

Knitty: 3D Modeling of Knitted Animals with a Production Assistant Interface

Yuki Igarashi¹ Takeo Igarashi^{1,2} and Hiromasa Suzuki¹

¹The University of Tokyo, Japan ²JST SORST

Abstract

Knitty is an interactive design system for creating knitted animals. The user designs a 3D surface model using a sketching interface. The system automatically generates a knitting pattern and then visualizes the shape of the resulting 3D animal model by applying a simple physics simulation. The user can see the resulting shape before beginning the actual knitting. The system also provides a production assistant interface for novices. The user can easily understand how to knit each stitch and what to do in each step. In a workshop for novices, we observed that even children can design their own knitted animals using our system.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques Interaction Techniques I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling Geometric Algorithms

1. Introduction

Making knitted animals (Fig. 1c) is part of traditional Japanese culture. Animals are made by assembling several knitted parts, which are either hollow bodies or flat pieces (see the ears of the monkey in Fig. 1c). The former are made by spirally knitting yarn and then stuffing with cotton. A knitting pattern (Fig. 1b, right) is a special diagram from which knitters can read sequences of stitches to produce each part.

To design an original knitted animal, it is necessary to construct an appropriate knitting pattern, but this is very difficult for novices. For this reason, all existing knitting patterns have been created by professional designers. To enable laypeople, including novices, to design their own original animals, we developed Knitty, an interactive knitted animal design system. Using Knitty, the user can design a 3D knitted animal model using a sketching interface. The system automatically generates a knitting pattern and also maps the 3D shape of the resulting animal by applying a simple physics simulation. The system provides a production assistant interface to guide novices in easily understanding how to knit each stitch and what to do in each step. It also estimates total knitting time. We ran a small workshop for novices to try our system, and observed that even children can design their own knitted animals using this system.

Various methods have been proposed for making toys from 3D models. Mitani and Suzuki [MS04] and Shatz et al. [STL06] presented automatic segmentation of a 3D

model into developable patches for constructing paper craft models. Julius et al. [JKS05] and Mori and Igarashi [MI06] presented methods for creating stuffed animals from 3D models. The Plushie system [MI07] combines a 3D modeling process and physics simulation to design original stuffed animals. In this paper, we propose a new algorithm for creating a 3D surface made of knitted yarn instead of cloth. Our system also provides a production assistant interface for novices. The user can easily learn how to knit a given pattern using this interface.

2. 3D modeling interface

Fig. 1 shows an overview of our system. Knitty consists of two windows: one shows the 3D knitted animal model being constructed, and the other shows the corresponding knitting pattern (Fig. 2). A knitting pattern is a pattern figure for making animals, equivalent to 2D cloth patterns for making stuffed animals. The user can design a 3D model on the 3D model window using a mouse or pen tablet. The system automatically creates a knitting pattern based on the user's input strokes. The system also visualizes the shape of the resulting animal by applying a simple physics simulation. The modeling operations are based on the gestural interface introduced by Igarashi et al. [IMT99]. The system also provides special editing operations tailored to knitted animal design.

Creating a new object: The user creates a new knitted animal model by drawing its silhouette as a closed freeform figure. After the user draws a silhouette of the model, the

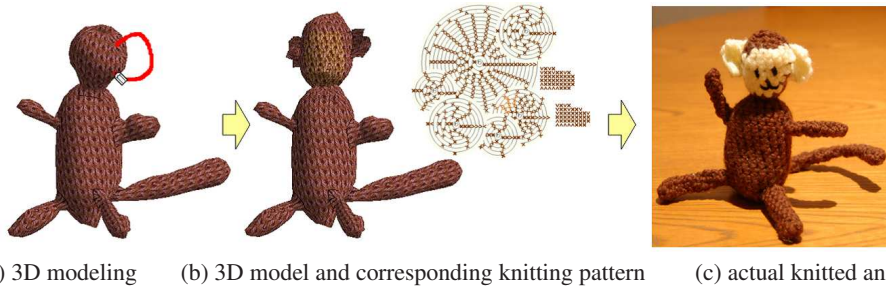


Figure 1: Illustration of the Knitty system. (a) The user interactively designs a 3D model using a sketching interface. (b) The system automatically generates a 3D knitted model and corresponding knitting pattern. (c) The final product made from the knitting pattern.

system automatically generates a knitting pattern. The system also visualizes the shape of the resulting animal by applying a simple physics simulation (Fig. 3, top). If the user wishes to bend the model, a wire is inserted inside the actual knitted animal (Fig. 3, bottom). In future, we hope to create a pattern that can accommodate bending without the use of a wire.

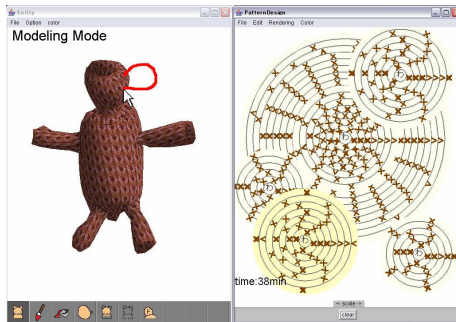


Figure 2: Screenshot of the Knitty system.

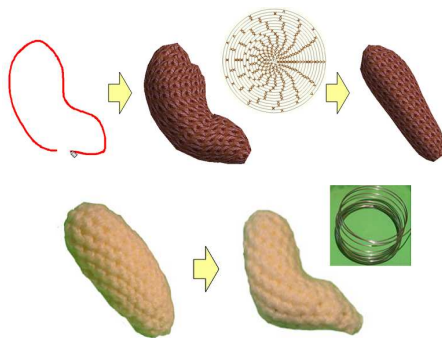


Figure 3: Creating a new object and an actual knitted animal using the pattern (top) and twisting it using a wire (bottom).

Adding parts: The user can add protruding parts, such as arms or legs, to the base model by drawing a single stroke that defines the silhouette of the part (Fig. 4, top). The stroke should start and end on the base model. The user can also

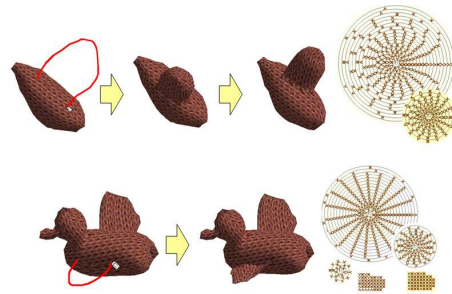


Figure 4: Adding protruding (top) and flat (bottom) parts.

create a flat part in a special mode, which is useful for creating parts such as ears or feathers (Fig. 4, bottom).



Figure 5: The paint tool.

Using the paint tool: The user can color the 3D model using the paint tool (Fig. 5). The system automatically updates the color of corresponding stitches on the knitting pattern.

3. Production assistant interface

This mode teaches a novice how to make a knitted animal. The user can change the thickness of the yarn (thick, normal, or thin) and the size of the crochet needle (size 5-10). The system updates the number of stitches and indicates the total knitting time and total length of yarn required. The system automatically marks the current knitting position on the knitting pattern column when the user pushes the up/down

key (Fig. 6 left). Clicking the right mouse button on the knitting pattern also shows the user how to knit (Fig. 6 right).

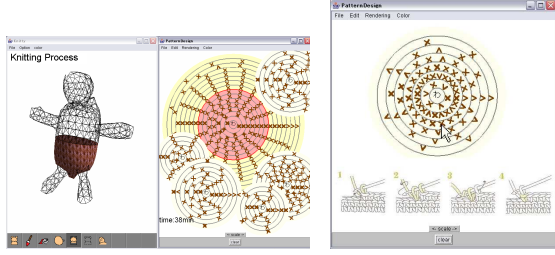


Figure 6: The production assistant interface. Right-clicking the mouse button on the knitting pattern shows the user how to knit.

4. Algorithm

Knitty is implemented as a JavaTM program. A physical simulation runs in real-time on a 1.1 GHz Pentium M PC. As the hollow parts of the animal are knit circularly around an axis, the knitted animal takes on a nearly axisymmetric shape. Therefore, we used a polygonal mesh representation similar to a sweep-based model [HYKJ03, YK06] or generalized cylinder [SK97]. Actual knitted animals can contain many types of stitches, but our system uses the three most basic stitches (normal, increase, and decrease stitches) because it was designed for novice users.

Create a new object: We create a closed planar polygon by connecting the startpoint and endpoint of the stroke, and determine the spines or axes of the polygon using the chordal axis [Pra97]. After the user draws a stroke (Fig. 7a), the system constructs a 2D polygon (Fig. 7b). It then performs constrained Delaunay triangulation of the polygon (Fig. 7c). The system divides the triangles into three categories: those with two external edges (terminal triangles), those with one external edge (sleeve triangles), and those with no external edges (junction triangles) [IMT99]. The chordal axis is obtained by connecting the midpoints of the internal edges (Fig. 7d). To remove branches, the system extracts the longest chordal axis after searching from terminal triangle to terminal triangle (Fig. 7e).

After extracting a single axis, the system constructs vertices (v_i) on the axis, and a plane (P_i) that passes through the vertex v_i and has normal vector v_{i-1} to v_i . The system calculates radius r_i from the user's input stroke to the center vertex (v_i). The system then constructs circular vertices on the plane (P_i).

The system then stitches between the circular vertices on a plane P_i and the circular vertices on the next plane P_{i+1} (Fig. 8). The system calculates the number of vertices and constructs the circular vertices on the plane as shown in Fig. 8. The first plane (P_0) and the last plane (P_n) have only a center vertex, and so the system constructs virtual edges.

The system also constructs a knitting pattern using the number of stitches and kind of stitches. As the final shape of the actual knitted animal is determined by the physical interaction between the knitted "skin" and inner cotton stuffing, a physical simulation must be conducted to modify the input shape to be consistent with such physical conditions.

We used a simple static method for the physical simulation as in Plushie [MI07] to achieve interactive modeling on a standard PC. The system first moves each vertex to its normal direction to simulate the effect of the internal pressure exerted by the inner cotton. It then adjusts the length of each edge to preserve the integrity of the knit yarn. This length enforcement is applied to vertical and horizontal edges only and is not applied to diagonal edges (Fig. 9).

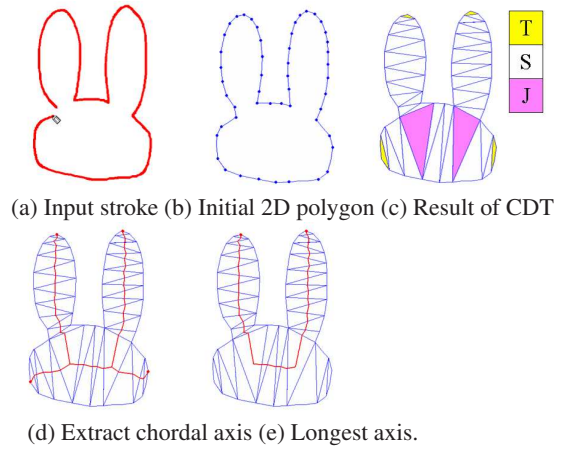


Figure 7: Algorithm of a single axis. CDT = constrained Delaunay triangulation; T = terminal triangle; S = sleeve triangle; J = junction triangle.

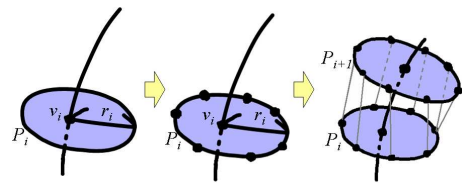


Figure 8: Stitching between a plane P_i and the next plane P_{i+1} .

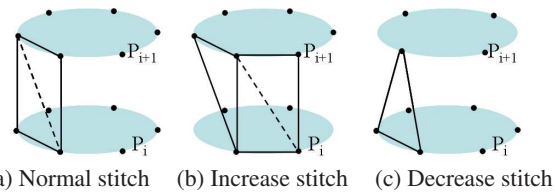


Figure 9: Stitching between a plane P_i and the next plane P_{i+1} . Length enforcement is applied to solid edges only and is not applied to dashed edges.

Algorithm for adding a part: The system projects the two endpoints of the input stroke onto the base model surface. It constructs a plane that passes through these 3D points and faces toward the screen, and then projects the input stroke onto it. In normal mode, the system then draws a base circle on the model surface that passes through two endpoints of the stroke. Finally, the base circle is swept along the projected extruding stroke (Fig. 10, top). In flatpart mode, the system constructs a plane that passes through two endpoints of the input stroke and faces toward the screen. It then projects the input stroke onto this plane. The system constructs a single skeleton on the plane and stitches on the flat plane (Fig. 10, bottom).

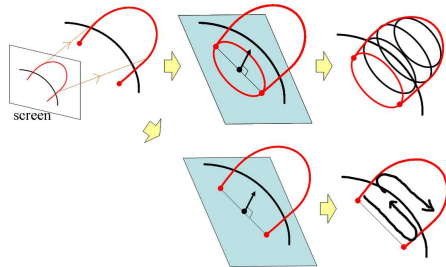


Figure 10: Algorithm for adding a part.

Production assistant interface: The total knitting time varies depending on the design, and it is very helpful to know during the design phase how long it will take to create the actual knitted animal. The system automatically estimates the total knitting time and presents this information to the user during the design phase. As a result of a preliminary user experiment, we set the time for creating a stitch to 4 s and the time for changing yarn to 55 s. Knitting time varies greatly between individuals, so the absolute time may not be accurate. However, the estimate is useful for comparing various designs. The system also visualizes which column the user is currently stitching. This region is rendered with texture, and the remaining region is rendered as a wire frame model.

4.1. Results

We designed a couple of animals using our system and created actual knitted animals based on the resulting patterns (Fig. 11). A modeling session typically took 10 to 20 min, and knitting took 3 to 5 hours. We ran a small workshop for novices to try our system. Ten children aged 10 to 14 accompanied by their parents participated in the workshop and created their own knitted animals using the system. Most did not have prior experience with knitting. It took about one hour to design the animals (Fig. 12, left), and they knitted the actual animals in approximately 3 hours (Fig. 12, right).

5. Conclusion and Future Work

We developed Knitty, an interactive design system that enables users to easily design original knitted animals.



Figure 11: Knitted animals designed using our system.



Figure 12: Modeling (left) and knitting (right) original animals in the workshop.

In future, we plan to develop additional easy-to-use systems for novices incorporating more sophisticated simulation methods. We also plan to create knitted animals from existing 3D models using a single thread of yarn.

References

- [HYKJ03] HYUN D. E., YOON S. H., KIM M. S., JUTTLER B.: Modeling and deformation of arms and legs based on ellipsoidal sweeping. In *Computer Graphics and Applications* (2003), pp. 204–212.
- [IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proceedings of ACM SIGGRAPH 1999* (1999), pp. 409–416.
- [JKS05] JULIUS D., KRAEVOY V., SHEFFER A.: D-charts: quasi developable mesh segmentation. *Computer Graphics Forum* 24, 3 (2005), 981–990.
- [MI06] MORI Y., IGARASHI T.: Pillow: Interactive pattern design for stuffed animals. In *DVD publication at SIGGRAPH 2006 Sketches* (2006).
- [MI07] MORI Y., IGARASHI T.: Plushie: An interactive design system for plush toys. *ACM Transactions on Graphics* 23, 3 (2007), 45.
- [MS04] MITANI J., SUZUKI H.: Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Transactions on Graphics* 23, 3 (2004), 259–263.
- [Pra97] PRASAD L.: Morphological analysis of shapes. In *CNLS Newsletter* (1997), pp. 139: 1–18.
- [SK97] SHAFF E. B., KUIJLAARS A. B. J.: Distributing many points on a sphere. *Springer-verlag new york* 19, 1 (1997).
- [STL06] SHATZ I., TAL A., LEIFMAN G.: Paper craft models from meshes. *The Visual Computer: International Journal of Computer Graphics* 22, 9 (2006), 825–834.
- [YK06] YOON S. H., KIM M. S.: Sweep-based freeform deformations. *Computer Graphics Forum* 25, 3 (2006), 487–496.