Interactive Hexagonal Mesh Editing for Beadwork Design

Yuki Igarashi*
University of Tsukuba

Takeo Igarashi
The University of Tokyo / JST ERATO

Jun Mitani
University of Tsukuba / JST ERATO

1 Introduction

Beadwork is the art of connecting beads together with wire. Igarashi et al. [2012] presented an interactive beadwork design system called Beady to help non-professionals design their own 3D beadwork. They observed that existing beadwork designs, especially large ones, typically consist of hexagonal faces. This is probably because a hexagonal mesh (honeycomb lattice) is the most efficient structure for holding flat surfaces with minimal support materials. After conducting physical simulations, they also found that a near-hexagonal mesh, obtained as the dual of a triangular mesh, yields a more aesthetically pleasing beadwork model. However, the interactive modeling interface of the original Beady system did not consider this. Thus the user had to carefully combine various editing operations to construct a near-hexagonal polyhedron. Existing 3D modeling software is also inconvenient for near-hexagonal mesh modeling. Therefore, we introduce mesh-editing operations specifically designed for creating near-hexagonal polyhedra. By combining the original Beady interface with our method, the user can more easily design near-hexagonal polyhedra.

2 Hexagonal Editing Operations

The user designs a near-hexagonal mesh via a gestural user interface based on the original Beady interface. We provide four special mesh-editing operations (merge faces, flip faces, insert faces and delete faces) for creating near-hexagonal meshes. Because it is sometimes easier to use standard editing operations to make small changes, the hexagonal mesh-editing operations are included as an option. The user carries out these operations while holding down the Shift key.

Merge faces: Drawing a line from one face to another face with a shared edge between them merges these faces (Fig. 1). The system takes the dual of the mesh and collapses the edge shared by the target faces. Then the system again takes the dual of the mesh.

Flip faces: When the user draws a line from one face to another face, and the two faces are bridged by an edge, the system flips the faces (Fig. 2). The system takes the dual of the mesh and flips the bridging edge. Then the system again takes the dual of the mesh.

Insert faces: Pressing the mouse button with the cursor on an edge and moving the cursor with the mouse button held down selects the edges touched by the cursor. This operation inserts new faces along the selected edges (Fig. 3). The system opens the mesh along the selected edges and inserts new faces in the gap, in such a way that the valences of the vertices along the gap are as close to three as possible. We formulate this as an optimization problem that finds a best match among the vertices on the two sides of the gap. The system inserts new edges connecting the matched pairs. The cost function is given by $\Sigma_{v \in v_0} |valence(v) - 3|$, where v_0 denotes the vertices along the gap, and $valence(v)$ denotes the valence of vertex v after inserting the edges. We impose the constraint that the matching order is monotonic, and solve the optimization problem using a beam search.

Delete faces: Pressing the mouse button with the cursor on a face and moving the cursor with the mouse button held down selects the faces touched by the cursor (Fig. 4). The system deletes the selected faces and closes the gap by merging the vertices along the gap. This is again formulated as an optimization problem that finds a best match among the vertices on the two sides of the gap. The system merges the matched pairs. The cost function is given by $\Sigma_{v \in v_1} |valence(v) - 3|$, where v_1 denotes the vertices after the merge, and $valence(v)$ denotes the valence of the merged vertex v. We again impose the constraint that the matching order is monotonic, and solve the optimization problem using a beam search.

*e-mail: yukim@acm.org

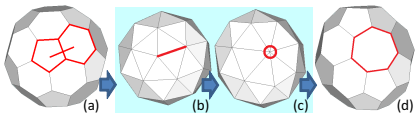

Figure 1: *Merge faces. The system takes the dual (b), collapses the edge (c), and again takes the dual (d).*

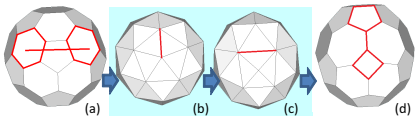

Figure 2: *Flip faces. The system takes the dual (b), flips the edge (c), and again takes the dual (d).*

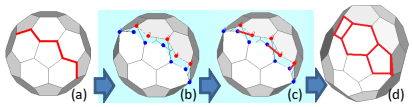

Figure 3: *Insert faces. The system opens the mesh (b) and inserts new edges (c).*

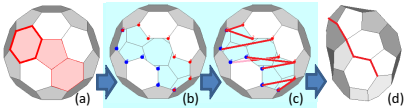

Figure 4: *Delete faces. The system deletes faces (b) and merges the matched vertex pairs (c).*

3 Results

Figure 5 shows modeling sequences using the near-hexagonal mesh-editing operations. The user can design these near-hexagonal models without expending any additional effort to make them hexagonal.

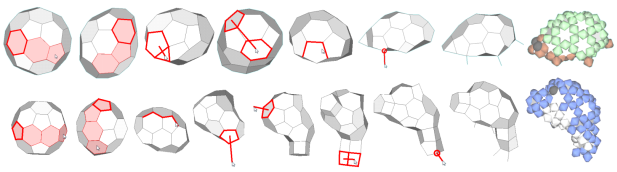

Figure 5: *Example of modeling sequences using the near-hexagonal mesh-editing operations.*

References

IGARASHI, Y., IGARASHI, T., AND MITANI, J. 2012. Beady: Interactive beadwork design and construction. *ACM Trans. Graph. 31*, 4.