symmetry

MDPI

Article

A Computational Design Method for Tucking Axisymmetric Origami Consisting of Triangular Facets

Yan Zhao [1,*], Yuki Endo [2], Yoshihiro Kanamori [3] and Jun Mitani [3]

1 School of Computer Science and Communication Engineering, Jiangsu University, 301 Xuefu Road, Zhenjiang 212013, China
2 Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi 4418580, Japan; endo@val.cs.tut.ac.jp
3 Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba 3050006, Japan; kanamori@cs.tsukuba.ac.jp (Y.K.); mitani@cs.tsukuba.ac.jp (J.M.)
* Correspondence: yanzhao_cs@ujs.edu.cn; Tel.: +86-177-6649-1505

Received: 14 September 2018; Accepted: 2 October 2018; Published: 10 October 2018

Abstract: Three-dimensional (3D) origami, which can generate a structure through folding a crease pattern on a flat sheet of paper, has received considerable attention in art, mathematics, and engineering. With consideration of symmetry, the user can efficiently generate a rational crease pattern and make the fabricated shape stable. In this paper, we focus on a category of axisymmetric origami consisting of triangular facets and edit the origami in 3D space for expanding its variations. However, it is difficult to retain the developability, which requires the sum of the angles around each interior vertex needing to equal 360 degrees, for designing origami. Intersections occur between crease lines when such a value is larger than 360 degrees. On the other hand, blank spaces (unfolded areas) emerge in the crease pattern when the value is less than 360 degrees. The former case is difficult to generate a realizable shape due to the crease lines are intersected with each other. For the latter case, however, blank spaces can be filled with crease lines and become a part of the origami through tucking. Here, we propose a computational method to add flaps or tucks on the 3D shape, which contains non-developable interior vertices, for achieving the resulting origami. Finally, on the application side, we describe a load-bearing experiment on a stool shape-like origami to demonstrate the potential usage.

Keywords: computer-aided design; computational origami; axisymmetric structure; tuck folding

1. Introduction

Origami, also known as paper folding, has a long history, which may have existed since the invention of paper. Origami can be defined by a crease pattern that contains a set of mountain folded lines and valley folded lines appearing on a paper when a piece of origami is opened flat [1]. The mechanism of producing a 3D structure from a flat sheet through folding without cutting has aroused considerable research interest in art, mathematics, and engineering. Studying and applying efficient algorithms for solving paper-folding problems has become a branch of computer science that is called "computational origami" [2].

Origami has a developable surface with G^1 discontinuity. Differential geometry provides a detailed mathematical description of developables: assuming piecewise curvature-continuity, it consists of ruled surfaces with the additional property that the tangent plane along a ruling is constant [3]. Fundamental types of developable surfaces include the plane, the generalized cone, the generalized cylinder, and the tangent developable to a space curve [4]. The developable surface

can be bent to lie flat on the plane without tearing or stretching. Benefiting from this property, these surfaces can be achieved by thin material like paper, sheet metal, cardboard, and plywood [5]. Developable surfaces play an important role in both theory and practice; however, it is difficult for them to model arbitrary freeform shapes, especially those surfaces with dominant elliptic regions, due to the Gaussian curvature needing to be zero everywhere.

Recently, origami has taken to new heights. Several computational approaches for designing 3D origami have been proposed, and many appealing origami pieces have been generated [6–12]. In addition, the centuries-old origami has inspired many new applications, e.g., DNA origami [13], drug delivery [14], self-folding robots [15] and foldable solar panels [16–18]. However, the user cannot freely design an origami, due to underlying geometric constraints, e.g., developable constraints, limit of its design space. Lack of variations of origami could also restrict further research and application.

Based on symmetric properties, Zhao et al. proposed an interesting design method [19] for a category of axisymmetric 3D origami consisting of triangular facets. Under the symmetric properties, the user can efficiently generate a rational crease pattern and make the fabricated shape stable. An overview of this method is shown in Figure 1. First, the user designs the $1/N$ ($N = 8$ for this example) of the whole crease pattern (Figure 1a), where N indicates the order of rotational symmetry ($N > 2$) and the angle θ is determined by N, i.e., $\theta = 180^\circ/N$. Figure 1b shows the $1/N$ part in 3D space. $p_i (i = 0, 1, 2, \dots)$ and $p_i{}' (i = 2, 4, \dots)$ denote the points in the 2D crease pattern, and $P_i (i = 0, 1, 2, \dots)$ and $P_i{}' (i = 2, 4, \dots)$ indicate the corresponding points in 3D space (Figure 1c). P_0 at the origin of a Cartesian coordinate system. Π_1 is a vertical plane passing through the z-axis and y-axis. Π_2 is another vertical plane passing through the z-axis. Θ, which equals $180^\circ/N$, is an angle between such two vertical planes. P_i (with odd indices) lie on the plane Π_1 and P_i (with even indices) lie on the plane Π_2. $P_i^{'}$ and P_i (with even indices) are symmetric with respect to plane Π_1. The geometry can be calculated from its crease pattern together with a user-specified angle φ between edge P_0P_1 and the z-axis. Finally, the user can fabricate the origami piece as he/she designed (Figure 1d). We focus on this method because the resulting origami pieces are geometrically appealing due to the symmetric properties, and can be potentially used to fold an origami dome or applied in developable architectures.

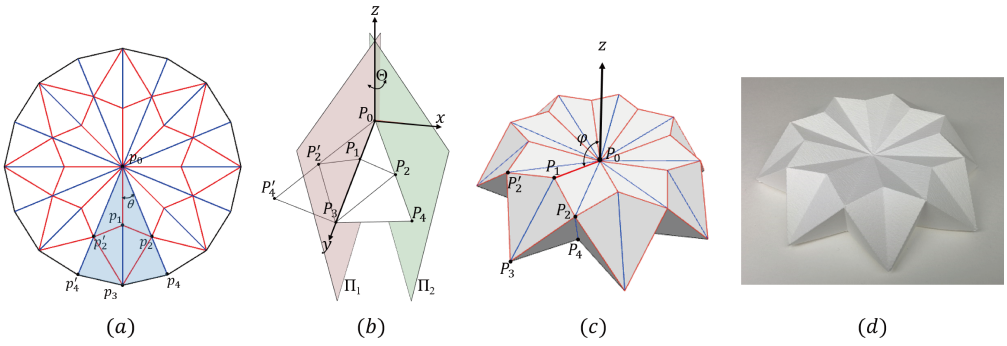


Figure 1. An overview of the method [19]: (**a**) a whole crease pattern; (**b**) a $1/N$ part of the crease pattern in 3D space; (**c**) calculated geometry; (**d**) a fabricated origami piece.

For designing an origami, each interior vertex should be developable, which means that the sum of the angles around each interior vertex equals 360 degrees. We call these vertices with zero angle deficit (e.g., p_1 and p_2 as shown in Figure 1a). These developable conditions for vertices can be easily satisfied in the case of [19], in which the 3D shape is generated by editing its crease pattern with symmetric properties.

In this paper, we edit the kind of origami [19] by moving its vertices in 3D space for expanding its variations. An overview of our method is shown in Figure 2. First, we take one origami (e.g., Figure 2a) as an input and then edit the shape by moving its $P_i (i > 1)$. Mountain and valley folded lines are rendered in red and blue, respectively. Those mountain lines are also known as ridges in the differential geometry literature [20]. We render the crease lines connected by two almost flat facets in

green. P_i (with odd indices) can be moved along plane Π_1 and P_i (with even indices) can be moved along plane Π_2.

Hereafter, we set i as four and move P_4 along plane Π_2 (Figure 2b). During the editing process, the crease pattern is updated by recalculating the location of p_4 with considering the distance constraints between P_iP_{i-1} and P_iP_{i-2} (Figure 2c). Note that it is inevitable to generate non-developable interior vertices in the crease pattern during this process. The sum of the angles around the interior vertex could be larger than or less than 360 degrees in the updated crease pattern. Intersections occur between crease lines when such a value is larger than 360 degrees. We call these patterns invalid. On the other hand, blank spaces (unfolded areas), colored in gray in Figure 2c, emerge in the crease pattern when the sum of the angles around interior vertex p_2 is less than 360 degrees. It is difficult for the former case to generate a realizable shape due to the crease lines intersecting with each other. For the latter case, however, blank spaces can be filled with crease lines and become a part of the origami through tucking.

For achieving the resulting origami, we propose a computational method to add flaps or tucks on the 3D shape, which contains non-developable interior vertices. We add crease lines in blank spaces to let the vertices become developable (e.g., p_2 and p_4 as shown in Figure 2c). Then, we calculate flaps outside (Figure 2d) or tucks inside, which are folded from such areas. By adding flaps or tucks on the edited 3D shapes, we make such shapes realizable through tucking the crease pattern without cutting those blank spaces.

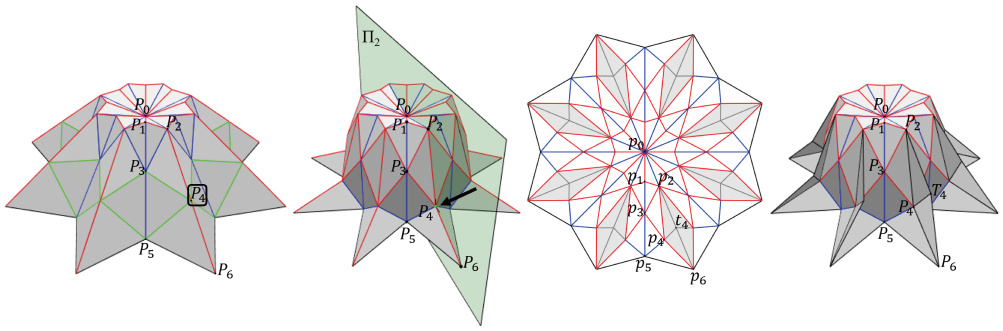


Figure 2. An overview of our method: (**a**) an input origami shape generated by [19]; (**b**) 3D editing to generate variations; (**c**) handling updated crease pattern for tucking; (**d**) the resulting 3D origami with flaps outside.

The main contributions of this paper are three-fold. First, although the user of our previous method [19] must edit the crease pattern to design a shape, the user in this method can manipulate 3D vertex positions directly in a WYSIWYG manner. In other words, our previous/new methods employ forward/inverse designs of crease pattern, respectively. Second, while our previous method cannot handle the crease pattern with blank spaces, our new method can by adding appropriate crease lines with consideration of symmetry. Third, by using our new method, novel origami pieces with flaps or tucks are achieved through tucking. The added shapes could support the whole structure; therefore, we conducted a load-bearing experiment on a stool shape-like origami to demonstrate the potential usage.

2. Related Work

Benefiting from the developing mathematical understanding, more and more complex and detailed origami pieces emerged. In addition, designing an origami becomes more efficient with the help of the computer-aided-design software, using which the user can preview the final shape.

TreeMaker is software used to design flat-foldable origami [21]. Its basic concept was first introduced by Meguro [22] and fully described by Lang [23]. This software generates the crease

pattern from a graph tree that represents the base structure of the object. *Tess* is a computer program that can make crease patterns for origami tessellations involving twist folds in a repeating pattern [24]. These approaches focus on flat-foldable origami, but we are aiming at making 3D origami.

The *Origamizer* algorithm by Tachi [6] is a general approach that can generate a crease pattern for an arbitrary 3D triangle mesh model with a topological disc condition. Demaine and Tachi fully described the algorithm in [25]. The approach is based on the tucking technique, a technique to hide the unnecessary areas of a sheet of paper inside the shape. The algorithm places the triangles on the plane with some margins that are "tucked" inside. Our approach is similar to this. While Tachi [6] utilizes a numerical optimization, our approach is based on a simple analytical formula because of the symmetrical property of the target shape. Although the Origamizer can handle our target shapes, the generated pattern tends to be overly complicated because adding tucks *inside* is the one and only solution. However, our system adds tucks *outside* (we call them "flaps") if adding tucks inside is not a simple solution. Modeling freeform surfaces is nontrivial due to its very diverse nature. Bartoň et al. used circular arc splines [26], sweeps of planar profiles [27] for approximating general free-form geometry. Tachi [10,11] proposed design systems for constructing freeform surfaces that can be manufactured by folding a sheet of paper.

Mitani proposed a method for designing 3D origami based on a rotational sweep [7,8] that generates a simpler crease pattern for an axisymmetric structure by adding flaps or 3D structures outside of the target shape. Although the outer flaps might be considered obtrusive, his methods succeed in generating 3D curved origami from simple crease patterns. His other method [9], which combines the advantages of the rotational sweep and mirror reflection approaches, has been used to build geometrically attractive origami pieces.

Zhao et al. [19] proposed a method for handling a family of axisymmetric 3D origami consisting of triangle facets. The user can intuitively design a rotationally-symmetric crease patterns and then preview the axisymmetric 3D shape. Then, Zhao et al. [28] proposed a method for designing axisymmetric 3D origami with generic six-crease bases. Developability can be satisfied benefiting from the mirror-symmetry of the crease pattern. The method also presented a rigid folding motion that can axisymmetrically deploy or flatten the 3D shape. By developable deformations of many small faces connected through fold lines, origami can approximate intrinsically curved surfaces. The resulting origami can have "global" Gaussian curvature without stretching or shrinking of the flat sheet [4]. In particular, Dudte et al. [29] used modified Miura cells to approximate orientable 3D surfaces with positive, zero, negative, and mixed Gaussian curvatures. Zhao et al. [30] provided a solution for approximating 3D surfaces with varying curvatures using generalized waterbomb tessellations.

The interactive system is a creative and explorative tool for designers in computer-aided design modeling and in 3D origami design. Mitani and Igarashi's system [12] allows the user to design 3D curved origami surfaces by using mirror operation with selecting and moving a vertex on the 3D origami while maintaining the developability of the resulting shape. The proposed system is also interactive. When the user edits the 3D origami by moving its vertices, the crease pattern is automatically updated. Then, flaps outside or tucks inside are added if they are needed to hold geometric consistency.

Applying origami-concepts has lead to lots of applications in various areas. Kuribayashi et al. [31] proposed an origami stent graft for biomedical application. Zirbel et al. [17,18] proposed a method to build a large solar array for space application. The family of axisymmetric 3D origami consisting of triangular facets proposed by [19] can be potentially used to fold an origami dome or applied in a developable architecture. This paper reports a load-bearing experiment we conducted on a stool shape-like origami with tucks inside demonstrating the potential usage of our origami piece.

3. Methodology

3.1. Designing 3D Origami

First, we introduce our user interface for editing origami in 3D space. Second, we describe a calculation of flaps folded from quadrilateral blank spaces. Finally, we describe a general case for calculating flaps that are folded by polygonal blank spaces.

We take one origami (e.g., Figure 3a) as an input. Such a shape is generated by [19]. By editing the vertices in 3D space, we can explore new variations which could not be achieved by the previous method. Specifically, the origami is edited by moving its P_i (with odd indices) along plane Π_1 and P_i (with even indices) along plane Π_2. Figure 3a shows the origami to be edited by moving its P_4 along plane Π_2. P_4 can be moved along the u direction (Figure 3b or Figure 3c) and the v direction (Figure 3d or Figure 3e) on the plane Π_2. As shown in Figure 3, the u-direction is parallel to x–y plane and the v-direction is parallel to the z-axis. Because the u- and v-directions are orthogonal, P_4 can be flexibly moved on plane Π_2 by moving along such two directions repeatedly.

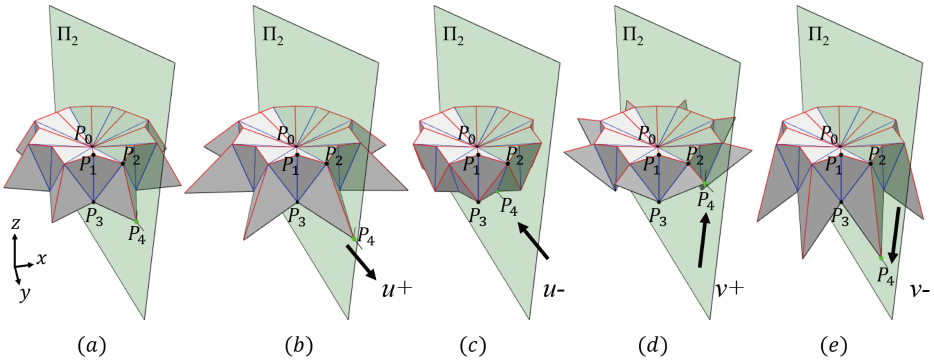


Figure 3. Designing origami in 3D space: (**a**) origami to be edited by moving P_4 along plane Π_2; (**b**) moving P_4 along the $u+$ direction; (**c**) moving P_4 along the $u-$ direction; (**d**) moving P_4 along the $v+$ direction. (**e**) moving P_4 along the $v-$ direction.

During the design process, the system recalculates its crease pattern to keep it congruent with the newly edited shape. Specifically, for the shape shown in Figure 3e, we calculate the new position of p_4 (Figure 4a) that makes the distance between p_4p_3 and p_4p_2 the same as the distance between P_4P_3 and P_4P_2, respectively. In particular, p_4 is achieved as a crossover vertex of two circles whose centers are p_2 and p_3, respectively. Although p_4'' also satisfies such distance constraints, we leave such selection because it generates an invalid crease pattern with intersections. The updated crease pattern is shown in Figure 4b and the detail around interior vertex p_2 is shown in Figure 4c where the angle $a_{i,k}$ denotes the k-th incident sector angle of p_i. Note that the sum of the angles around interior vertex p_2 is less than 360 degrees, and hence the blank spaces shown in gray emerge (Figure 4b).

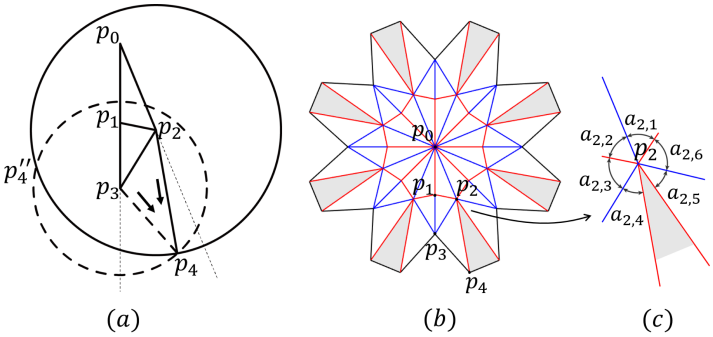


Figure 4. Recalculating crease pattern: (**a**) determining new position of p_4; (**b**) updated crease pattern; (**c**) details of angles around p_2.

Editing an origami in 3D space can hardly retain zero angle deficit for each interior vertex. Figure 5 shows the updated crease patterns corresponding to the 3D editing shown in Figure 3, respectively. The new position of p_4 (Figure 5b,d) makes the sum of the angles around p_2 larger than 360 degrees, thereby leading to invalid crease patterns due to intersections. By checking the sum of the angles around an interior vertex, our system can give feedback to the user when an invalid crease pattern is generated. When the sum of the angles around p_2 less than 360 degrees, blank spaces without crease lines are emerged in the crease pattern (Figure 5c,e) hindering us from folding.

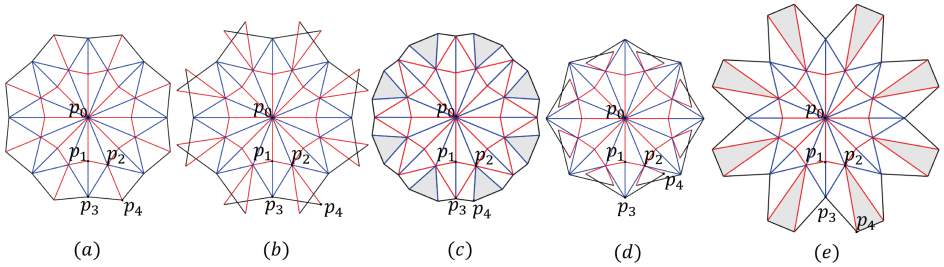


Figure 5. Updated crease patterns corresponding to the 3D editing shown in Figure 3, respectively. (**a**) the original crease pattern before editing; (**b**) an invalid crease pattern due to self-intersections; (**c**) a crease pattern with blank spaces; (**d**) another invalid crease pattern with self-intersections; (**e**) another crease pattern with blank spaces.

For folding the 3D shape with non-developable interior vertices using one piece of paper, we need to tackle those blank spaces. *Origamizer* [6,25] proposed the tucking technique, which hides unnecessary areas of a sheet of paper inside the shape using a numerical optimization. Here, by using the rotationally-symmetric feature of the crease pattern, blank spaces can be efficiently handled. Consider the crease pattern in Figure 6a and its part shown in Figure 6b, where the blank space is quadrilateral. We first add a crease line between p_2 and p_6 to divide the blank space $p_2p_4{}'p_6p_4$ equally. Under the symmetry context, the p_2p_6 can make the edges $p_2p_4^{'}$ and $p_4^{'}p_6$ coincide with p_2p_4 and p_4p_6, respectively, by folds. Then, we add a new vertex t_4 along segment p_2p_6 with two crease lines t_4p_4 and $t_4p_4^{'}$ to fold the blank space. Note that t_4 takes any position on the segment p_2p_6. Next, we calculate the shape of a flap or tuck in 3D space by calculating the coordinates of T_4, whose distances to P_2, P_4, and P_6 are $|p_2t_4|$, $|p_4t_4|$, and $|p_6t_4|$, respectively. Therefore, T_4 lies on the intersection points of three spheres whose centers are P_2, P_4, and P_6 and radius equal $|p_2t_4|$, $|p_4t_4|$, and $|p_6t_4|$, respectively.

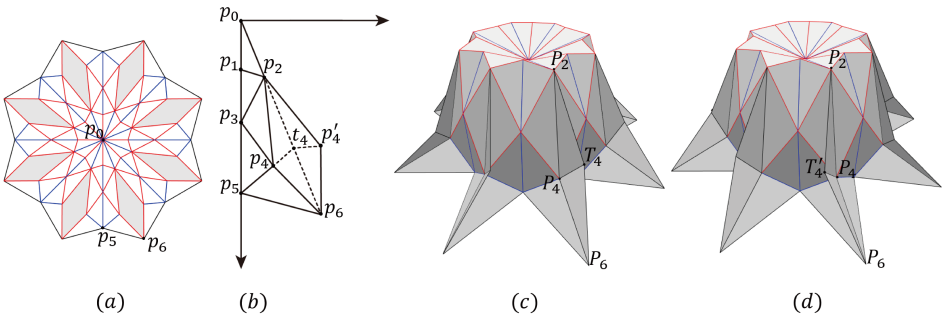


Figure 6. Calculating 3D shape of flap: (**a**) a crease pattern with quadrilateral blank spaces; (**b**) handling the blank space with consideration of symmetry; (**c**) one resulting 3D origami with flaps; (**d**) the other candidate with flaps having different orientation.

If two intersection points exist, our system allows the user to choose either of them as the solution of T_4 to obtain different resulting 3D origami (Figure 6c,d). Our system gives feedback when the flaps cannot be calculated. Here, the user is required to manually adjust the crease pattern for calculating flaps, such an interface can be improved in the future.

We can also explore various flaps by moving t_4 along the segment p_2p_6. Figure 7a shows one scenario where three choices of t_4, i.e., $t_{4,1}, t_{4,2}, t_{4,3}$ and the resulting 3D origami are shown in Figure 7b–d, respectively. Some location of t_4 could make the generated 3D flaps penetrate facets in the resulting 3D origami (Figure 7b). In such a situation, the user changes t_4 to another location (e.g., $t_{4,2}$ or $t_{4,3}$) until no penetration occurs (if such a location exists). By interactively editing possible 3D flaps, the user can find and revise the invalid location of t_4 in the design process. Using a computational way to add crease lines without causing penetrations is left as future work.

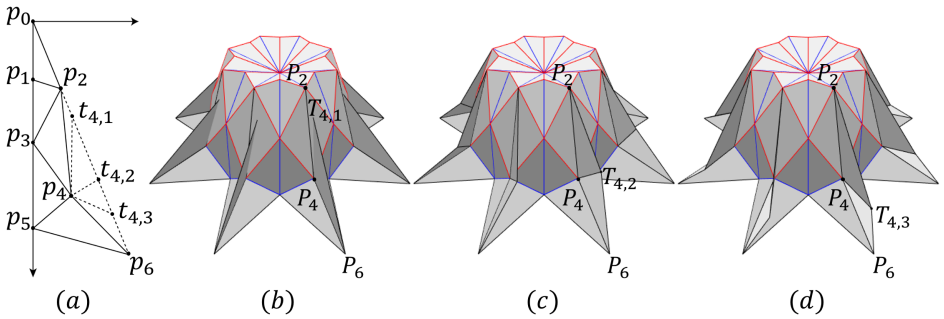


Figure 7. Variations of the 3D flaps. (**a**) Exploring various flaps by moving t_4 along the segment p_2p_6; (**b**) Resulting origami by selecting $t_{4,1}$; (**c**) Resulting origami by selecting $t_{4,2}$; (**d**) Resulting origami by selecting $t_{4,3}$.

Finally, we describe a general case for calculating flaps that are folded by polygonal blank spaces. As shown in Figure 8, (a) shows a edited 3D origami and (b) shows its crease pattern whose blank spaces are polygons. We first equally divide the polygonal area by adding a crease line p_3p_9. Then, we divide such area into triangles by adding t_5 and t_7 along segment p_3p_9 as shown in (c). Finally, we calculate the 3D coordinates of T_5 with the distance constraints from P_3, P_5, and P_7. In the same way, T_7 is calculated by considering the distance constraints from T_5, P_7, and P_9. Because the points are positioned inside the shape, the blank space area is folded inside as a tuck, as shown in (d). This process is also applicable for general polygons that have more than six vertices.

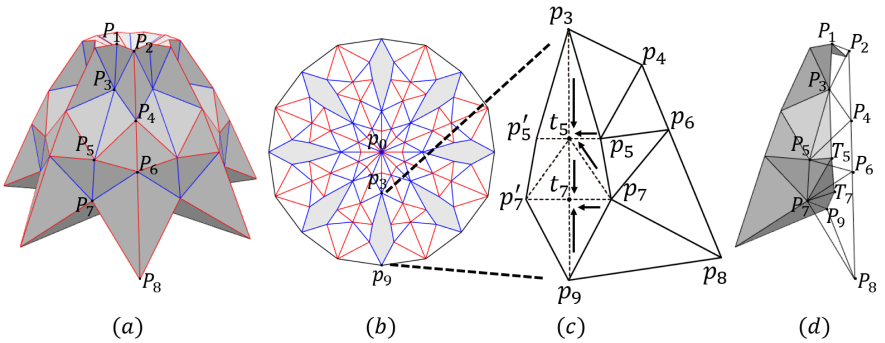


Figure 8. Calculating 3D flaps folded from polygonal blank spaces: (**a**) edited 3D origami; (**b**) the crease pattern whose blank spaces are polygons; (**c**) handling the blank space; (**d**) the calculated tuck inside the shape.

3.2. Special Case of 3D Flaps

On the basis of the aforementioned explanation, the method adds flaps outside or tucks inside the origami. In this section, we describe a special case where 3D flaps lie exactly on the surface of the origami. Figure 9 shows one example of such a special case. P_5, the middle vertex of P_4 and P_4', is on the plane Π_1 because P_4 and P_4' are symmetric to the plane Π_1. In addition, P_5 makes $P_3P_5P_4$ a 90-degree angle. P_7, which is another middle vertex of P_6 and P_6', makes $P_5P_7P_6$ a 90-degree angle. P_5, P_4, P_6, and P_7 are coplanar because line P_4P_4' is parallel to line P_6P_6'. Next, we add appropriate

crease lines in the blank space as shown in Figure 9b and the part shown in Figure 9c. Angles $p_3p_5t_5$ and $p_5p_7t_7$ should equal 90 degrees to let flaps lie exactly on the surface. Therefore, we specify t_5 by extending line p_4p_5 and line $p_4{}'p_5{}'$. Similarly, t_7 is specified by extending line p_6p_7 and line $p_6{}'p_7{}'$. Under such a configuration, the 3D flaps become a part of the origami whether they are viewed from inside or outside (Figure 9a).

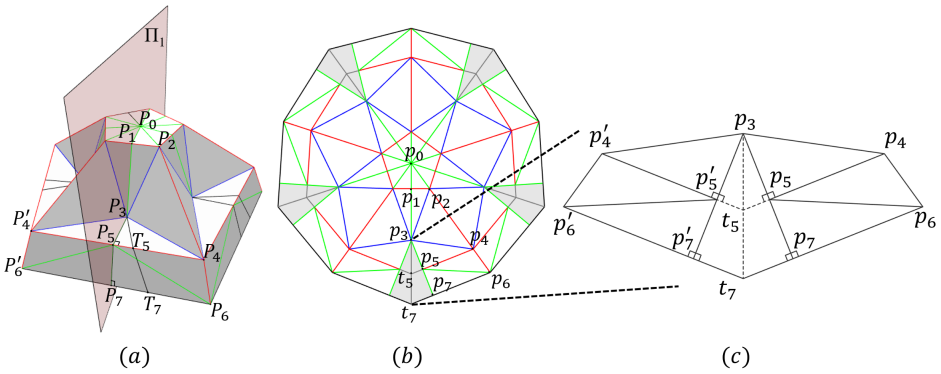


Figure 9. A special case of 3D flaps: (**a**) calculated flaps on the surface of the 3D origami; (**b**) handling the blank space; (**c**) a part of the crease pattern.

4. Results and Discussion

We show several resulting origami pieces in this section. As shown in Figure 10, the first column is the crease pattern, the second column is the 3D shape, and the third column is a photo of the origami piece. In the left side, (a) shows the flaps lie outside of the origami and (b) shows a similar shape but with a flat center area; (c) shows the tucks inside the origami; (d) shows the origami piece whose flaps lie exactly on itself. In the right side, (e) and (f) illustrate the origami pieces with open and closed blank spaces. In particular, (e) has a flat center area and (f) consists of triangular facets. (g) and (h) show two origami pieces with the shape of a stool.

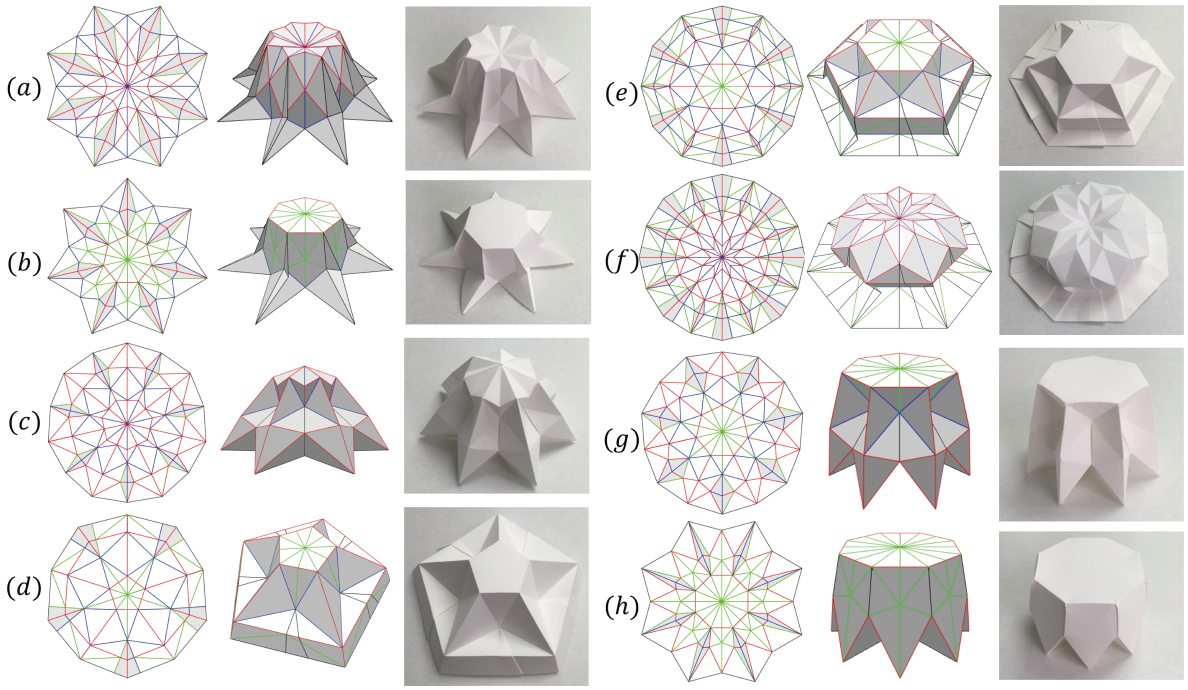
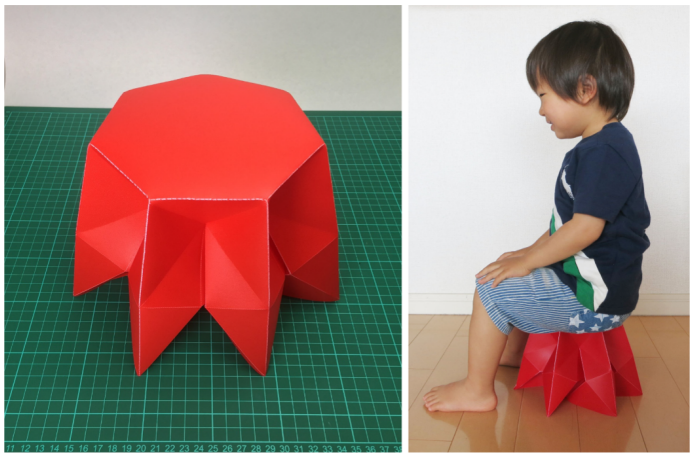


Figure 10. Resulting origami pieces with 3D flaps or tucks.

These origami pieces are locked by tucks and thus do not easily open at the bottom when we put pressure on their top surfaces. This feature could be potentially used as a stool for sitting.

Therefore, we fabricated the origami piece (Figure 10g) using polypropylene with 0.75 mm thickness to demonstrate its potential usage. The length and width of the material used in this experiment were about 50 cm. We made all crease lines on the top surface of the material using a cutting plotter. Then, to valley fold smoothly, we manually made valley lines on the other side of the material. We took almost two hours to fold the material. As a result, we obtained an origami stool (Figure 11a) with 22 cm length and width and 15 cm height without gluing. Furthermore, we found that a two-year-old boy of 13 kg could sit on it (Figure 11b).

Beyond observation of this experiment, we found that bending and distortions occurred in some facet of the origami stool. The main reason could be that we did not consider designing an origami piece with material thickness. Revising the crease pattern to adapt for thick material is left as future work.



(a) (b)

Figure 11. Load bearing experiment on stool shape-like origami with tucks inside.(**a**) An origami stool using polypropylene with 0.75 mm thickness; (**b**) A two-year-old boy of 13 kg could sit on the stool.

5. Conclusions

We focused on a category of axisymmetric origami consisting of triangular facets and edited the shape by moving its vertices in 3D space for expanding variations. We demonstrated that it is inevitable to generate non-developable interior vertices, which can generate blank spaces (unfold areas) in the crease pattern. Symmetry was taken into consideration in our computational procedure for tucking those blank spaces into flaps or tucks. By adding those extra shapes, we achieved the resulting origami that can be folded from one piece of paper without cutting. We expanded the variations of origami and demonstrated several new origami pieces with flaps or tucks. Finally, we did a load-bearing experiment on a stool shape-like origami to demonstrate the potential usage.

As future work, three aspects of our study can be improved: (i) using a computational way to add crease lines in the blank spaces without causing penetrations, (ii) revising crease patterns with considering the thickness of material, and (iii) exploring an optimal shape of tucks or flaps to increase the strength of the origami structure. We hope our demonstration of axisymmetric 3D origami can help motivate further advances in the origami-inspired research and application.

Author Contributions: Y.Z. and J.M. proposed and conceived this approach; Y.Z. wrote the paper; Y.E. and Y.K. proofread the paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mitani, J. A method for designing crease patterns for flat-foldable origami with numerical optimization. *J. Geom. Graph.* **2011**, *15*, 195–201.
2. Demaine, E.D.; Demaine, M.L. Recent results in computational origami. In *Origami 3: Third International Meeting of Origami Science, Mathematics and Education*; A K Peters/CRC Press: Boca Raton, FL, USA, 2002; pp. 3–16.
3. Tang, C.; Bo, P.; Wallner, J.; Pottmann, H. Interactive design of developable surfaces. *ACM Trans. Graph.* **2016**, *35*, 12. [CrossRef]
4. Callens, S.J.; Zadpoor, A.A. From flat sheets to curved geometries: Origami and kirigami approaches. *Mater. Today* **2018**, *21*, 241–264. [CrossRef]
5. Solomon, J.; Vouga, E.; Wardetzky, M.; Grinspun, E. Flexible developable surfaces. *Comput. Graph. Forum* **2012**, *31*, 1567–1576. [CrossRef]
6. Tachi, T. Origamizing polyhedral surfaces. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 298–311. [CrossRef] [PubMed]
7. Mitani, J. A design method for 3D origami based on rotational sweep. *Comput.-Aided Des. Appl.* **2009**, *6*, 69–79. [CrossRef]
8. Mitani, J. A Design Method for Axisymmetric Curved Origami with Triangular Prism Protrusions. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*; CRC Press: Boca Raton, FL, USA, 2011; pp. 437–447.
9. Mitani, J. Column-shaped Origami Design Based on Mirror Reflections. *J. Geom. Graph.* **2012**, *16*, 185–194.
10. Tachi, T. Freeform variations of origami. *J. Geom. Graph.* **2010**, *14*, 203–215.
11. Tachi, T. Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. *Adv. Archit. Geom.* **2010**, *2010*, 87–102.
12. Mitani, J.; Igarashi, T. Interactive design of planar curved folding by reflection. In Proceedings of the Pacific Conference on Computer Graphics and Applications, Kaohsiung, Taiwan, 21–23 September 2011; pp. 77–81.
13. Nangreave, J.; Han, D.; Liu, Y.; Yan, H. DNA origami: A history and current perspective. *Curr. Opin. Chem. Biol.* **2010**, *14*, 608–615. [CrossRef] [PubMed]
14. Zhang, Q.; Jiang, Q.; Li, N.; Dai, L.; Liu, Q.; Song, L.; Wang, J.; Li, Y.; Tian, J.; Ding, B.; et al. DNA origami as an in vivo drug delivery vehicle for cancer therapy. *ACS Nano* **2014**, *8*, 6633–6643. [CrossRef] [PubMed]
15. Felton, S.; Tolley, M.; Demaine, E.; Rus, D.; Wood, R. A method for building self-folding machines. *Science* **2014**, *345*, 644–646. [CrossRef] [PubMed]
16. Miura, K. Map fold a la Miura style, its physical characteristics and application to the space science. *Res. Pattern Form.* **1994**, pp. 77–90.
17. Zirbel, S.A.; Lang, R.J.; Thomson, M.W.; Sigel, D.A.; Walkemeyer, P.E.; Trease, B.P.; Magleby, S.P.; Howell, L.L. Accommodating thickness in origami-based deployable arrays. *J. Mech. Des.* **2013**, *135*, 111005. [CrossRef]
18. Zirbel, S.A.; Trease, B.P.; Thomson, M.W.; Lang, R.J.; Magleby, S.P.; Howell, L.H. HanaFlex: A large solar array for space applications. In *Micro- and Nanotechnology Sensors, Systems, and Applications VII*; International Society for Optics and Photonics: Washington, DC, USA, 2015; Volume 9467, p. 94671C.
19. Zhao, Y.; Kanamori, Y.; Mitani, J. Geometry of Axisymmetric 3D Origami Consisting of Triangular Facets. *J. Geom. Graph.* **2017**, *1*, 107–118.
20. Ohtake, Y.; Belyaev, A.; Seidel, H.P. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.* **2004**, *23*, 609–612. [CrossRef]
21. Lang, R.J. TreeMaker. Available online: http://www.langorigami.com/article/treemaker/ (accessed on 10 September 2018).
22. Meguro, T. The method to design origami. *Origami Tanteidan Newspaper*, 10 September 1991.
23. Lang, R.J. A computational algorithm for origami design. In Proceedings of the Twelfth Annual Symposium on Computational Geometry, Philadelphia, PA, USA, 24–26 May 1996; pp. 98–105.
24. Bateman, A. Paper Mosaic Origami Tessellations. Available online: http://www.papermosaics.co.uk/software.html (accessed on 10 September 2018).
25. Demaine, E.D.; Tachi, T. Origamizer: A Practical Algorithm for Folding Any Polyhedron. 2017; Volume 77. Available online: http://drops.dagstuhl.de/opus/volltexte/2017/7231/(accessed on 10 September 2018).

26. Bartoň, M.; Shi, L.; Kilian, M.; Wallner, J.; Pottmann, H. Circular arc snakes and kinematic surface generation. *Comput. Graph. Forum* **2013**, *32*, 1–10. [CrossRef]
27. Bartoň, M.; Pottmann, H.; Wallner, J. Detection and reconstruction of freeform sweeps. *Comput. Graph. Forum* **2014**, *33*, 23–32. [CrossRef]
28. Zhao, Y.; Kanamori, Y.; Mitani, J. Design and motion analysis of axisymmetric 3D origami with generic six-crease bases. *Comput.-Aided Geom. Des.* **2018**, *59*, 86–97. [CrossRef]
29. Dudte, L.H.; Vouga, E.; Tachi, T.; Mahadevan, L. Programming curvature using origami tessellations. *Nat. Mater.* **2016**, *15*, 583–588. [CrossRef] [PubMed]
30. Zhao, Y.; Endo, Y.; Kanamori, Y.; Mitani, J. Approximating 3D surfaces using generalized waterbomb tessellations. *J. Comput. Des. Eng.* **2018**, *5*, 442–448. [CrossRef]
31. Kuribayashi, K.; Tsuchiya, K.; You, Z.; Tomus, D.; Umemoto, M.; Ito, T.; Sasaki, M. Self-deployable origami stent grafts as a biomedical application of Ni-rich TiNi shape memory alloy foil. *Mater. Sci. Eng. A* **2006**, *419*, 131–137. [CrossRef]