Beady: Interactive Beadwork Design and Construction

Yuki Igarashi[*]
Univ. of Tsukuba / JSPS Research Fellow

Takeo Igarashi[†]
The Univ. of Tokyo / JST ERATO

Jun Mitani[‡]
Univ. of Tsukuba / JST ERATO

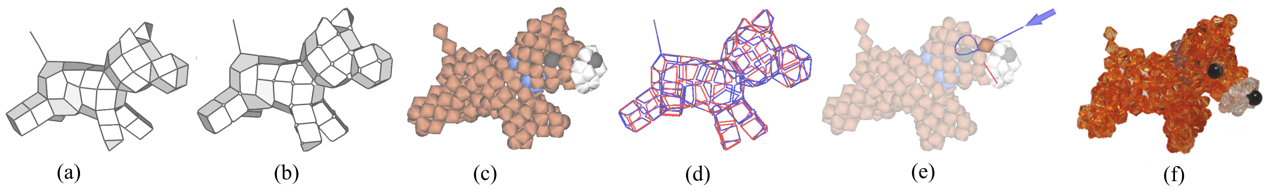

Figure 1: *System overview. (a) The user creates a design model. (b) The system runs a simulation on a structure model to adjust the geometry. (c) The beadwork model visualizes the expected result. (d) The system computes the wire path. (e) The system provides a step-by-step construction guide. (f) The user manually constructs a physical beadwork.*

1 Introduction

Beadwork is the art of connecting beads together by wires. While common beadwork is two-dimensional (2D), three-dimensional (3D) beadwork is also popular in oriental regions such as Japan and China. However, the design and construction of 3D beadwork is very difficult. The final shape is defined by the complicated three-dimensional interaction between beads and wires, thus making it very difficult to design manually. One also needs to specify an appropriate wire path to hold the beads together and to manually insert the wire into the beads one by one following the path to construct the beadwork. Careful observation of existing beadwork structures shows several geometrically interesting problems, which make beadwork design an interesting technical challenge.

This paper presents an interactive computational system to assist the design of original beadwork and its construction. Fig. 1 shows the overall process. The user first creates a polygonal mesh model, called a *design model*, which represents the overall structure of the beadwork (Fig. 1(a)). A bead of the beadwork is represented as an edge (not a vertex) of the design model. The system then converts the design model into a beadwork model by placing beads on the edges with the appropriate wiring (Fig. 1(c)). Finally, the user manually constructs the physical beadwork by following the step-by-step instruction generated by the system (Fig. 1(e, f)).

The design of a 3D model of a real-world object requires that certain physical constraints be satisfied. For example, a paper toy model has to be represented as a set of developable patches. This kind of restriction is often discussed in the design of architecture consisting of freeform surfaces [Liu et al. 2006; Schiftner et al. 2009]. Pottmann et al. [2010] called this new research area *architectural geometry*. Related to this trend, recent work presented methods to slightly modify the geometry of a polygonal mesh to reduce the number of unique polygons contained in the original model [Singh and Schaefer 2010; Eigensatz et al. 2010; Fu et al. 2010].Our work differs in that we support the interactive design of a model instead of pursuing automatic conversion.

The contributions of this work are summarized as follows. (1) We present a mesh modeling user interface specialized for beadwork models by combining gestural operations and physical simulations. (2) We present an algorithm based on face stripification to compute wire paths for a beadwork model designed by the user. (3) We present a step-by-step construction guide to assist the user in the manual construction process. (4) We show the feasibility and effectiveness of our approach by presenting a solid implementation.

[*]e-mail: yukim@acm.org
[†]e-mail:takeo@acm.org
[‡]e-mail:mitani@cs.tsukuba.ac.jp

2 User Interface

2.1 Geometric Modeling

The system provides a specialized modeling interface for the design of a simple polygonal mesh model with uniform edge length. The user first builds an approximate shape by combining predefined primitives. Next, the user modifies the shape by applying basic mesh editing operations such as face extrusion, edge insertion, edge split, edge deletion, and vertex merger (Fig. 2). We provide a modeless gestural interaction for applying these operations to support rapid exploration. The system also allows the user to add auxiliary parts (bead chains) to the model. The user specifies the color and shape of individual beads in the beadwork model by using a painting interface.

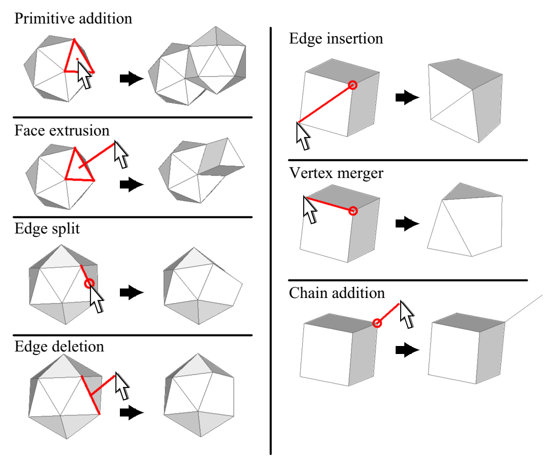

Figure 2: *Mesh editing operations.*

2.2 Construction Guide

The system guides the manual construction of a physical beadwork by showing step-by-step instruction. The traditional printed beadwork instructions in textbooks use a specialized 2D diagram representation as the guide, but it is very tedious and difficult because the user needs to keep track of the relation between beads in the physical 3D beadwork and those in the 2D instruction. Our step-by-step instruction takes advantage of the expressiveness of interactive 3D graphics and makes it easier to understand the construction procedure.

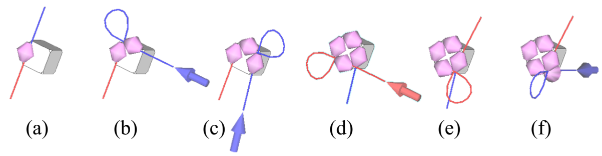

Figure 3: *An example of the visual construction guide. (a) Initial state. (b,c,f) Blue wire passes a newly added bead. (d) Red wire passes a newly added bead. (e) Red wire passes an existing bead.*

The construction guide shows which wire passes which bead in each step as 3D graphics. Fig. 3 shows an example sequence. The user can view each step from an arbitrary viewing direction. The user presses the "next" button to proceed to the next step and presses the "prev" button to return to the previous step.

3 Algorithm

Conversion of the design model into a *beadwork model* consists of two main steps. The first step is geometry computation. The system generates another polygonal mesh model, called a *structure model*, by adding local wire connections between neighboring beads of the design model (Fig. 1(b)). The system runs a physical simulation to compute the geometry of the resulting beadwork model by considering the physical interaction among beads and wires. This simulation runs during interactive modeling and updates the shape of the design model after each editing operation. The second step is wire path planning (Fig. 1(d)). The wire path should be appropriately defined to efficiently connect the beads. We show that a valid wire path is given as an Euler loop on a graph derived from the structure model.

We start the process with the structure model (Fig. 4(b)). It defines the local wire connectivity. A global wire path is given as a loop that meets all wire edges once and all bead edges twice. This is the Euler loop of the wire graph (Fig. 4(c)), obtained by contracting the bead edges into vertices, with an additional constraint that a wire going into a bead should go out from the other side of the bead. The existence of an Euler loop is guaranteed by the construction because each bead edge always has four wire edges. Various methods exist for obtaining such an Euler loop.

However, an arbitrary Euler loop (e.g., Fig. 4(d)) can be inconvenient in the manual construction process because it can cause many unstable beads during construction (Fig. 4(e)). An *unstable bead* makes manual construction extremely difficult because the user has to manually hold it. We consider a bead to be *stable* when the position of the bead is stably fixed to a specific location by the wires. We carefully examined existing beadwork designs and found that the problem of reducing unstable beards can be solved by using a face strip (Figs. 4(f, g)). The design model is covered by a face strip, and then a wire path is placed so that it completes the faces in the strip one by one.

This method makes the beads in the previously visited faces all stable during construction (Fig. 4(h)). However, a single wire path often makes manual construction process difficult. We therefore present an algorithm that covers the design model with multiple face strips with short branches, where each strip corresponds to a wire.

4 Results

The prototype system is implemented using Java on a laptop (1.2 GHz CPU, 2 GB RAM). The top two rows of Fig. 5 show some example beadwork designed and constructed by using the system. Designing these models required 10 to 20 minutes including creative experimentation and exploration. Manual construction of each by following the guide required a few hours. The beadwork bear is an exception: it was bought at a local shop. We created the model by referring to the product spending approximately 90 minutes.

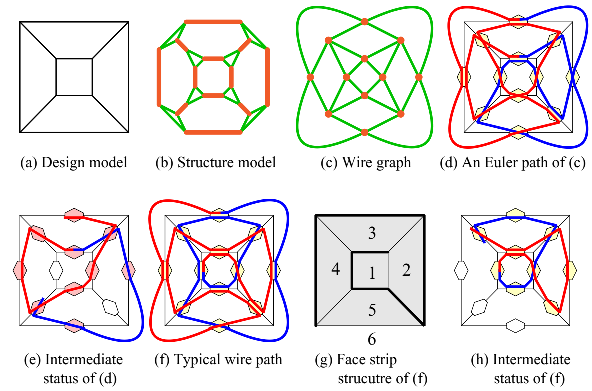

Figure 4: *Details of the wire path-planning algorithm. Yellow beads are stable beads and pink beads are unstable beads.*

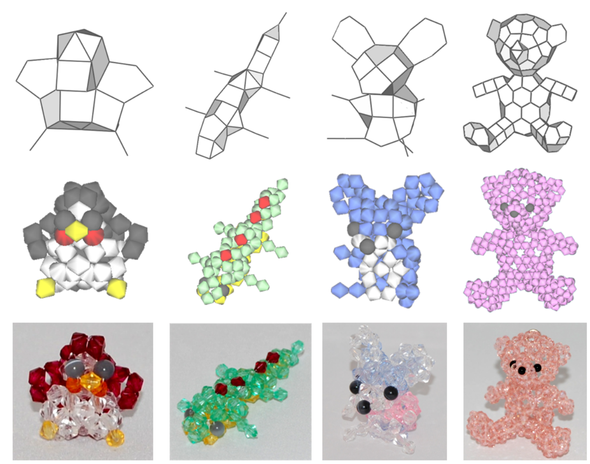

Figure 5: *Example models designed by the authors. Top: design models. Middle: beadwork models. Bottom: real beadworks.*

References

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graph. 29* (July), 45:1–45:10.

FU, C.-W., LAI, C.-F., HE, Y., AND COHEN-OR, D. 2010. K-set tilable surfaces. *ACM Trans. Graph. 29* (July), 44:1–44:6.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25* (July), 681–689.

POTTMANN, H., HUANG, Q., DENG, B., SCHIFTNER, A., KILIAN, M., GUIBAS, L., AND WALLNER, J. 2010. Geodesic patterns. *ACM Trans. Graph. 29* (July), 43:1–43:10.

SCHIFTNER, A., HÖBINGER, M., WALLNER, J., AND POTTMANN, H. 2009. Packing circles and spheres on surfaces. *ACM Trans. Graph. 28* (December), 139:1–139:8.

SINGH, M., AND SCHAEFER, S. 2010. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph. 29* (July), 46:1–46:7.