An Interactive Design System for Deltahedron-based Modular Origami

N. Tsuruta

Abstract: *In this paper, we propose an interactive design system for modular origami that have the underlying geometry of deltahedra. By using operations such as augmentation and elongation, various deltahedra can be generated. Our system helps users create physical models by first visualizing assembled forms of them by placing simple unit-like models on each edge.*

1 Introduction

Modular origami is a technique that is used for composing larger structures from smaller origami units. The process requires the target structure to follow some geometric constraints because assembly is restricted to a single, or very limited, number of unit types. One of the more well-known modular systems, called Sonobe, uses such units to construct deltahedra that have pyramids on each face. However, while this type of modular system is capable of creating any deltahedral surfaces, it is difficult to design desired shapes using it because the face shapes are strictly constrained to equilateral triangles.

In this paper, we propose an interactive design system for polyhedra that have faces that are congruent with equilateral triangles. This allows users to create shapes without being conscious of geometric constraints, simply by augmenting and elongating an initial deltahedron. Operations implemented in our system are described in Section 3. Our system visualizes the assembled form of a deltahedron by placing simple unit-like models on each edge. Colors are assigned to each paper unit so that no two units with the same color interlock. The use of our proposed system will contribute to the creation of a wide variety of deltahedron-based modular origami works.

2 Related Work

Constructing a deltahedral surface is a difcult problem. The tetrahedral-octahedral honeycomb is a good example of a method for producing equilateral triangle tessellation. In another study, Lang and Hayes proposed an algorithm for generating surfaces from aperiodic patterns and modular systems [Lang and Hayes 13].

In addition, Tamura et al. proposed a method that approximates any three-dimensional shape using a space lled with octahedra and tetrahedra [Tamura et al. 10].

Although this method allows the detail level to be controlled by changing the resolution, the surface eventually takes on a jagged appearance because only two kinds of deltahedra are used.

Another approach is discussed by Singh and Schaefer in response to the paneling problem, which is an architectural exercise that aims at finding ways to reduce the number of panels required to build a large structure [Singh and Schaefer 10]. That study described a method for approximating a given triangular surface with the small number of unique triangle faces. While the number of unique triangles can be reduced, their face shapes are not limited to equilateral triangles.

In the eld of computer graphics (CG), a remeshing method using an equilateral triangle grid has been proposed [Payan et al. 15]. In this method, the input surface is first divided into several patches, after which each patch is remeshed using a grid. Merging all of the patches together results in semi-regular triangular surfaces. While this CG technique can be used to improve the quality of surfaces, the triangles that are produced are too small for use in the creation of physical origami.

Our proposed method uses an augmentation-based approach for interactive deltahedra design. Although the forms that can be constructed are limited compared to those that can be produced via the approximation or tessellation approaches, the geometric constraints are easier to maintain.

3 Our Proposed System

Our system constructs a desired deltahedron by using the following editing operations: augmentation, elongation, tucking, subdivision, and regularization. Figure 1 shows our system interface. First, the user chooses an initial polyhedron (which should be a deltahedron), after which the augmentation and the elongation operations are used to make the object larger. The regularization tool is helpful when geometric constraints are eliminated. This tool deforms the polyhedron being edited by numerically solving an optimization problem in order to ensure that all edges have the same length. Finally, our system visualizes the three-colored assembled form of a deltahedron. The details of each operation are described below.

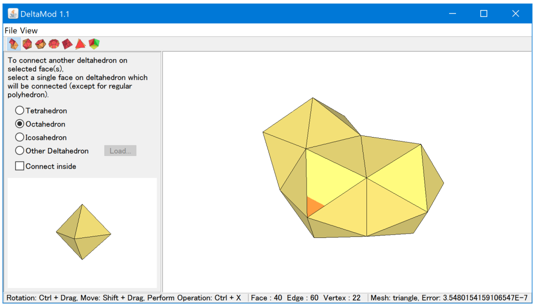


Figure 1: *Our proposed system.*

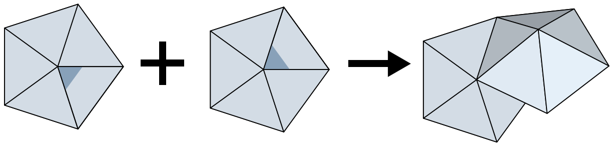


Figure 2: *Augmentation example: joining two pentagonal dipyramids. Colored corner indicates that the face is selected in that orientation.*

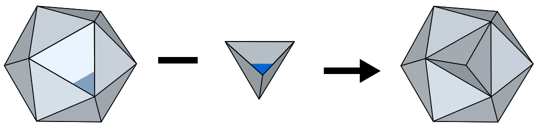


Figure 3: *Attaching a tetrahedron to the inside of the selected face on an icosahedron.*

3.1 Augmentation

Augmentation is an operation that joins two polyhedra by aligning and connecting faces from each polyhedron. The user first selects the faces to join and a vertex for each deltahedron. After joining, the selected faces are removed because they have become hidden. Figure 2 shows an augmentation example that joins two pentagonal dipyramids.

This operation is basically used to make an object larger, but we can add a polyhedron inside the selected face if it is smaller than the original polyhedron. An example of hollowing augmentation is shown in Figure 3. Since the hollow space is filled with flipped faces, the face orientations are corrected so that the polyhedron is closed.

The initial polyhedron does not have to be deltahedron. However, if a different polyhedron is used, the polygonal faces must be replaced with a set of equilateral triangles. Square and regular pentagonal faces can be removed by joining square and pentagonal pyramids, respectively. The augmenting Johnson solid is one of the ways to remove other regular polygons.

3.2 Elongation

In elongation, a prism is inserted between the selected and other faces. In other words, the selected faces are extruded by the edge length. This operation creates square faces that must be removed by joining other polyhedrons to each face. Figure 4 shows an elongation example in which square pyramids are joined on the inserted square faces, whereas Figure 5 shows an example of gyloelongation, in which an antiprism is inserted to maintain triangle faces.

Both operations work properly when the boundary of the selected faces forms a plane curve. In other cases, the edge length of the inserted faces will not be the same.

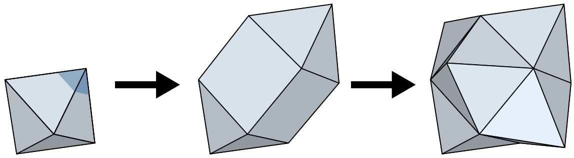


Figure 4: *Octahedron elongation example. The square faces on the side can be removed by placing square pyramids on them.*

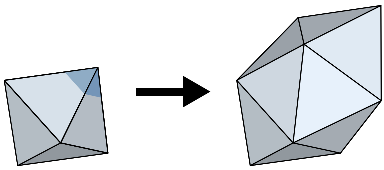


Figure 5: *Octahedron gyroelongation.*

3.3 Tuck/Pull

A pyramid part of the polyhedron can be tucked inside or pulled out of a polyhedron. Figure 6 shows an example of tuck in a pyramid-like part of an icosahedron. This operation also works when the boundary of the selected faces is a plane curve. The operation shown in Figure 6 can also be performed by a hollowing augmentation operation with a pentagonal dipyramid.

3.4 Subdivision

This operation divides all equilateral triangle faces into smaller faces. Subdivision operations do not change the overall shape and are useful for designing self-similar surfaces, as shown in Figure 7. A special subdivision case is used when removing a hexagonal face. A regular hexagon can be replaced with six equilateral triangles.

3.5 Regularization

Regularization operations deform the polyhedron in order to make their edge lengths unique. These operations do not guarantee that the resulting polyhedra will have unique edge lengths, so it differs from other operations on that point. We implemented a geometric optimization method [Tsuruta et al. 16] that numerically solves

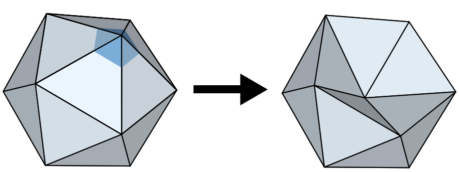


Figure 6: *Tuck in a pyramid-like part of an icosahedron.*

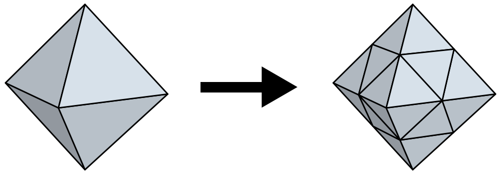


Figure 7: *Subdivision operation example.*

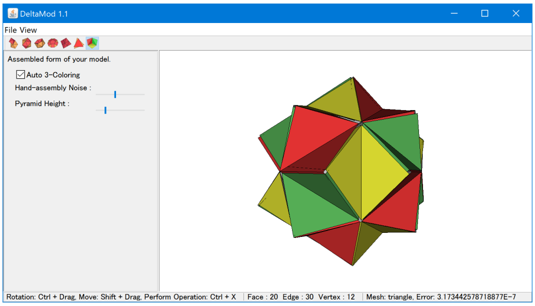


Figure 8: *3-colored assembled form of icosahedron made with isosceles triangle units.*

the error function used to calculate the sum of length edge differences. If a polyhedron has a small number of non-unique edges, regularization works well. However, in cases involving completely inconsistent edges, it causes excessive deformation and/or does not converge.

3.6 Assembled Form and Coloring

The assembled form of a deltahedron is generated by placing simple unit-like models on each edge (Figure 8). Herein, we have implemented two models, one of which resembles an isosceles triangle unit, and another resembles a Sonobe unit. These models are shown in Figures 10a and 10b, respectively. While these models can be used to construct a pyramid with three pieces, we can obtain various other forms by changing the pyramid heights. The folding angles of each unit are calculated using given height parameters.

Physical models created by hand often contain misalignments and gaps between

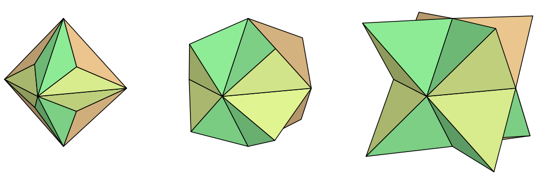
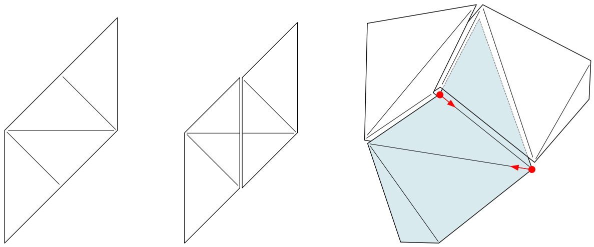
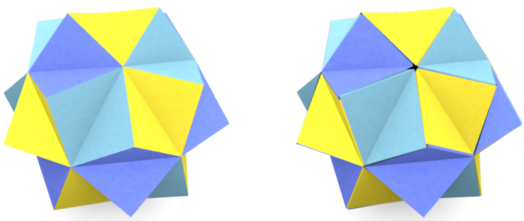


Figure 9: *Assembled form with different pyramid height parameters.*



(a) (b) (c)

Figure 10: *Simple unit models and assembling image with three pieces: (a) Isosceles triangle unit, (b) Sonobe unit (one unit with two pieces), and (c) Assembly consisting of three pieces. The arrows show the deformation direction of the vertices.*



(a) (b)

Figure 11: *Comparison of (a) accurate model, and (b) deformed model. These images were rendered using 3D CG software (Blender)*

modules. Accordingly, we introduced random deformations in order to reproduce a hand-assembled appearance. To accomplish this, two unit model vertices were moved slightly in the direction shown in Figure 10c. Figure 11 shows examples of accurate and deformed models. The deformation may cause interpenetrating faces, however, improves the visual appearance and makes the deformed model look more natural than the accurate version.

A unit piece, such as a Sonobe unit, corresponds to a polyhedron edge. This modular system is 3-colorable. In other words, we can assemble an object in which all adjacent pieces have different colors using just three colors [Hull 94]. As shown in Figure 8, we have implemented a simple backtracking algorithm that allows our system to visualize the coloring result of an assembled form.

4 Results and Discussion

We implemented our system using the Java programming language. A number of deltahedra constructed using our system are shown in Figure 12. As can be seen in the figure, we could the construct simple shapes of an animal, a motif,

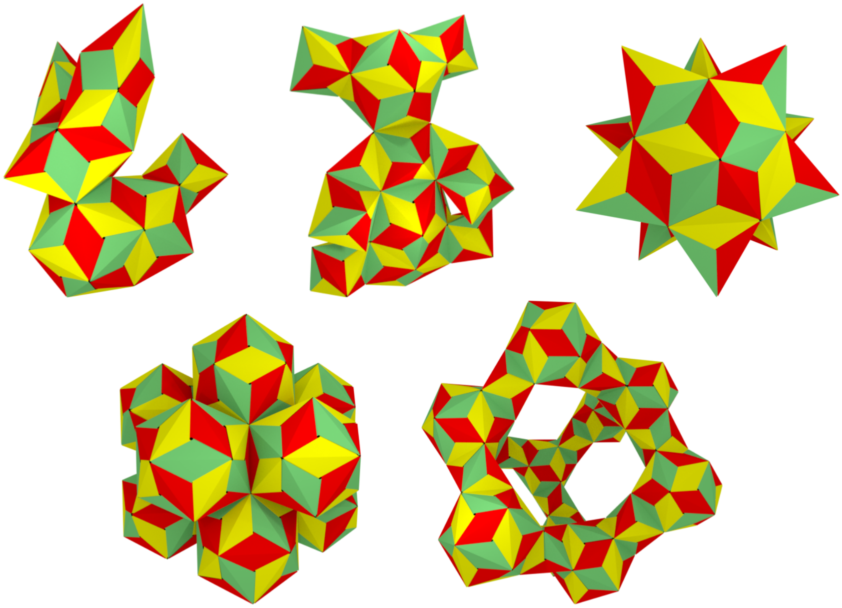


Figure 12: *Deltahedra constructed using our system.*

and geometric forms with holes. The bottom left deltahedron was constructed by augmenting pentagonal pyramids on each pentagonal face of the dodecahedron, after which we gyroelongated all the pentagonal pyramids.

The origami unit used for the model shown in Figure 12 is a 30° isosceles triangle unit. The assembled form will be the same as its underlying deltahedron because this unit constructs zero-height pyramids. Figure 13 shows a physical model constructed with 96 origami units.

Since our system does not have a collision detection mechanism, the user must conrm that the model has no interpenetrating faces after each operation. Interpenetrating faces may also appear in the assembled form because the additional pyramids are made using three units.

Because the most basic operation of our system is augmentation, the resultant deltahedra tend to have numerous branching parts that make roundish or inated shapes hard to create. The optimization process is one of methods that can be used to solve this problem. However, it sometimes converges too far from the original polyhedron, as noted in the Section 3.5, so other operations will be required to design a wider variety of deltahedra.

5 Conclusions

Herein, we proposed an interactive system for designing the deltahedra in which users can construct a desire deltahedron via operations such as augmentation and

Figure 13: *Assembled model with 96 origami pieces of 30° isosceles unit.*

elongation. The assembled forms of deltahedron are generated by placing simplified unit models on each edge. Our system is useful for creating a physical model by allowing it to be visualized beforehand.

Our ultimate goal is to provide a system that can be used to create any form of deltahedron. Although our regularization process does not work for large polyhedra, a wide range of deltahedra can be constructed if augmentation is used in combination with regularization. To improve this approach, the use of 3D segmentation method may be helpful for breaking down the input model into small parts.

References

[Hull 94] Thomas Hull. "Planar graphs and modular origami." *Origami science and art: Proceedings of the Second International Meeting of Origami Science and Scientific Origami*, p. 151159. Otsu, Shiga, Japan.

[Lang and Hayes 13] Robert J. Lang and Barry Hayes. "Paper Pentasia: An periodic Surface in Modular Origami." *The Mathematical Intelligencer* 35 (2013), 61–74.

[Payan et al. 15] Frdric Payan, Cline Roudet, and Basile Sauvage. "Semi-Regular Triangle Remeshing: A Comprehensive Study." *Computer Graphics Forum* 34:1 (2015), 86–102.

[Singh and Schaefer 10] Mayank Singh and Scott Schaefer. "Triangle surfaces with discrete equivalence classes." *ACM Transactions on Graphics* 29:4 (2010), Article 46.

[Tamura et al. 10] Tomokazu Tamura, Yoshiaki Takai, and Nami Takai. "3D Shape Approximation by Unit-Origami." *IPSJ SIG Technical Report* 141:13 (2010), 1–6. Available online (https://ci.nii.ac.jp/naid/110007997512/).

[Tsuruta et al. 16] Naoya Tsuruta, Jun Mitani, Yoshihiro Kanamori, and Yukio Fukui. "Random Realization of Polyhedral Graphs as Deltahedra." *Journal of Geometry Graphics* 19:2 (2016), 227–236.

Naoya Tsuruta
Tokyo University of Technology, e-mail: tsurutany@stf.teu.ac.jp