

Pillow: Interactive Flattening of a 3D Model for Plush Toy Design

Yuki Igarashi¹ and Takeo Igarashi²

¹ The University of Tokyo, 4-6-1 Komaba, Meguro, Tokyo, 153-8904, Japan
yukim@acm.org

<http://www.den.rcast.u-tokyo.ac.jp/~yuki>

² The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, 113-0033, Japan

Abstract. Pillow is an interactive pattern-design system for creating plush toys. The user imports a 3D surface model into the system and specifies the segmentation boundaries interactively by drawing seam lines on the model surface. Based on this segmentation, the system generates a 2D pattern by flattening each outlined region, and then visualizes the shape of the resulting plush toy by applying a simple physics simulation. If the result is not satisfactory, the user can make different seam lines. This closed-loop framework allows users to experiment with various seam patterns before actually working on real fabric to obtain the best-looking result. The system also estimates total sewing time based on the total length of the seam lines. We implemented the system to produce plush toys and a balloon.

1 Introduction

To design an original plush toy, one needs to construct an appropriate 2D pattern which is very difficult. For this reason, all existing patterns have been developed by professional designers. To enable people to design their own original plush toys and similar products, we developed an interactive pattern-design system. Called Pillow, the system allows users to import a 3D model and interactively specify segmentation boundaries by drawing seam lines on the model surface. The system generates 2D patterns by flattening the user-defined segments and then simulates the shape of the plush toy. If the simulation is not satisfactory, the user can try another set of seam lines. This closed-loop interaction allows one to experiment with various seam line designs without actually sewing real fabric. The system makes it easier to construct 2D patterns, and allows users to design their own original plush toys and other products. The system also estimates sewing time based on the total length of seam lines. We produced some plush toys and a balloon using the system, and discuss the results below.

Mori and Igarashi [1] presented an interactive system that allows nonprofessional designers to develop their own original plush toys. The user interactively designs a 3D model by a sketching tool, and then the system generates a corresponding 2D pattern. However, the system is not useful when the user already has a 3D model. Julius et al. [2] presented an automatic segmentation algorithm

for plush toys and demonstrated its viability by creating real toys using the machine-generated pattern. However, their method relies on purely geometric criteria, and it is difficult to capture the perceptually important features of the original shape. As a result, seams can appear in undesirable areas, such as across the face of a toy, or in ways that ignore the symmetry of the model. Our approach focuses on the user’s artistic process of pattern design and provides tools to support the exploration process.

2 User Interface

Figure 2 shows a snapshot of the system. The left window shows the original 3D model, the center window shows the flattened 2D pattern, and the right window shows the reconstruction. Users import a 3D surface model (Fig. 1(a)), upon which they draw free-form lines to mark segmentation boundaries (Fig. 1(b)). These can be erased if necessary, Once all the segments are outlined, the system

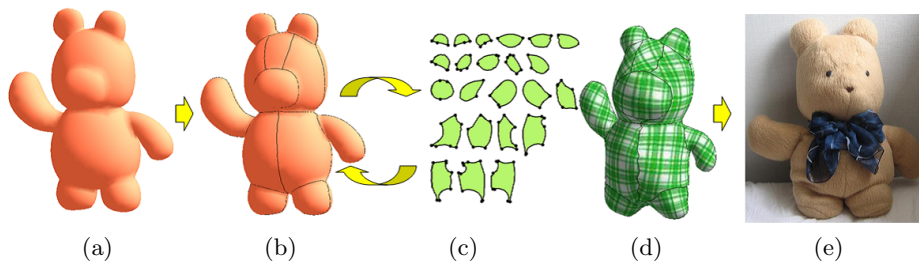


Fig. 1. Illustration of the Pillow system. (a) Input 3D model, (b) seam lines drawn by the user, (c) 2D sewing pattern generated by flattening the digitized regions, (d) reconstruction generated by applying a physics simulation to the pattern. The user repeats (b) through (d) until obtaining a satisfactory result. (e) The final product.

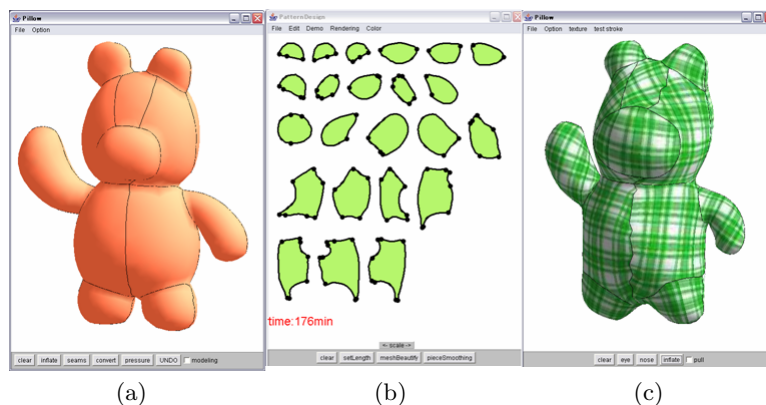


Fig. 2. A screen snapshot of the Pillow system

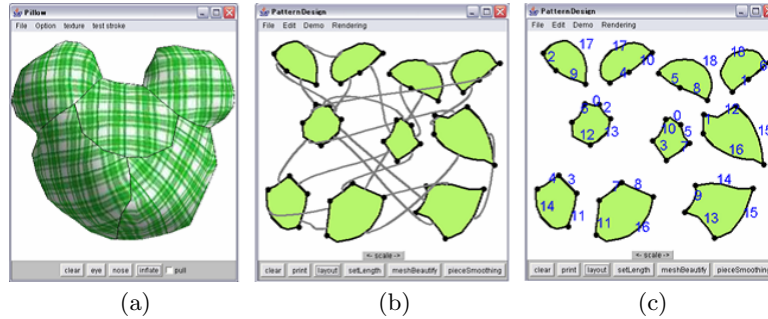


Fig. 3. Segments joined using connectors (b) and numbers (c)

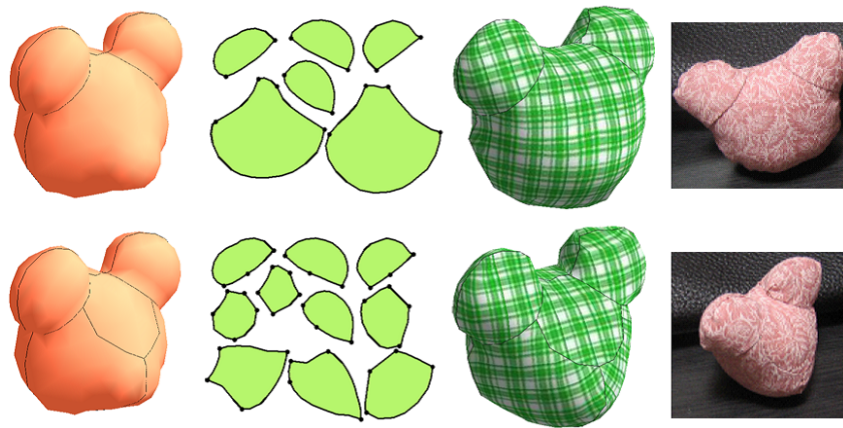


Fig. 4. Example of pattern designs

automatically flattens them (Fig. 1(c)) and reconstructs the 3D geometry by virtually sewing all of the flattened pieces together and applying a simple physics simulation (Fig. 1(d)). This view represents how the stuffed toy might look when a real toy is created using the current segmentation. If the user dislikes the result, he can quickly change the segmentation by erasing and redrawing the seam lines. The system displays how patches are joined by showing connectors or paired numbers (Fig. 3). Connectors are useful for understanding the relationship on the screen and numbers are useful as a printed reference on each patch. The system provides an automatic layout and manual arrangement interface for preparing the final pattern to be printed.

Figure 4 shows examples of pattern designs. Users can quickly experiment with various segmentation strategies using the system, and the simulation results successfully capture the overall shape of the final sewing results.

3 Estimation of Total Sewing Time

Total sewing time differs depending on the design as shown in Table 1, and it is very helpful to know how long it will take to create a plush toy during the design phase. To achieve this goal, our system automatically estimates total sewing time and presents this information to the user during seam design. It is assumed that the user will sew the pattern by hand.

Table 1. Geometric values of the 2D pattern

	Time(min)	L_{seam}	N_{seam}
Fig. 4(top)	40	2756.5	16
Fig. 4(bottom)	68	3706.3	38

3.1 Classification of Time Costs

Four main factors, or costs, determine the time required to create a plush toy: cutting a printed pattern (C_{cut}), tracing the pattern boundary on the cloth (C_{trace}), cutting the cloth (C_{cut_cloth}), and sewing the cloth (C_{sew}). The cost of stuffing is ignored because it is negligible compared to the sewing time. Therefore, the total cost (C_{total}) to create a real plush toy is defined as:

$$C_{total} = C_{cut} + C_{trace} + C_{cut_cloth} + C_{sew}. \quad (1)$$

The cost of cutting a pattern (C_{cut}) has already been explored by Mitani [3]. We approximate C_{trace} and C_{cut_cloth} by using C_{cut} because all operations essentially trace the same seam lines on the 2D pattern. The total cost is written as follows:

$$C_{total} = 3C_{cut} + C_{sew}. \quad (2)$$

Cutting Cost: Mitani [3] computed the cost of cutting a pattern as follows:

$$C_{cut} = W_{cut}L_{seam} + W_{cut_stoit}N_{cut_stoit}, \quad (3)$$

where L_{seam} is the length of the seam line, W_{cut} is the weight per unit L_{seam} , N_{cut_stoit} is the number of stoit points where the scissors turn, and W_{cut_stoit} is the weight per stoit point. We use the weights $W_{cut} = 1.1s/cm$ and $W_{cut_stoit} = 3.2s/stoit$, following [3]. The number of start and endpoints of the seams are N_{cut_stoit} , and therefore $N_{cut_stoit} = N_{seam}$.

Sewing Cost: The cost of sewing (C_{sew}) increases proportionally to total seam length, which is the sum of the circumference of each piece. The cost of sewing all seam lines is represented as $W_{sew}L_{seam}$, where W_{sew} is the weight per unit L_{seam} . The cost of knotting at seam endpoints also affects sewing time. The cost of knotting is represented as $W_{knot}N_{knot}$, where N_{knot} is the total number of knots and W_{knot} is the cost per knot. N_{knot} equals $N_{seam}(= 2N_{seam}/2)$ because

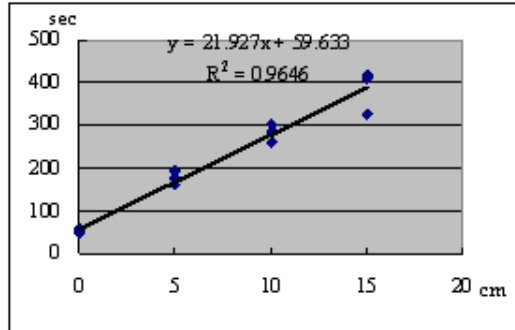


Fig. 5. Seam length and sewing time of a subject

the user knots at the start and endpoints of a seam. Therefore, the cost of sewing (C_{sew}) is represented as follows:

$$C_{sew} = W_{sew}L_{seam} + W_{knot}N_{knot}. \quad (4)$$

3.2 Estimation of Weights

We ran an experiment to estimate the two weights W_{sew} and W_{knot} . Five users sewed three lines (5, 10, and 15cm) by reverse-stitching, and we recorded their sewing times. According to the result, we set W_{sew} to 22s/cm and W_{knot} to 80s/knot.

3.3 Discussion

The experiment revealed that sewing time differs markedly between individuals. In addition, it is difficult to estimate the exact sewing time because of several unpredictable factors such as the physical interference between cloth and string. However, the study did reveal that for each individual, sewing time was well approximated as a linear function of the total seam length (correlation coefficients: 0.8-0.9 (Fig.5)). The prediction result provides valuable information to avoid excessively complicated seam designs.

4 Implementation

Pillow, our prototype system, is implemented as a JavaTM program. Flattening and reconstruction run in real-time on a standard PC. We use ABF++ as the flattening algorithm [4]. It changes a 3D surface segment to a 2D pattern by minimizing the distortion in the angle space. For the physics simulation, we currently use a very simple mass-spring model. First, each vertex is moved in a normal direction to mimic outward pressure (Fig. 6(a)). Then, the system adjusts edge length (Fig. 6(b)). We plan to combine our system with more sophisticated simulation methods [5] in the future.

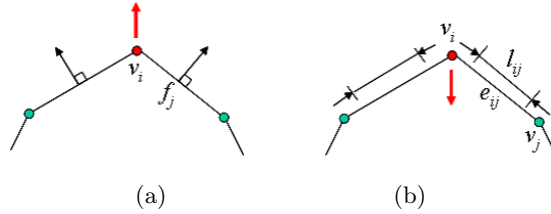


Fig. 6. Our simple model to mimic the effect of stuffing. (a) First, the system moves each face to its normal direction to simulate the effect of internal pressure; (b) then the system adjusts the length of each edge to preserve the integrity of the cloth.

5 Results

We created 2D patterns for a teddy bear model using our system and then created real plush toys using the resulting patterns. We used Teddy [6] for creating the input 3D model, and a standard polygonal mesh representation, so that users can use any polygonal model available on the Internet. It is also possible to create one's own model using a standard modeling program. Figure 1 shows an example of a seam design, simulation, and the final product. The simulation successfully captured the overall shape of the real plush toy. We also applied our method to design a balloon (Fig. 7). Figure 2 shows the pattern used to create this balloon. According to a professional balloon designer, the construction of a



Fig. 7. A balloon created from the pattern shown in Figure 2

balloon can be roughly divided into five steps: creating a 3D model using standard 3D modeling software, manually designing a 2D pattern by cutting and pasting paper sections representing the 3D model (our model was a 1/10 scale version of the final balloon), scanning the 2D pattern and tracing the contours using a commercial drawing program, scaling the traced 2D patterns to actual size and printing it using a plotter, and cutting and sewing the 2D pattern. The required time from request to delivery is typically 1 month. Using Pillow, steps 1-3 can be significantly shortened (to about 2 weeks), thereby halving total production time.

6 Conclusion and Future Work

We developed Pillow, an interactive pattern-design system for creating plush toys and other products. Using the system, users can create 2D patterns easily, and experiment with various segmentations before actually sewing real fabric. In the future, we plan to improve the flattening algorithm to take cloth properties into account because existing algorithms (e.g., [7], [4]) for 2D parameterization do not consider cloth property. We also plan to incorporate more sophisticated simulation methods.

References

1. Mori, Y., Igarashi, T.: Plushie: An interactive design system for plush toys. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)* 23(3), Article No.45 (2007)
2. Julius, D., Kraevoy, V., Sheffer, A.: D-Charts: quasi developable mesh segmentation. *Computer Graphics Forum (Proc. Eurographics 2004)* 24(3), 981–990 (2004)
3. Mitani, J.: Research for a design support system for three-dimensional paper models using computer. The University of Tokyo Ph.D. dissertation (in Japanese) (2004)
4. Sheffer, A., Levy, B., Mogilnitsky, M., Bogomyakov, A.: ABF++: Fast and robust angle based flattening. *ACM Transactions on Graphics* 24(2), 311–330 (2005)
5. Grinspun, E., Krysl, P., Schroder, P.: CHARMS: A simple framework for adaptive simulation. *ACM Transactions on Graphics* 21(3), 281–290 (2002)
6. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A sketching interface for 3D freeform design. In: *Proc. SIGGRAPH 1999*, pp. 409–416 (1999)
7. Levy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics* 21(3), 362–371 (2002)