user to alter garments and showcase the capabilities of our proposed pattern adjustment method, but more operations could be added to enhance the redesign capacity such as adding folds and darts.

CONCLUSION

We present a pattern adjustment method that aims to preserve the embedded design for garment alteration. Besides, we develop a set of editing operations to support alteration and showcase the capability of our adjustment method and functions via several examples throughout the paper. The editing operations introduced in this paper are not exhaustive, the the core algorithm, "scale-preserving flattening" is general and can be applied to other 3D modelling operations.

Limitations. In this project, we have not investigated the quality-efficiency trade-off between our method and physical simulation methods, such as Bartle *et al.*[9]. Generally speaking, we believe our method surpasses Bartle *et al.* in terms of efficiency due to the omission of physical simulation, though it may fall short in quality. A detailed quantitative analysis between these methods would be beneficial to the field. Additionally, we have not delved into the potential discrepancies between garments manufactured using sewing patterns generated by our method and the expectations of the user. We leave the exploration of such deviations through a user study as the future work. Another limitation is that our set of operations cannot compete with commercial software. More operations could be added to enhance the redesign capacity such as designing free-form surface deformation, adding folds and darts, and *etc.*

ACKNOWLEDGMENT

This work is supported by SHIMA SEIKI MFG., LTD. This work is also supported by JST AdCORP, Grant Number JPMJKB2302, Japan.

REFERENCES

1. A. Bartle, A. Sheffer, V. G. Kim, D. M. Kaufman, N. Vining, and F. Berthouzoz, " Physics-driven pattern adjustment for direct 3D garment editing," *ACM Transactions on Graphics (TOG).*, vol. 35, no. 4, pp. 50–1, 2016. (journal)
2. R. Brouet, A. Sheffer, L. Boissieux, and M. P. Cani, " Design preserving garment transfer," *ACM Transactions on Graphics (TOG).*, vol. 35, no. 4, pp. 1–11, 2012. (journal)
3. T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: a sketching interface for 3D freeform design," *ACM Transactions on Graphics (TOG).*, pp. 409—416, 1999. (journal)
4. T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Transactions on Graphics (TOG).*, vol. 24, no. 3, pp. 1134–1141, 2005. (journal)
5. N. Pietroni, C. Dumery, R. Falque, M. Liu, T. Vidal-Calleja, and O. Sorkine-Hornung, "Computational pattern making from 3D garment models," *ACM Transactions on Graphics (TOG).*, vol. 41, no. 4, pp. 1–14, 2022. (journal)
6. N. Sharp, and K. Crane, "Variational surface cutting," *ACM Transactions on Graphics (TOG).*, vol. 37, no. 4, pp. 1–13, 2018. (journal)
7. A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, "ABF++: fast and robust angle based flattening," *ACM Transactions on Graphics (TOG).*, vol. 24, no. 2, pp. 311–330, 2005. (journal)
8. A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," *Foundations and Trends® in Computer Graphics and Vision.*, vol. 24, no. 2, pp. 105–171, 2007. (journal)
9. N. Umetani, D. M. Kaufman, T. Igarashi, and E. Grinspun, "Sensitive couture for interactive garment modeling and editing," *ACM Transactions on Graphics (TOG).*, vol. 30, no. 4, pp. 1–12, 2011. (journal)
10. K. Wolff, P. Herholz, V. Ziegler, F. Link, N. Brügel, and O, Sorkine-Hornung, "Designing Personalized Garments with Body Movement," *Computer Graphics Forum.*, vol. 42, no. 1, pp. 180–194, 2023. (journal)
11. A. Sheffer, B. Lévy, M. Mogilnitsky, and A.Bogomyakov, "ABF++: fast and robust angle based flattening," *ACM Transactions on Graphics (TOG).*, vol. 24, no. 2, pp. 311–330, 2005. (journal)
12. O. Sorkine, and M.Alexa, "As-rigid-as-possible surface modeling," *Symposium on Geometry processing.* Vol. 4, pp. 109–116, 2007. (journal)
13. L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S.J. Gortler, "A local/global approach to mesh parameterization," *Computer graphics forum.*, Vol. 27, No. 5, pp. 1495–1504, 2008. (journal)
14. J. McCartney, B.K. Hinds, and B.L. Seow, "The flattening of triangulated surfaces incorporating darts and gussets," *Computer-Aided Design.*, Vol. 31, No. 4, pp. 249–260, 1999. (journal)
15. C.C. Wang, S.S. Smith, and M.M. Yuen, "Surface flattening based on energy model," *Computer-Aided Design.*, Vol. 34, No. 11, pp. 823–833, 2002. (journal)
16. A. Nealen, O. Sorkine, M. Alexa, and D. Cohen-Or, "A sketch-based interface for detail-preserving mesh editing," *ACM Transactions on Graphics.*, pp. 1142–1147, 2005. (journal)
17. K. Liu, X. Zeng, P. Bruniaux, X. Tao, X. Yao, V. Li, and Jianping Wang, "3D interactive garment pattern-making technology,“ *Computer-Aided Design.*, pp. 113–124, 2021. (journal)

Anran Qi is a Project Assistant Professor at the Graduate School of Information Science and Technology at The University of Tokyo. Her research focuses on garment design and human sketch understanding from graphics, machine learning, and human-computer interaction perspectives. She received a Ph.D from the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey in 2021. Contact her at annranqi@gmail.com.

Takeo Igarashi is a Professor of Computer Science Department at The University of Tokyo. His research interest is in user interfaces and interactive computer graphics in general. He received a Ph.D. from the Department of Information Engineering at The University of Tokyo in 2000. He received the SIGGRAPH Significant New Researcher Award, JSPS Award, CHI Academy Award, and the Asia Graphics 2020 Outstanding Technical Contributions Award. Contact him at takeo@acm.org.