Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2013
August 4-7, 2013, Portland, Oregon, USA

DETC2013-12326

FREEFORM ORIGAMI TESSELLATIONS BY GENERALIZING RESCH'S PATTERNS

Tomohiro Tachi
Department of General Systems Studies
Graduate School of Arts and Sciences
The University of Tokyo
3-8-1 Komaba, Meguro-Ku, Tokyo 153-8902, Japan
Email: tachi@idea.c.u-tokyo.ac.jp

ABSTRACT

In this research, we study a method to produce families of origami tessellations from given polyhedral surfaces. The resulting tessellated surfaces generalize the patterns proposed by Ron Resch and allow the construction of an origami tessellation that approximates a given surface. We will achieve these patterns by first constructing an initial configuration of the tessellated surfaces by separating each facets and inserting folded parts between them based on the local configuration. The initial configuration is then modified by solving the vertex coordinates to satisfy geometric constraints of developability, folding angle limitation, and local non-intersection. We propose a novel robust method for avoiding intersections between facets sharing vertices. Such generated polyhedral surfaces are not only applied to folding paper but also sheets of metal that does not allow 180° *folding.*

INTRODUCTION

Origami is the art of folding a sheet of paper into various forms without stretching, cutting, or gluing other pieces of paper to it. Therefore, the concept of origami can be applied to the manufacturing of various complex 3D forms by out-of-plane deformation, i.e., bending and folding, from a watertight sheet of hard material such as paper, fabric, plastic, and metal. By definition, origami is a developable surface; however, unlike a single G^2 continuous developable surface, i.e., a single-curved surface, origami enables complex 3D shapes including the approximation of double-curved surfaces. Therefore, by utilizing origami, we can create a desired surface from a single (or a small number of) developable part(s), instead of using the papercraft approach of making an approximation of the desired surface by segmenting it into many single-curved pieces and assembling them again.

An advantage of folding for use in fabrication is that the resulting 3D form is specified by its 2D crease pattern because of the geometric constraints of origami. This helps in obtaining a custom-made 3D form by half-cutting, perforating, or engraving an appropriately designed 2D pattern by a 2- or 3-axis CNC machine such as a laser cutter, cutting plotter, and milling machine. Origami fabrication can be a fundamental technology for do-it-yourself or do-it-with-others types of design and fabrication. Here, computational methods are required for solving the inverse problem of obtaining a crease pattern from a given folded form based on the topological and geometric properties that origami has.

A generalized approach to realize the construction of an arbitrary 3D origami form is to use the *Origamizer* method [1], which provides a crease pattern that folds the material into a given polyhedron. The method is based on creating flat-folded tucks between adjacent polygons on the given surface and crimp folding them to adjust the angles such that they fit the 3D shape of the surface. However, the flat folds, i.e., 180° folds, and the crimp folds that overlays other flat folds on the folded tucks produce kinematically singular complex interlocking structures. This forbids the origami model to be made with thick or hard materials, and is a significant disadvantage in applications to personal or industrial manufacturing processes. Additionally, even as a folding method

for thin sheets of paper, it requires an expert folder to fold such complex origami models.

On the other hand, important designs of 3D origami tessellation patterns and/or their structural applications have been investigated, e.g., series of 3D origami tessellations by Fujimoto [2], PCCP shells and Miura-ori [3], tessellation models by Huffman (see [4] for the reconstruction work), and Resch's structural patterns [5, 6]. In this paper, we focus on the series of patterns proposed by Resch in the 1960s and 70s; one of these patterns is shown in Figure 1. If we look at its final 3D form, we can observe that the surface comprises surface polygons and tucks to be hidden similar to the origamizer method; the difference is that the tuck part is much simpler and can exist in a half-folded state as well (Figure 2). The flexibility of a half-folded tuck not only avoids interlocking structures but also controls the curvature of the surface by virtually shrinking the surface to form a double-curved surface. The pattern in Figure 1 forms a synclastic (positive Gaussian curvature) surface when it is folded halfway. However, possible 3D forms are limited by their 2D patterns, e.g., the aforementioned pattern cannot fold into an anticlastic surface. In order to obtain a desired freeform double-curved surface, the generalization of the 2D patterns from a repetitive regular pattern to appropriately designed crease patterns is necessary.

The author had previously proposed the system *Freeform Origami* for interactively editing a given pattern into a freeform by exploring the solution space or hypersurface formed by the developability constraints [7]. However, the method for generating the initial pattern suited for the target 3D form was not investigated in this approach. Moreover, collisions between facets at each vertex are not sufficiently taken into account in the existing method, whereas in complex tessellated origami models, such as the one we are targeting at in this study, the collision between facets is fundamental because facets sharing vertices frequently touch each other.

In this paper, we propose a system for generating 3D origami tessellations that generalize Resch's patterns. This is achieved by inserting a tuck structure in the 3D form and numerically solving the geometric constraints of the developability and local collision (Figure 3). First, we delineate the method for generating the topology and initial estimate configuration of the tessellation pattern from a given polyhedral surface. Then, we propose a novel, robust method for numerically solving the developable configuration, taking into account the local collisions between facets sharing vertices. We illustrate design and fabrication examples based on this method.

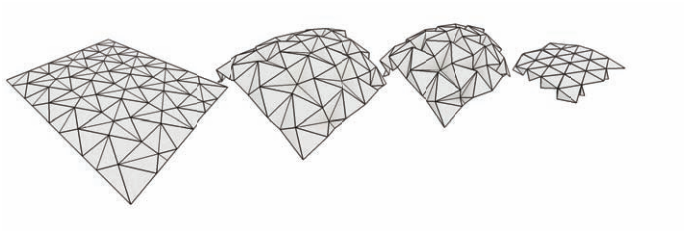

FIGURE 1. Regular triangular tessellation by Resch.

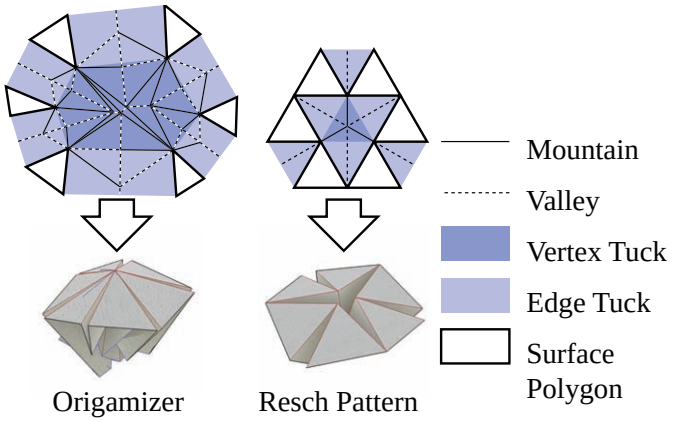

FIGURE 2. Origamizer and Resch's tessellation. Both are comprised of surface polygons and tucks that are hidden. Notice that Resch's pattern can have the tuck folded halfway, whereas origamizer vertex keeps the tuck closed because of the crimp folds.

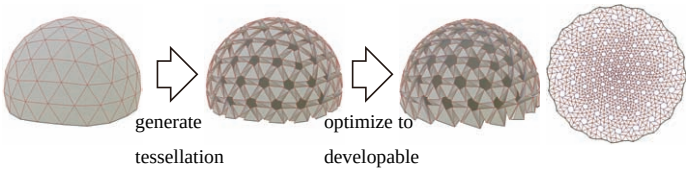

FIGURE 3. The process for obtaining a freeform origami tessellation.

GENERATING INITIAL CONFIGURATION

We first generate the families of origami tessellations from a polyhedral tessellation. Since the initial polyhedral mesh corresponds to the patterns that appear on the tessellated surface, the surface can be re-meshed to have a homogeneous or adaptive tessellation pattern using established algorithms for triangulating or quadrangulating meshes. Here, we focus on the topological correctness and the validity of the mountain and valley assignment of fold lines and ignore the validity of the material being an origami surface.

Basic Pattern

The basic Resch-type origami tessellation is generated by the insertion of a star-like folded tuck; here, we call such a structure a *waterbomb tuck* [1]. First, we assume that every vertex of the planar tessellation has an even number of incident edges, and thus the facets can be colored with two colors (say, red and blue) similar to a checkerboard pattern. For each vertex with $2n \geq 6$ edges, we insert a star tuck comprising a corrugated triangular fan with $2n$ triangles surrounding the *pivot vertex* created on the backside offset position of the original vertex. The star tuck structures are inserted by splitting facets, where the split occurs only at one of the sharing vertices of the adjacent facets. The separating vertex is chosen such that from the viewpoint of the vertex, the left and right incident facets are colored red and blue, respectively (Figure 4 Top).

For a general tessellation that is not colored into a checkerboard pattern with a vertex incident to odd number of edges, we color every facet red and insert a blue digon between each pair of adjacent facets, so that every vertex with n edges is replaced by a $2n$-degree vertex (Figure 4 Bottom). This makes it possible for any mesh connectivity to be used as the initial mesh.

In general, there is no guarantee that a developable mesh can be constructed with this procedure alone. Special well-known cases, such as regular triangular, quadranglular, and hexagonal tilings allow the construction of developable meshes as shown in Figure 5, when the depth d of the pivot vertex is adjusted to $\ell \cot \pi/n$, where ℓ is the length of the edges, and $2n$ is the number of boundary edges of the star tuck. For a non-planar general polyhedron, we use the value of d above and the normal vector at the vertex for determining the offset position of the pivot vertex. Then, we parametrically shrink each facet by scaling with respect to its center by $0 < s \leq 1$. This builds up gaps between facets to make the connecting tuck in a halfway-unfolded state.

Variations

Figure 6 shows variations of the parametric tuck structures that can be used for the construction of origami tessellations. The regular versions of original star tuck, the truncated star, and the twist fold are used in Resch's original works, while the curly star is not.

Truncated Star The star shape can be truncated so that the pivot vertex is replaced by a flat n-gon for an n-degree vertex, and each valley fold is replaced by a triangle between two valley folds, splitting the fold angle in halves. The amount of truncation is an additional controllable parameter, which allows for increased freedom in the design space to flexibly fit to the desired 3D form in the succeeding numerical phase.

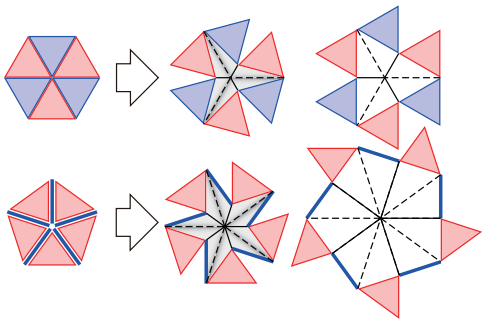

FIGURE 4. Top: Insertion of a star tuck. Bottom: Vertex with odd number of incident edges n can be interpreted as the vertex with $2n$ edges by the insertions of digons between the facets.

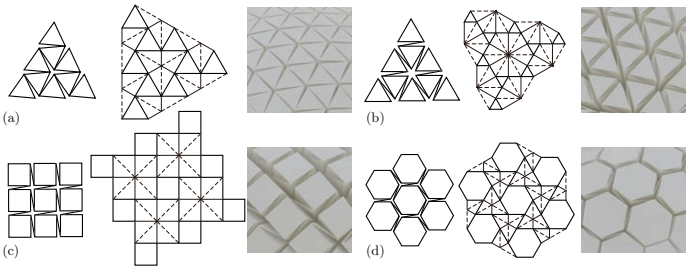

FIGURE 5. Example tessellations generated from regular planar tilings. (a) Triangular pattern with regular 6-degree vertices. (b) (c) (d) Triangular, quadrangular, and hexagonal pattern with the insertion of digons.

Curly Star By adding extra folds to the star tuck, we can have a curly variation of the star tuck. The surface polygons are pulled together by twisting the star to fit to a surface with an increased curvature than the original star tucks. Here, the amount of twist is an additional controllable parameter.

Twist Fold By flattening the pivot vertex of the curly star, we can obtain a truncated n-gonal bottom figure with connecting triangles. This is a 3D interpretation of planar twist tessellations [8, 9]. The amounts of truncation and twist are the additional controllable parameters.

SOLVING CONSTRAINTS

From the generated approximation of folding, a valid origami surface, and thus, a developable mesh without intersection, is numerically computed by solving nonlinear equations. The vari-

[1] Waterbomb tuck is thus a generalization of waterbomb base used for origami tessellation

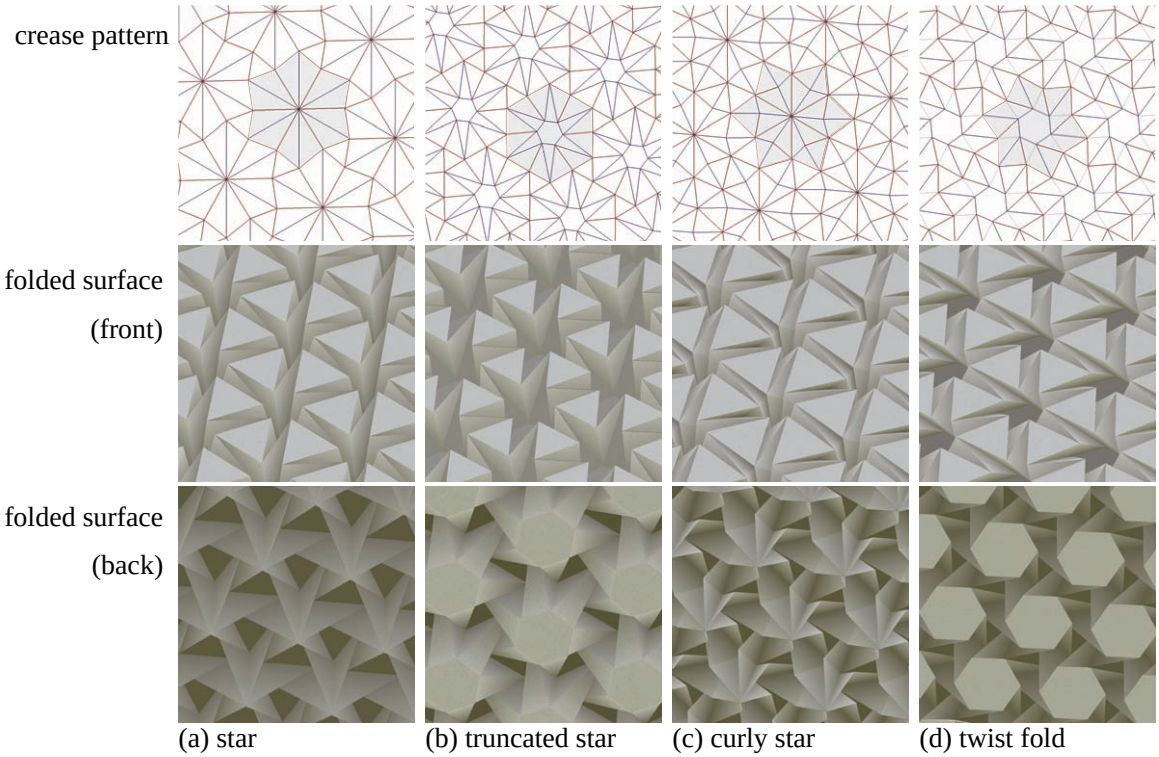

FIGURE 6. Variations of the tuck structures for a regular triangular mesh.

ables in this system are coordinates of n vertices $\mathbf{x}_1,\ldots,\mathbf{x}_n$ that can be represented as a single $3n$ vector $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)^{\mathsf{T}}$, and the equations are the developability conditions as in [7]. When we apply the developability constraints directly, our generated origami tessellations produce multiple facets intersecting each other at their sharing vertices (vertex-adjacent facets). The method proposed in [7] using the simple penalty function for fold angles suffers from the instability at the singular configuration where the facets are very thin. Also, the approach was not capable of dealing with intersections between vertex-adjacent but not edge-adjacent facets. Here, we introduce a novel robust technique to avoid local intersections between facets based on constructing angular constraints for edge-adjacent and vertex-adjacent facets.

Developability Constraints

The isometric mapping of the entire polyhedral disk to a plane is ensured by the angular condition of each interior vertex: for each interior vertex v with n_v incident facets given by

$$g_v(\mathbf{X}) = 2\pi - \sum_{i=0}^{n_v-1} \theta_{v,i} = 0, \tag{1}$$

where $\theta_{v,i}$ is the sector angle of the i-th (mod n_v) facet incident to v. This condition is a necessary but not sufficient condition for having a one-to-one isometric mapping; thus, the development of the mesh can overlap itself at the boundary. Although boundary overlapping does not frequently occur in our generated pattern, we avoid the boundary collision using the following sufficient condition, similar to the one used for the post process of angle-based flattening [10]. Denote the loop of n_0 boundary edges in counterclockwise, $e_1,\ldots,e_{n_0}$. For any portion of the loop, $e_i, e_{i+1},\ldots,e_j$ in modulo n_0, the sum of the outer angles is

$$b_{i,j}(\mathbf{X}) = \sum_{v=i}^{j-1} \angle(e_v, e_{v+1}) = \sum_{v=i}^{j-1} \left(2\pi - \sum_{\ell=1}^{n_v} \theta_{v,\ell}\right) + \pi \geq 0. \tag{2}$$

This condition forbids any pair of edges e_i and e_j on the boundary to form an angle greater than 180° clockwise when the edges appear in a counterclockwise order. In order to satisfy the condition, we extract pairs of edges that do not follow this constraint from the candidate edges at the point of inflection of the boundary curve. For such pairs we apply constraints $b_{i,j}(\mathbf{X}) = 0$.

Intersection Constraints

Pairs of facet that can intersect are classified into the following three types.

1. Edge-adjacent facets, i.e., facets sharing an edge.
2. Vertex-adjacent (but not edge-adjacent) facets, i.e., facets share a vertex but not an edge.
3. Non-adjacent facets, i.e., none of the above.

The global intersection of type 3 is not specific to origami, and such an intersection is less frequent than the other two cases. Such an intersection is comparatively easier to be observed and avoided through the user manipulation by dragging the surface using the mouse input. In this paper, we focus on the local intersections types 1 and 2, which are specific to origami.

Edge-Adjacent Facets Avoiding the intersections between edge-adjacent facets is straightforward. This can be done by keeping the fold angle, i.e., the outer angle of the dihedral angle, between adjacent facets in a valid range of $[-\pi, \pi]$. Since the configuration described by vertex coordinates cannot distinguish folding angles ρ and $\rho + 2\pi$, and thus $-\pi - \delta$ and $\pi - \delta$, we use the mountain and valley assignment of fold lines to crop the angles to $(-\pi/2, 3\pi/2]$ for valley folds and $(-3\pi/2, \pi/2]$ for mountain folds. We force the valley and mountain folds to have cropped fold angle in $[0, \pi]$ and $[-\pi, 0]$, respectively.

A straightforward representation of the aforementioned condition is to directly use the fold angle as the constraint equation as in [7]: for any foldline with fold angle ρ exceeding the limit fold angle ρ_{limit}, $\rho_{i,j} - \rho_{limit} = 0$. However, this can make the calculation unstable since ρ is undefined at degenerated configurations in which one of the facets is too thin to have a reliable normal vector. In order to avoid such an instability, we modify the constraint as

$$f_{p,q}(\mathbf{X}) = 2\left(\sin\frac{\rho_{i,j}}{2} - \sin\frac{\rho_{limit}}{2}\right) h_p h_q = 0, \qquad (3)$$

where h_p and h_q are the relative heights of triangles p and q, respectively, measured from the sharing base edge $e_{i,j}$ (Figure 7). This can be written as

$$f_{p,q} = \frac{2\left(\sin\frac{\rho_{i,j}}{2} - \sin\frac{\rho_{limit}}{2}\right)}{\cot\theta_{p,i}\cot\theta_{p,j}\cot\theta_{q,i}\cot\theta_{q,j}}. \qquad (4)$$

Figure 8 illustrates $\rho(\mathbf{x})$ and $f_{p,q}(\mathbf{x})$ when the axis projected to $\mathbf{x}$ can freely move between fixed edges drawn as fixed points.

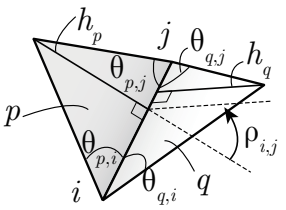

FIGURE 7. Pair of edge-adjacent facets p and q.

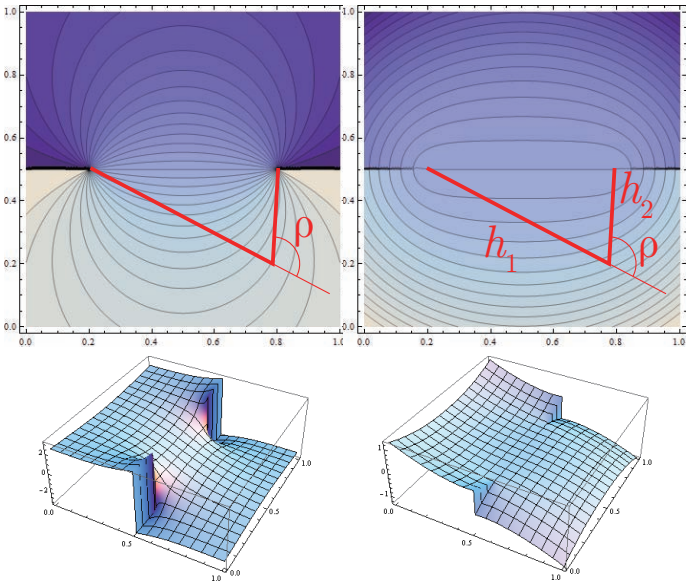

FIGURE 8. Left: Fold angle $\rho(\mathbf{x})$ of the vertex between two fixed points. Note the singularity at the end points. Right: Modified angular evaluation $f_{p,q}(\mathbf{x})$.

Vertex-Adjacent Facets The constraints mentioned above cannot deal with the local collision between facets sharing a vertex if a vertex is shared by more than four facets (Figure 9 Left). The intersection problem at a single vertex is essentially equivalent to that of a 2D closed chain, e.g., an unfolding algorithm of a 2D chain can be applied in the unfolding of a single vertex [11]. An unfolding algorithm of a 2D chain uses a barrier function to avoid self intersection [12]. Our approach is similarly based on the energy-driven approach, however, we use a penalty function instead of a barrier function that would make the solution unique and forbid searching of the entire solution space; the forbidden configuration includes an interesting boundary case in which a foldline touches a facet. Our approach is to construct an appropriate penalty function whose value stays zero when the configuration is valid and continuously increases when a foldline sinks into a facet. Note that the method we propose only applies to interior vertices and provides only a necessary condition. The limitation comes from that a penalty function temporarily allows

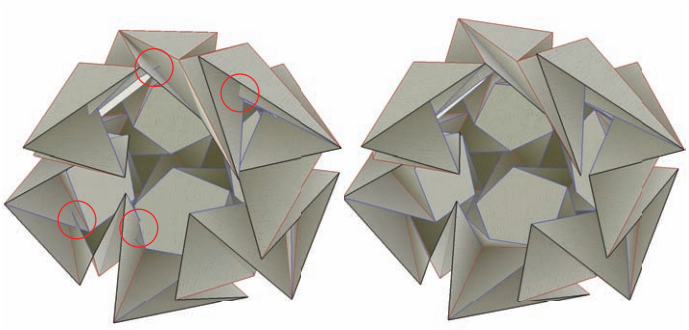

FIGURE 9. Left: Local collision between vertex-adjacent facets. Right: Mesh after the penalty function is applied to avoid the intersection.

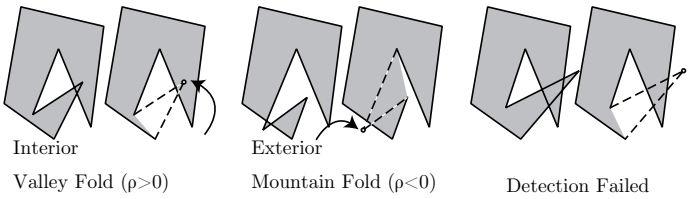

FIGURE 10. Detection of invalid vertices.

for the intersection and there is no obvious way to unfold it if the intersection is too far to be fixed.

First, the intersection of facets sharing a vertex v can be detected by finding an "invalid foldline" as follows. We first cut a unit sphere with the facet fan and call the portion of the sphere with area smaller than 2π the "interior." Without the loss of generality, we assume that the front side of the surface is in this direction. Since the vertex is developable, the interior portion stays within a hemisphere and thus remains to be interior throughout the folding motion.

For each foldline, we derive another facet fan by removing the two incident facets and capping the gap by a new triangular facet. We can similarly construct the portion by intersecting the derived facet fan with a unit sphere, which also stays within the hemisphere. The foldline is detected as invalid when either (1) the foldline has a positive fold angle $\rho > 0$ (valley fold) and is located in the interior of the derived facet fan, or (2) the foldline has a negative fold angle $\rho < 0$ (mountain fold) and is located on the exterior of the derived facet fan (Figure 10). An edge is detected to be interior of a facet fan if the sum of the view angle $\sum_i^{n_v-1} \alpha_{o,i,i+1}$ of each facet around the edge equals 2π and exterior if they sum up to 0. The view angle from an edge represented by a vector $\mathbf{v_o}$ to the facet spanning two vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$, where the vectors are normalized, is calculated as follows

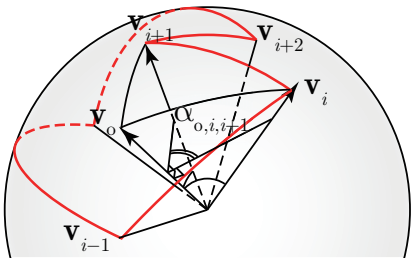

FIGURE 11. View angle $\alpha_{o,i,i+1}$ and the vectors.

(Figure 11).

$$\alpha_{\mathbf{o},i,i+1} = \arctan \frac{\|(\mathbf{v_o} \times \mathbf{v}_i) \times (\mathbf{v_o} \times \mathbf{v}_{i+1})\|}{(\mathbf{v_o} \times \mathbf{v}_i) \cdot (\mathbf{v_o} \times \mathbf{v}_{i+1})} \tag{5}$$

This per-edge method successfully detects the cases when the configuration is close to valid, whereas two pairs of facets intersecting each other simultaneously could fail to be detected as an invalid case using the method (Figure 10 Right).

The configuration can be modified using a penalty function for each invalid foldline to pull back to a position on the closest boundary. We form such a function using an angular measurement similar to Equation 3 to represent the distance between $\mathbf{v_o}$ and the facet spanning $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$.

$$\begin{aligned} d_{\mathbf{o},i,i+1} &= (1+\cos\alpha_{o,i,i+1})\,\|\mathbf{v_o} \times \mathbf{v}_i\|\,\|\mathbf{v_o} \times \mathbf{v}_{i+1}\| \\ &= \cos\angle(\mathbf{v}_i, \mathbf{v}_{i+1}) - \cos(\angle(\mathbf{v_o}, \mathbf{v}_i) + \angle(\mathbf{v_o}, \mathbf{v}_{i+1})) \end{aligned} \tag{6}$$

The constraints for the total facet fan can be represented using the harmonic mean of the distance functions.

$$f_{\mathbf{o}}(\mathbf{X}) = \frac{n_v - 1}{\sum_i^{n_v-1} \frac{1}{d_{\mathbf{o},i,i+1}}} \tag{7}$$

Figure 12 illustrates an example of the resulting penalty function.

Numerical Solution

We solve the nonlinear constraints and penalty function given by equations g, b, and f in an iterative manner. The equations are given as a vector $\mathbf{g}(\mathbf{X}) = \mathbf{0}$. We solve the geometric constraints based on the generalized Newton-Raphson method using the search direction of $-\mathbf{C}(\mathbf{X}_i)^+\mathbf{g}(\mathbf{X_i})$ for each step, where $\mathbf{C}(\mathbf{X})^+$ is the Moore-Penrose generalized inverse of the Jacobian matrix $\mathbf{C}(\mathbf{X})$. Since the constraints are the function of angles $\boldsymbol{\theta}$

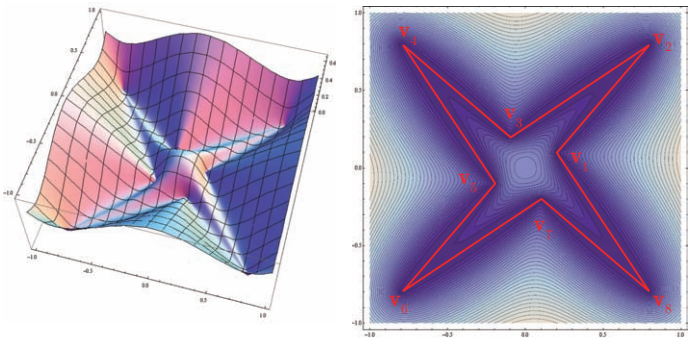

FIGURE 12. Penalty function given from a star-like facet fan. Illustrated on a plane by the Gnomonic projection of the spherical surface.

and $\boldsymbol{\rho}$ between edges and facets, this is calculated by

$$\mathbf{C}(\mathbf{X}) = \frac{\partial \mathbf{g}(\mathbf{X})}{\partial \mathbf{X}} = \frac{\partial \mathbf{g}(\boldsymbol{\theta},\boldsymbol{\rho})}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{X}} + \frac{\partial \mathbf{g}(\boldsymbol{\theta},\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \mathbf{X}}. \tag{8}$$

We use the conjugate gradient method for each step to calculate the least norm search direction.

Fitting

The geometric constraints are generally less than the variables and the system constructs a multi-dimensional solution space within which we can search for the solutions that look attractive and are desirable. This can be done in an interactive manner as in [7], using the initial deformation mode $\Delta\mathbf{X}_0$ arbitrarily given by the user through a 2D input device.

$$\Delta\mathbf{X} = \left[\mathbf{I} - \mathbf{C}(\mathbf{X})^{+}\mathbf{C}(\mathbf{X})\right] \Delta\mathbf{X}_0 \tag{9}$$

Here, $\Delta\mathbf{X}$ is the modified deformation mode projected to the constructed solution space.

In order to obtain a surface closest to the original polyhedral surface, we can set the fitness function and use its gradient as the initial deformation mode. An example fitness function is given by the distance of vertices on the outer positions to that of the original polyhedral surface. For each vertex of the upper facets whose coordinate is $\mathbf{x}$, we set the target position $\mathbf{x}^{targ}$ by referring to the initial positions of the generated surface. We set the fitness function

$$d = \sum_{v} \left((1-w)\mathbf{n} + w(\mathbf{x}_v - \mathbf{x}_v^{targ}) / \left\|\mathbf{x}_v - \mathbf{x}_v^{targ}\right\|\right)^{\mathsf{T}} (\mathbf{x}_v - \mathbf{x}_v^{targ}), \tag{10}$$

where $\mathbf{n}$ is the normal vector at the original vertex position and $0 \leq w \leq 1$ is the weight for the distance measured perpendicular to the normal vector.

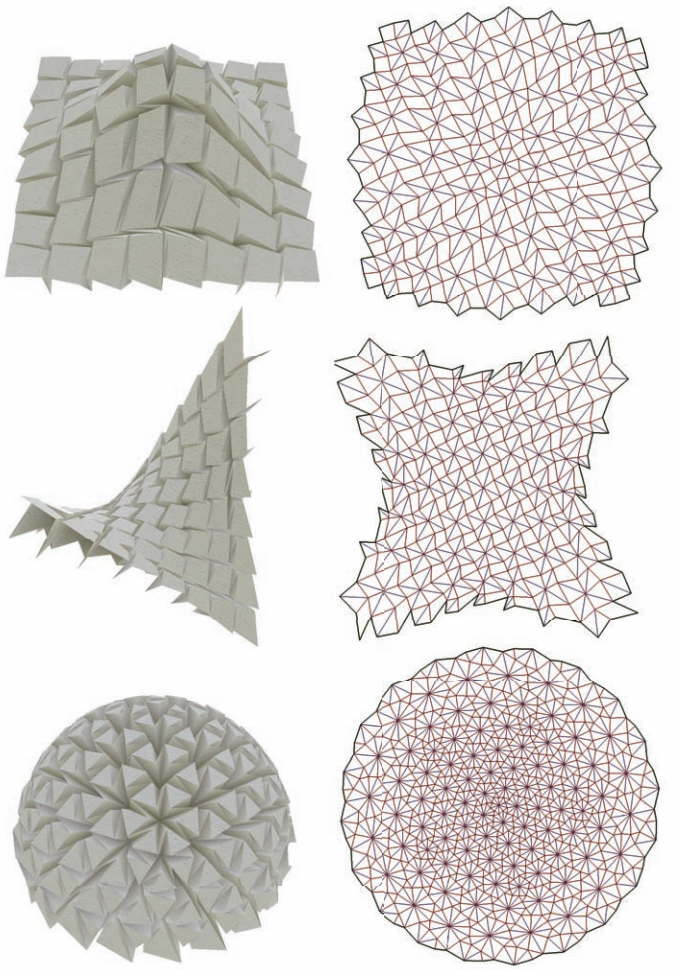

FIGURE 13. Example designs of bell shape, hyperbolic surface (anticlastic), and spherical surface (synclastic) from star-tuck origami tessellations.

DESIGN

Figures 13 and 14 show example designs of freeform origami tessellations approximating double-curved surfaces in a halfway-folded state, which are made possible for the first time with the proposed method. The results demonstrate the flexibility in the design of the origami tessellations.

There is a drawback that the optimization process does not guarantee the convergence to a valid solution; it fails to obtain a valid configuration without intersection when the initial polyhedral surface is "far" from a developable surface.. Figure 15 shows an example that fails to yield a valid origami tessellation. Here, the tuck structures are too large at the boundary, and the intersections between facets are no longer avoidable. This implies that the problem is not specific to our proposed method, but is generally attributed to the fact that a non-developable surface is

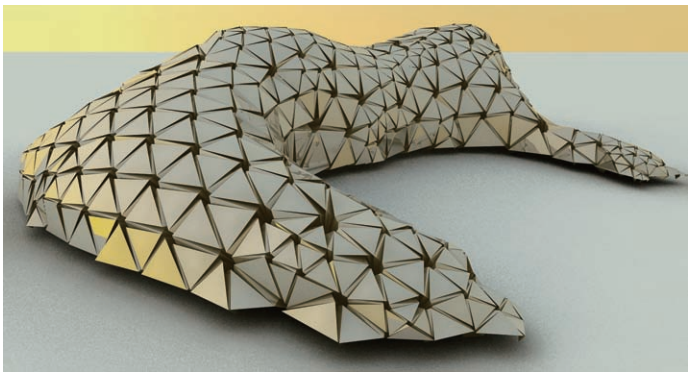

FIGURE 14. Blobby surface realized from a single developable surface.

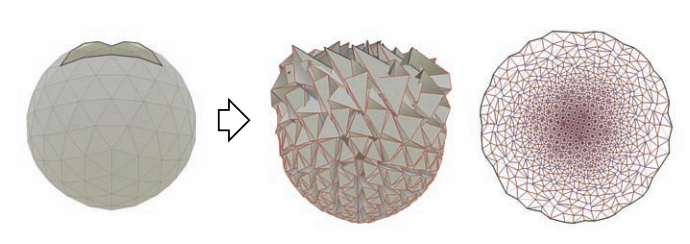

FIGURE 15. Spherical origami tessellation failing to avoid intersection

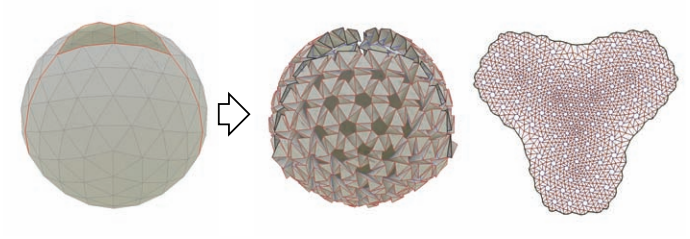

FIGURE 16. Spherical origami with cuts.

realized by virtually shrinking the surface with the cost of accumulating the size of the tuck. Such an accumulated error can be reduced by appropriately adding cuts to the initial polyhedron, and thus by locating the position of the boundary of the paper (Figures 16 and 17).

FABRICATION

The resulting origami tessellations can be physically fabricated by first grooving or perforating a sheet of material along the crease pattern generated as a vector data using 2-axis CNC machines such as a cutting plotter and laser cutter, and then folding the sheet along the marked foldlines (Figure 18). When the material does not allow for a π folding, we can design the origami tessellations by modifying the intersection avoidance condition for edge-adjacent facets by using the limit of $\pi - \delta$ instead of π.

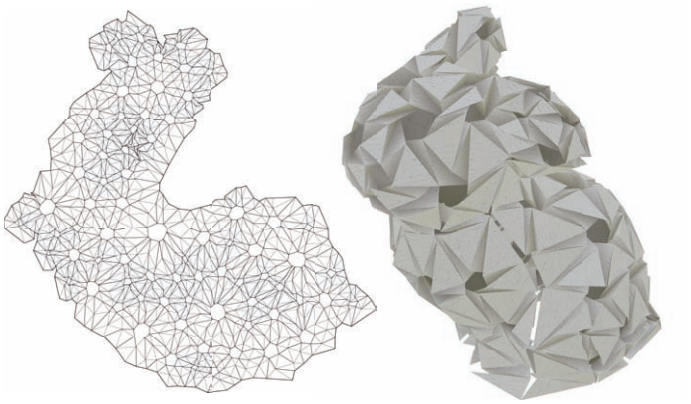

FIGURE 17. Tessellated origami bunny using the initial cut of the mesh.

The developability condition does not generally ensure the existence of a continuous folding motion from a planar sheet to a folded 3D form without the stretch of the material or the relocation of the creases; such a folding mechanism without the deformation of each facet is termed rigid origami. The resulting origami tessellation forms a rigid origami with multiple degrees of freedom when non-triangular facet is triangulated because the number of degrees of freedom of generic triangular mesh homeomorphic to a disk is calculated as $E_o - 3$, where E_o is the number of edges on the boundary of the mesh [13], whereas the configuration space is potentially disconnected because of the local and global collisions between facets, in which case, the folding from a planar state to the 3D form does not exist.

We checked this continuity of folding using *Rigid Origami Simulator* by unfolding the resulting 3D form to a planar sheet; some resulting patterns were successfully unfolded to a planar sheet without local and global collision (Figure 19), whereas other patterns encountered a global collision of facets. Even though we were not able to characterize the continuity of rigid origami folding motion, the patterns from our method have far better manufacturability than the ones from Origamizer method, which is completely locked and cannot even fold infinitesimally in a folded state. The results suggest that the method is potentially applicable to the manufacturing of an arbitrary 3D form from a hard metal sheet or panels.

CONCLUSION

We presented an approach for the design of freeform variations of Resch-like origami tessellations from a given polyhedral sur-

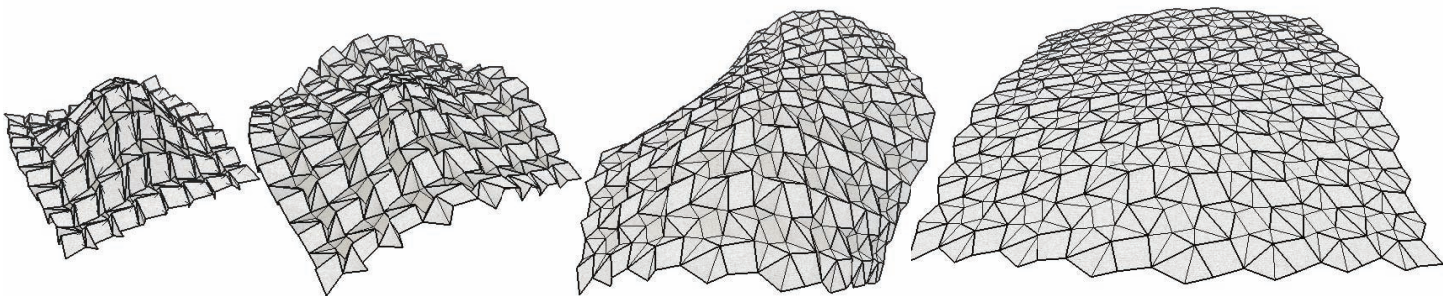

FIGURE 19. Continuous unfolding motion from a 3D form to a planar sheet.

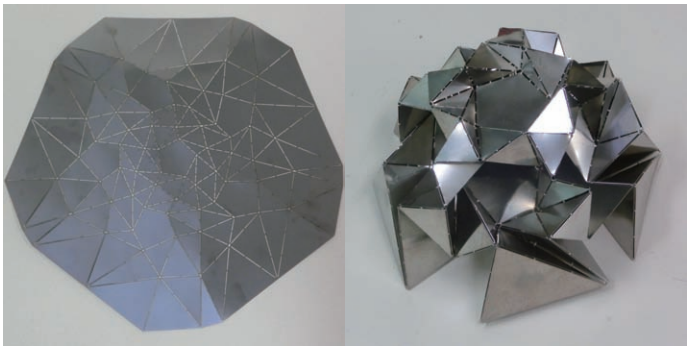

FIGURE 18. Example folding of a perforated steel sheet.

face. We presented the concept of star tucks and the variational tuck structures to be inserted between polygonal facets to construct a variation of tessellated surfaces. Such a generated surface is then optimized to make the surface developable and also non-intersecting at the vertices. We showed a penalty function for robustly calculating the intersections between vertex-adjacent facets. The method results in novel designs of freeform origami tessellations that neither Origamizer nor Freeform Origami could achieve.

ACKNOWLEDGMENT

This work was supported by the JST Presto program.

REFERENCES

[1] Tachi, T., 2010. "Origamizing polyhedral surfaces". *IEEE Transactions on Visualization and Computer Graphics*, ***16***(2), march, pp. 298–311.

[2] Fujimoto, S., 1976. "*Souzousei wo kaihatsu suru rittai origami" (in Japanese)*. Hyougo-ken Gakkou Kouseikai Tamba Shibu.

[3] Miura, K., 1970. "Proposition of pseudo-cylindrical concave polyhedral shells". In Proceedings of IASS Symposium on Folded Plates and Prismatic Structures.

[4] Davis, E., Demaine, E. D., Demaine, M. L., and Ramseyer, J., 2013. "Reconstructing david huffman's origami tessellations". In Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

[5] Resch, R. D., 1968. Self-supporting structural unit having a series of repetitious geometrical modules. United States Patent No. 3,407,558.

[6] Resch, R. D., and Christiansen, H., 1970. "The design and analysis of kinematic folded plate systems". In Proceedings of IASS Symposium on Folded Plates and Prismatic Structures.

[7] Tachi, T., 2010. "Freeform variations of origami". *Journal for Geometry and Graphics*, ***14***(2), pp. 203–215.

[8] Bateman, A., 2002. "Computer tools and algorithms for origami tessellation design". In Origami3: Proceedings of the 3rd International Meeting of Origami Mathematics, Science, and Education, pp. 121–127.

[9] Lang, R. J., and Bateman, A., 2011. "Every spider web has a simple flat twist tessellation". In Origami^5, P. Wang-Iverson, R. J. Lang, and M. Yim, eds., CRC Press, pp. 455–473.

[10] Sheffer, A., and de Sturler, E., 2001. "Parameterization of faceted surfaces for meshing using angle-based flattening". *Engineering with Computers*, ***17***(3), pp. 326–337.

[11] Streinu, I., and Whiteley, W., 2005. "Single-vertex origami and spherical expansive motions". In Lecture Notes in Computer Science 3742, pp. 161–173.

[12] Cantarella, J. H., Demaine, E. D., Iben, H. N., and O'Brien, J. F., 2004. "An energy-driven approach to linkage unfold-

ing”. In Proceedings of the 20th Annual Symposium on Computational Geometry, pp. 134–143.

[13] Tachi, T., 2009. “Simulation of rigid origami”. In Origami^4: The Fourth International Conference on Origami in Science, Mathematics, and Education, R. Lang, ed., A K Peters, pp. 175–187.