



Visualization of Folding Motion of Rotationally Symmetric Curved Folding

Yuka Watanabe¹  and Jun Mitani² 

¹University of Tsukuba, ywatanabe@cgg.cs.tsukuba.ac.jp

²University of Tsukuba, mitani@cs.tsukuba.ac.jp

Corresponding author: Yuka Watanabe, ywatanabe@cgg.cs.tsukuba.ac.jp

Abstract. We propose a graphical user interface (GUI) system to generate and visualize the folding motion of rotationally symmetric curved folding. The system supports the ruling transition during the folding motion by recalculating the ruling directions in accordance with the crease shape and the folding angle in every frame. To work with curved folding with many creases, the user edits the shape of one segment while the rest of the paper is cloned in accordance with the symmetric property of the crease pattern. The folding motion is generated by the interpolation of the key frames edited by the user.

Keywords: Curved Folding, Developable Surface Modeling, Folding Motion.

DOI: <https://doi.org/10.14733/cadaps.2020.513-524>

1 INTRODUCTION

Rotationally symmetric curved folding is a type of origami that has curved creases placed in rotational symmetry, and the folded sheet of paper forms a three-dimensional shape (Figure 1). Various paper artworks of this type have been created by many artists and hobbyists, and software to design the 3D shape and the crease pattern has been developed [7]. Although the crease patterns and their final folded 3D shapes are well known, their folding motions, or the paper shapes in between the flat unfolded state and the final folded state, are not clear. To help to clarify the folding motion of such paper artworks, our system generates and visualizes the paper shape in the intermediate state, with some user manipulation and the key frame interpolation. Clarifying the folding motion will help the users to learn how to fold the curved folding efficiently, which is sometimes difficult for beginners. It may also be useful in designing a new crease pattern by examining its foldability. Moreover, the folding motion must be understood for automating the folding, such as RoboFolds [1]. In some applications, the paper shape in the intermediate state is useful by itself, such as lamp shades and bowl-shaped containers.

One factor making the folding motion unclear is the ruling transition. Rulings are straight thin strips on a developable curved surface perpendicular to the bending directions of the paper, shown as pink and purple lines in the figures. Unlike the classical paper folding with straight crease lines,

the curved folded paper consists of rulings that transit as the paper is being folded, and the bending direction of the curved surface changes gradually. In our previous research, we proposed a method for interactively designing a curved folding considering its folding motion [15],[16]. In the work, the ruling transition is simulated by calculating the ruling directions in every frame, using the parameters of the crease curve: 2D or 3D curvature, the torsion, and the folding angle. This method allows the user to interactively design one primary curved crease and additional curved creases. The primary crease is first designed in accordance with the parameters described above. Then the additional crease is drawn on the crease pattern, approximated by a B-spline curve, and refined by the user by moving its control points. The user can change the parameters or the positions of the control points and check the result interactively until achieving an intended shape, with no intersection between the rulings on the paper or self-collision that does not occur in the real world. However, we found that dealing with multiple additional creases is a delicate process. As the user adds more creases, even a small displacement of the curve easily causes the rulings to intersect. In response to such problems, we propose a new framework by narrowing down the target to a specific type of crease pattern with rotational symmetry. The paper is modeled as a group of identical segments with only one crease, placed in rotational symmetry, as shown in Figure 1-middle row. By editing one segment, the rest of the paper is cloned in accordance with the symmetric property of the crease pattern. Consequently, the curved folding with several creases is modeled on the basis of our previous method, as each segment contains only a small number of creases. Moreover, by dividing the crease pattern into segments, the method becomes able to model the curved folding including straight creases. Mathematically, a straight crease is only able to bend or have a curved shape only in a flat or completely folded state. In other words, the surface adjacent to a straight crease should always be a flat surface in the intermediate state. Our model supports straight creases in the curved folding by allowing a small gap on the creases located between the segments. In this work, we assume that the paper shape is rotationally symmetric at all stages of the folding motion, though in the real world it may temporarily be asymmetric.

In the rest of this paper, related work about the paper modeling is first introduced in section 2, followed by the description of our method in section 3, the evaluation of the method and the numerical verifications in section 4, and the conclusions in section 5.

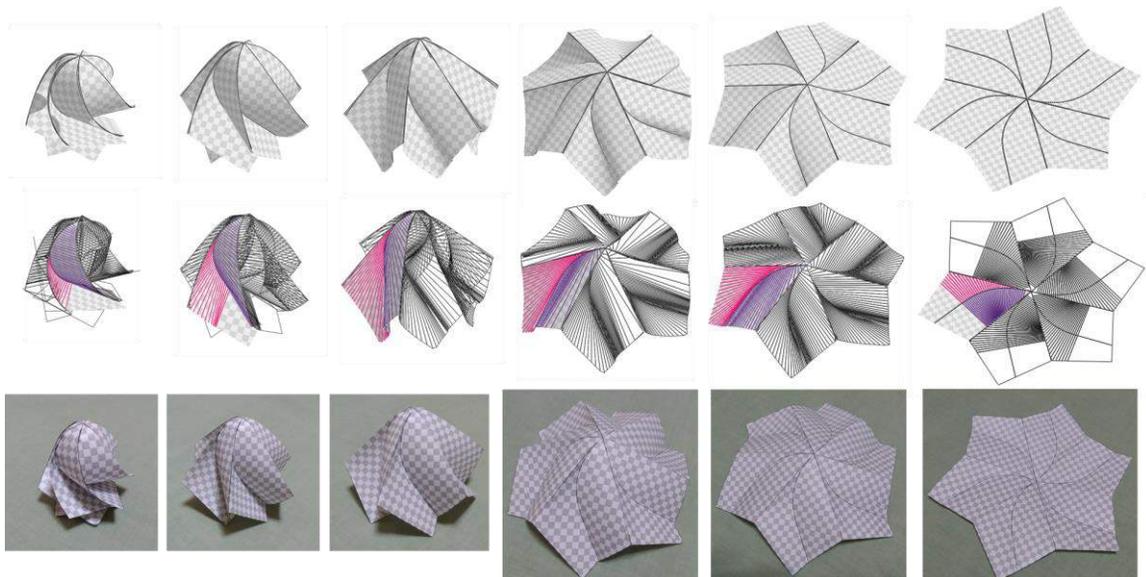


Figure 1: Transformation of origami-sphere between folded and unfolded states. Top: rendered 3D models. Middle: wireframe models with one segment rendered with rulings. Bottom: photos of real paper.

2 RELATED WORK

In the area of paper modelling and simulation, 3D surfaces have often been represented by a polygon mesh of fixed geometry composed of a group of quad strips and planar polygons. One early work of modeling a curved folded surface is the method proposed by Kilian et al. [4]. Their method reconstructs a 3D polygon mesh of a developable surface from input 3D data, such as point clouds, of a curved folding, by initially fitting planar polygons on the 3D shape and then optimizing their shapes and orientations for its developability and the connectivity. In contrast to this robust but time-consuming process, Mitani and Igarashi proposed a simple user interface to design multi-crease curved folding by folding the curved surface along planer curves using the reflective principle of the rulings [8]. Both methods successfully generate a developable surface of a user-intended shape but do not design or consider its folding motion. In the field of paper folding simulation, Tachi developed software to simulate the folding motion of the paper using the rigid folding method [9]. The method represents the paper by planar polygons of fixed geometry hinged at the edges, mainly targeting the paper folded on straight lines. He has also proposed some methods to design flat-foldable vault structures composed of flat-foldable tubes assembled by welding two sheets [10],[11]. Although the structure is folded along the curve, the folding angles are restricted to be constant throughout the curve, which ensures the rulings are fixed so that the structure can be modeled with the rigid folding method. However, it does not support any deformation where the ruling directions on the surface change during the folding motion.

In the most recent research on modeling developable surface with curved folding, various models are proposed to simulate more flexible motion of the paper, unrestricted except for its developability. Narain et al. proposed a method to simulate a sheet of thin materials by adaptive mesh refinement that could model irregular deformations such as crumpled paper [6]. Kilian et al. also take an approach of re-meshing the triangular mesh in accordance with the local surface curvatures in modeling the 3D shape of paper in string actuated folding motion [5]. Their methods successfully represent various shapes, but resulting mesh data may contain thousands of faces, not necessarily suitable for observing the rulings. Ghassaei et al. implemented a fast and interactive web application for origami simulation that works on a graphics processing unit (GPU) [3]. Their model is based on the constraints on edge length, folding angle, and the angle between the edges, allowing some stretching or shearing of the polygonal faces. The flexibility of the model enables the polygon shape to transit from the flat state to the folded state but does not consider the developability or the ruling transition in the intermediate state. Taking a more refined approach, Tang et al. defined a curved surface by a pair of spline curves and solved the constraints for its global developability through iterative processes [12]. Rabinovich et al. introduced discrete orthogonal geodesic nets, which approximate a smooth developable surface by optimizing the corner angles of the quad mesh to be orthogonal [13]. In their subsequent work, the model was refined to represent a curved folding by having two smooth curved surfaces intersect on an identical curved crease [14].

Partly similar but different from their works, our system models the curved folded surface by quad strips on the basis of the rulings, while supporting the ruling transition. The ruling directions are calculated instantly in accordance with the 2D and 3D shapes of the crease curve and the folding angles, changing the shape of the quads during the folding motion, keeping the mesh structure simple without subdivision. The system focuses on generating and visualizing the folding motion with the user manipulation, helping users to understand the movement of the paper, not by a physical simulation or a numerical calculation.

3 METHODS

In this section, we will explain the whole process by taking as an example the origami-sphere with six segments, which is a typical example of the rotationally symmetric curved folding designed by Mitani [7]. It is composed of two types of creases: the curved creases in a valley-fold (or concave fold) that maintain the sphere shape with the adjacent curved surfaces, and the straight radial creases in a mountain-fold (or convex fold) that pull out the extra parts of the paper not on the

sphere. We model the crease pattern as a set of segments divided by the radial line creases, and each segment contains one curved crease, as shown in Figure 1-Middle row. The following subsections describe the method to generate the shape and the motion of the paper, followed by the explanations of the graphical user interface (GUI) system.

3.1 Overall Process

Our system has a 3D model and a crease pattern. During the folding motion, the shape of the 3D model changes continuously, while the crease pattern, a set of creases in a flat state, stays unchanged. To generate the folding motion of the 3D model, the system starts with two key frames: the flat state and the final folded state. In the flat state, all creases have zero folding angle, i.e., the 3D model identical to the crease pattern. The final folded state is the complete form of the curved folding where, in this example, straight crease lines are folded completely and the curved surfaces bent by the curved creases make a sphere. Initially, the intermediate frames are generated by linearly interpolating the two key frames, as described in subsection 3.3 (Figure 2-top row). As this generally causes large gaps between the segments, the user picks one frame, corrects it to make the gap smaller, and adds it as a new key frame. The rest of the non-key frames are then re-interpolated (Figure 2-bottom row). This process is carried out until the user is satisfied with the output, where the segments are placed adjacently in all frames with no large gaps or major self-collision.

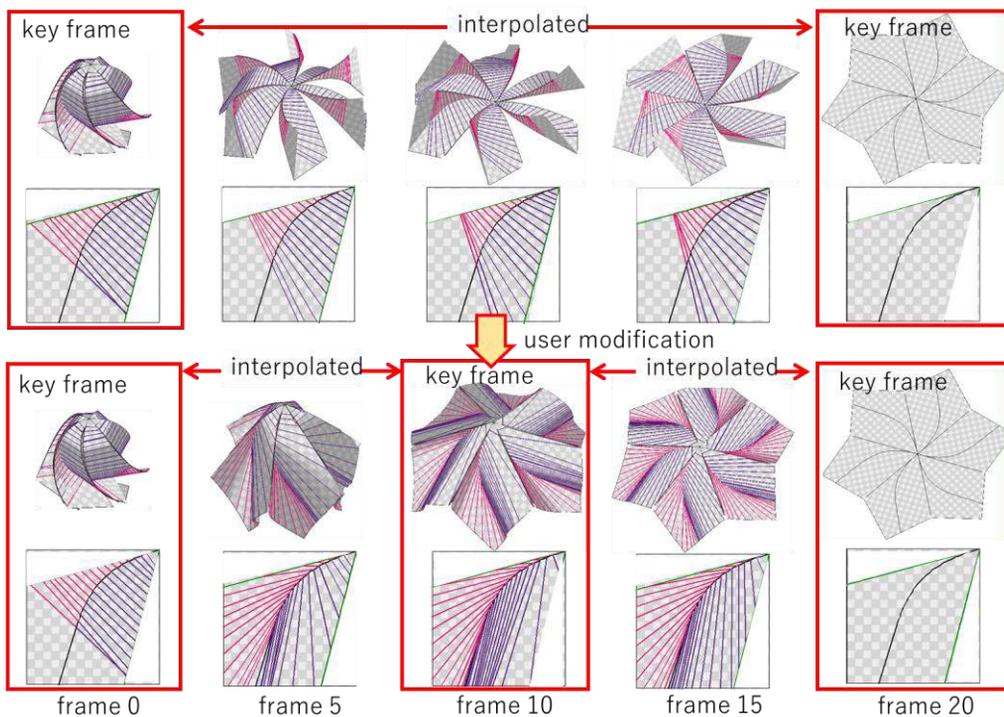


Figure 2: Overall process of folding motion generation. Top: initial state. Bottom: after one modification. 3D polygon models are the paper shapes generated by the system. 2D figure shows crease pattern of one segment of the 3D model.

3.2 Crease Pattern Generation

In this system, the crease pattern is the predefined data that may be designed by some other tools or methods. The whole crease pattern is composed of some identical segments placed side by side in rotational symmetry. The crease pattern of one segment consists of one curved crease in the center and two boundary creases on its sides, which are the boundaries between the adjacent segments. Theoretically, a segment may contain multiple curved creases that are not on the boundary, but our system supports only one crease for the usability. As in our previous works [15],[16], the curved crease is derived from 2D curvature values set on some control points, or the representative points sampled at equal intervals on the curve. They are interpolated throughout the curve and define the shape of the 2D curve (Figure 3). The placements of two boundary creases are determined by the division number N of the rotational symmetric design, which is equal to the number of the segments (Figure 4-left). Their shapes are defined by a constant value of 2D curvature k_{2D} (Figure 4-right). Whereas the origami-sphere has the boundary curves in straight line shape, our system allows them to have a curved shape but defines them by only one value to make the implementation simple. The two congruent boundary creases are placed with a rotation of $2\pi/N$ so that, in the flat state, the segments are placed with no gap. Figures 1 and 2 show an example with $N = 6$ and $k_{2D} = 0.0$, which means the boundary creases are the straight lines.

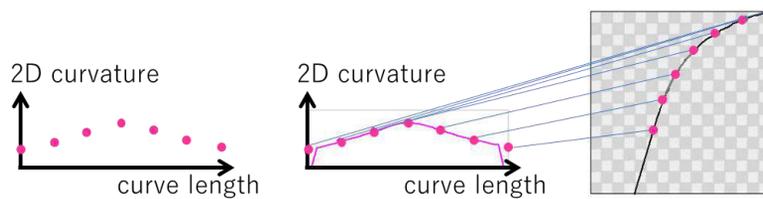


Figure 3: Curved crease on a segment. Left: 2D curvatures on the representative points plotted on a graph. Center: 2D curvature interpolated from the seven representative points. Right: curved crease generated from the 2D curvature and seven representative points plotted on the curve.

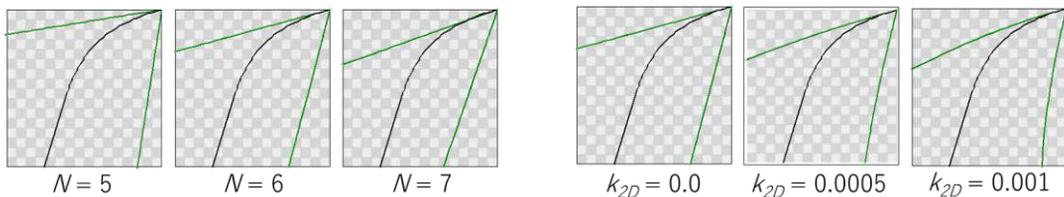


Figure 4: Variations of boundary crease. Left: boundary creases in different numbers of segments N . Right: boundary creases with different curvatures k_{2D} and $N = 6$.

3.3 3D Model Generation and Interpolation

The process of 3D model generation and interactive modification is applied to the key frames added by the user and to the final folded state. In this system, as the 2D crease pattern is given, the shape of a segment is defined by the torsion and the folding angle of the curved crease, as described later. The whole 3D model is then constructed with multiple segments placed in rotational symmetry with small gaps in between the adjacent boundary creases. In other words, the process of modifying or refining the 3D model is composed of two steps: (i) the 3D shape refinement of a segment, and (ii) the adjustment of the segment pose, or the 3D orientation of the segment. For the final folded state, the parameters of the curved crease and the segment pose should be given by other tools or

methods and, if necessary, refined on this system. The initial values of other key frames are given by interpolating key frames and then modified by the user.

In (i) the 3D shape refinement, the shape is controlled by the folding angle and the torsion of the curved crease. As in our previous works [15],[16], the parameters on the control points on the curved crease are given and adjusted by the user and then interpolated throughout the curve, like in the process of crease pattern generation. Using Fuchs and Tabachnikov's equations [2] with each parameter described in Figure 5 parameterized by the arc length s ,

$$k_{2D}(s) = k(s) \cos \alpha(s) \quad (3.1)$$

$$\cot \beta_L(s) = \frac{-\alpha(s)' + \tau(s)}{k(s) \sin \alpha(s)} \quad (3.2)$$

$$\cot \beta_R(s) = \frac{\alpha(s)' + \tau(s)}{k(s) \sin \alpha(s)} \quad (3.3)$$

the 3D curvature $k(s)$ is first calculated from the folding angle $\alpha(s)$ and the 2D curvature $k_{2D}(s)$ by equation (3.1). Then the 3D shape of the curved crease is calculated from the 3D curvature $k(s)$ and the torsion $\tau(s)$. The ruling angles on the left and right sides of the crease $\beta_L(s), \beta_R(s)$ are derived by equations (3.2) and (3.3). Finally, the directions of the rulings in 3D space $r_L(s), r_R(s)$ are calculated using the ruling angles $\beta_L(s), \beta_R(s)$, tangential vector $\mathbf{T}(s)$, the normal vector $\mathbf{N}(s)$, the binormal vector $\mathbf{B}(s)$ of the 3D crease curve, and the folding angle $\alpha(s)$, to define the whole 3D shape of the segment. In our new system, the 3D shape is also derived by giving the ruling angles $\beta_L(s), \beta_R(s)$ in a 2D crease pattern. An example of a modified segment shape is shown in Figure 6.

Then (ii) the segment pose is adjusted through an optimization process to minimize the gaps between the boundary creases, or by user adjustment through the mouse drag interface (Figure 7). In the optimization process, segments are placed so that their boundary creases fit the lateral edges of an equilateral pyramid, as shown in Figure 7-right. The shape of the equilateral pyramid is derived from the number of segments N and the relative 3D orientations of a pair of boundary creases on a segment. That is, the base of the pyramid has the same number of edges as the number of the segments, and the angle between the adjacent lateral edges of the pyramid is equal to the angles between the lines approximating the boundary crease in 3D space: passing through the center of the paper and its orientation calculated by the least square method using the sample points on the boundary crease. In the user adjustment process, as the pose of one segment is adjusted with a mouse drag, all other segments are also placed in rotational symmetry, or the i -th segment rotated $2\pi i/N$ around the vertical axis.

To integrate the segments to be one sheet of paper, (i) the target segment shape must have the 3D shapes of the two boundary creases to be as congruent as possible and (ii) the segment pose is to be adjusted to minimize the gaps, without major self-collision. (i) and (ii) are improved alternately until the user is satisfied with the result by visual check. After the key frame is modified, other interpolated frames are updated by calculating the folding angles, the torsion, and the pose by the linear interpolation of the key frames, followed by the same process as described above.

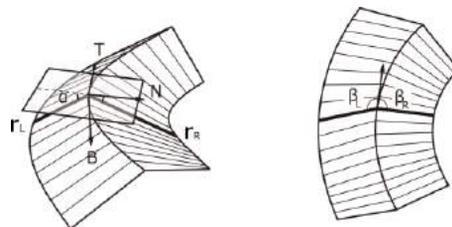


Figure 5: The definitions of folding angle α , ruling angles β_L, β_R , ruling vectors r_L, r_R , tangent vector \mathbf{T} , normal vector \mathbf{N} , and binormal vector \mathbf{B} . (Left: 3D space, Right: 2D space).

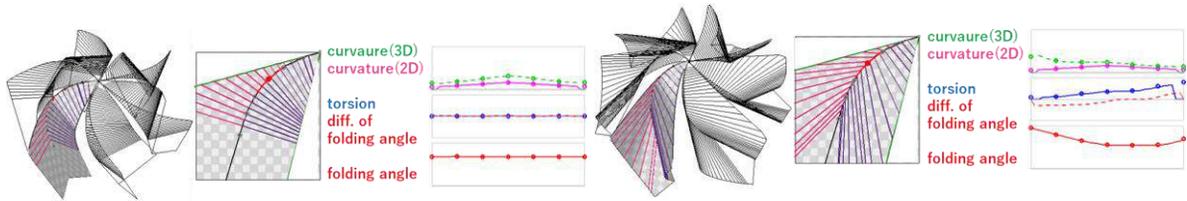


Figure 6: Modification of 3D shape of a segment. Left: Initial shape with constant folding angle and zero torsion. Right: Modified shape with folding angle and torsion adjusted by the user. From left and right in both sides: 3D model with one segment rendered with rulings, cease pattern of a segment, and parameter values along the curved crease shown in graphs.

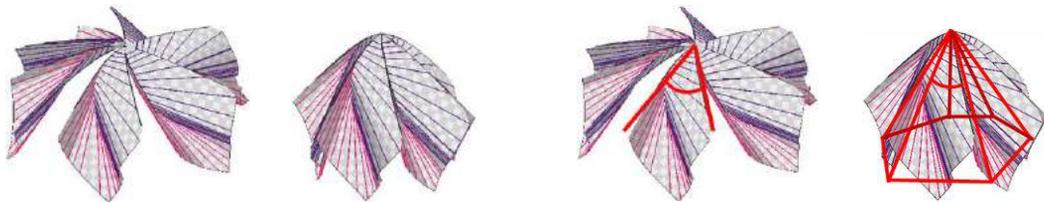


Figure 7: Adjustment of segment pose. From left to right: initial pose, optimized pose, initial pose with lines passing through boundary creases, optimized pose with boundary creases fit to lateral edges of equilateral pyramid.

3.4 GUI of the Prototype System

The proposed GUI system consists of four panes: the 3D model, the 2D crease of a segment, the parameters of the curved crease shown in the graphs, and the control panel (Figure 8). For the folding angles and the torsion, the graphs of the parameters in all frames are shown, with the current frame shown in thick colored lines, the key frames in thin colored lines, and the other interpolated frames in thin gray lines. For the other parameters, only the current frame is depicted. The control panel carries widgets such as scroll bars and the radio buttons to choose the segment number N , the current frame, parameters, etc. or to adjust the parameter values such as curvatures, torsions, and the folding angles.

Figure 9 shows the flowchart of the user operation. After the 2D and 3D creases of the final folded state are input, the user first defines the boundary curve by the method described in subsection 3.2. Then the user checks the generated shapes and modifies the 3D model if necessary, as described in subsection 3.3. The segment shape is refined by changing the torsion or the folding angle through the widgets while checking the generated shape visually on the screen. The pose is optimized by choosing the button or adjusted by mouse drag on the 3D model. When the 3D model in the frame is good enough, the user registers the frame as a new key frame and checks the folding motion by showing each frame specified by the scroll bar. The user picks a frame and modifies it until the 3D models in all frames are satisfactory, under a smooth deformation in the folding and the unfolding motion. As well as the shape of the 3D models, the transition of rulings and the parameters can also be observed through the 2D crease pattern and the graphs of the parameters.

4 EVALUATION

As an evaluation of our method, the 3D models of the rotational symmetric curved folding with the folding motion, generated by our prototype system, were evaluated in terms of (i) their developability, (ii) their connectivity, and (iii) a visual comparison with real paper. The examples used for the evaluation are listed in Table 1.

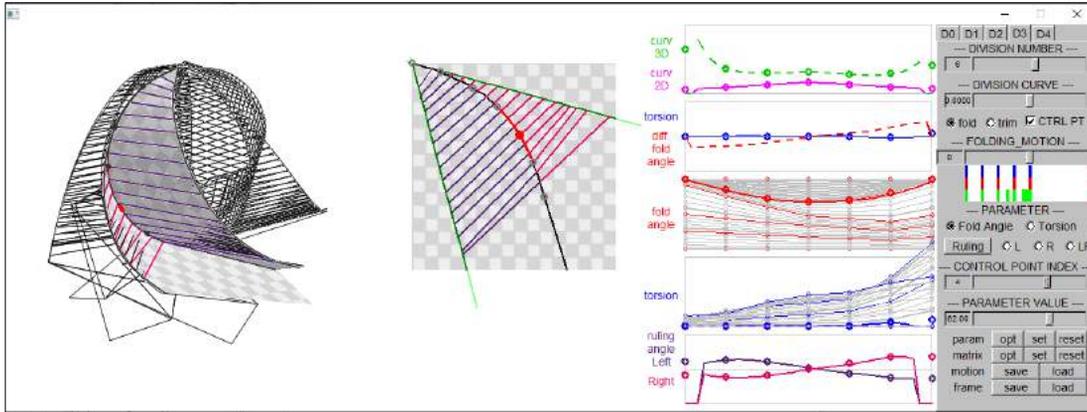


Figure 8: GUI of the prototype system.

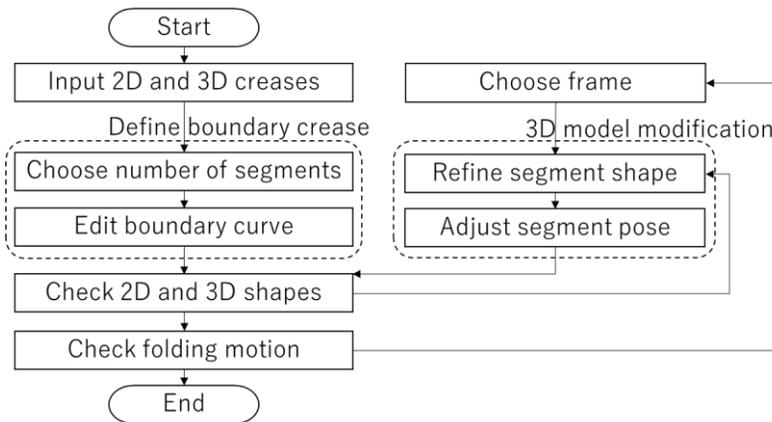


Figure 9: Flowchart of user operation.

Model Name	N6	N6_k1	N5	N8	N6_skew
Number of segments N	6	6	5	8	6
Curvature of boundary crease k_{2D}	0.0	0.001	0.0	0.0	0.0
Shape	sphere	sphere	sphere	sphere	skewed sphere

Table 1: 3D models used for evaluation.

4.1 Developability of a Segment

To evaluate the developability of the 3D model, the developability of a segment was evaluated by calculating (A) the sum of corner angles adjacent to each vertex on the curved crease and (B) the flatness of each quad composing the curved surface. Because the sum of corner angles is 2π around a vertex on a developable surface, (A) was evaluated as the error to 2π . For (B), the flatness of a quad was calculated as the distance between one vertex of a quad and the plane passing through the other three vertices. The units are in millimeters while the crease pattern of the 3D model used for the evaluation has a radius of approximately 300 mm. Table 2 shows the average and maximum of (A) and (B) of all vertices or faces over all frames. The results show that a segment is sufficiently

developable for all 3D models, though some errors are expected to be caused by discretizing the crease curve.

Model Name	N6	N6_k1	N5	N8	N6_skew
Average error of (A) [radian]	3.5×10^{-5}	1.2×10^{-5}	4.2×10^{-5}	4.4×10^{-5}	6.2×10^{-5}
Maximum error of (A) [radian]	3.47×10^{-4}	1.12×10^{-4}	2.45×10^{-4}	3.93×10^{-4}	4.92×10^{-4}
Average of (B) [mm]	7.19×10^{-3}	5.01×10^{-3}	7.1×10^{-3}	8.34×10^{-3}	1.09×10^{-2}
Maximum of (B) [mm]	1.19×10^{-1}	3.85×10^{-2}	7.46×10^{-2}	1.08×10^{-1}	6.69×10^{-2}

Table 2: Developability of a segment.

4.2 Connectivity of the Segments

The connectivity of the segments is calculated as the gap between the adjacent boundary creases in 3D space. To measure the gap between two curves, reference points are first sampled in an equal interval on one boundary crease. Then the distances from the reference points to the nearest edges on the other boundary crease are calculated. The average distance at all the sample points is evaluated as the gap. Because the segments are always placed in rotational symmetry in this system, the gaps are the same for all pairs of adjacent boundary creases. Figure 10 shows the gaps on each frame for the 3D models listed in Table 1, with frame 0 being the final folded state and frame 20 the flat state. The circles on the graph indicate the key frames where both the shape and the pose of the segments are modified by the user. The squares indicate the frames where only the segment pose is modified. The rest of the points are the interpolated frames. The results show that, by the user manipulation, the gaps are made to be below 2.5 mm for most frames, with the 3D models whose radiuses of the crease patterns are approximately 300 mm. We expect these gaps are small enough to be covered by a small deformation, such as the sliding of the creases, while processing with real paper. With more user effort, the gaps may be reduced.

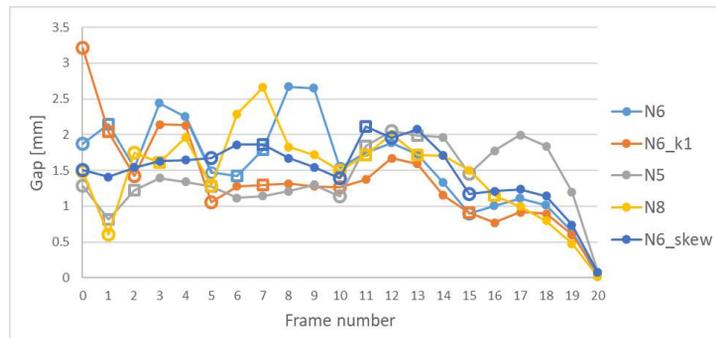


Figure 10: Connectivity of segments. Circles on the graph indicate key frames whose shape and pose are modified by the user. Squares indicate the frames where only the pose is modified.

4.3 Visual Comparison with Real Paper

As the visual comparison, the 3D models and the photos of the model *N6* are shown in Figure 1. For other models, the results are shown in Figure 11. The behavior of the paper shape and the rulings were visualized successfully, with only small differences between the photos. The 3D model shows that even if, for the final folded state, the angles between the rulings and a crease curve are designed to be nearly right angles, they become acute in the intermediate state. This makes most rulings end at the edge of the paper instead of the boundary crease, making the boundary creases have small curvatures while being folded.

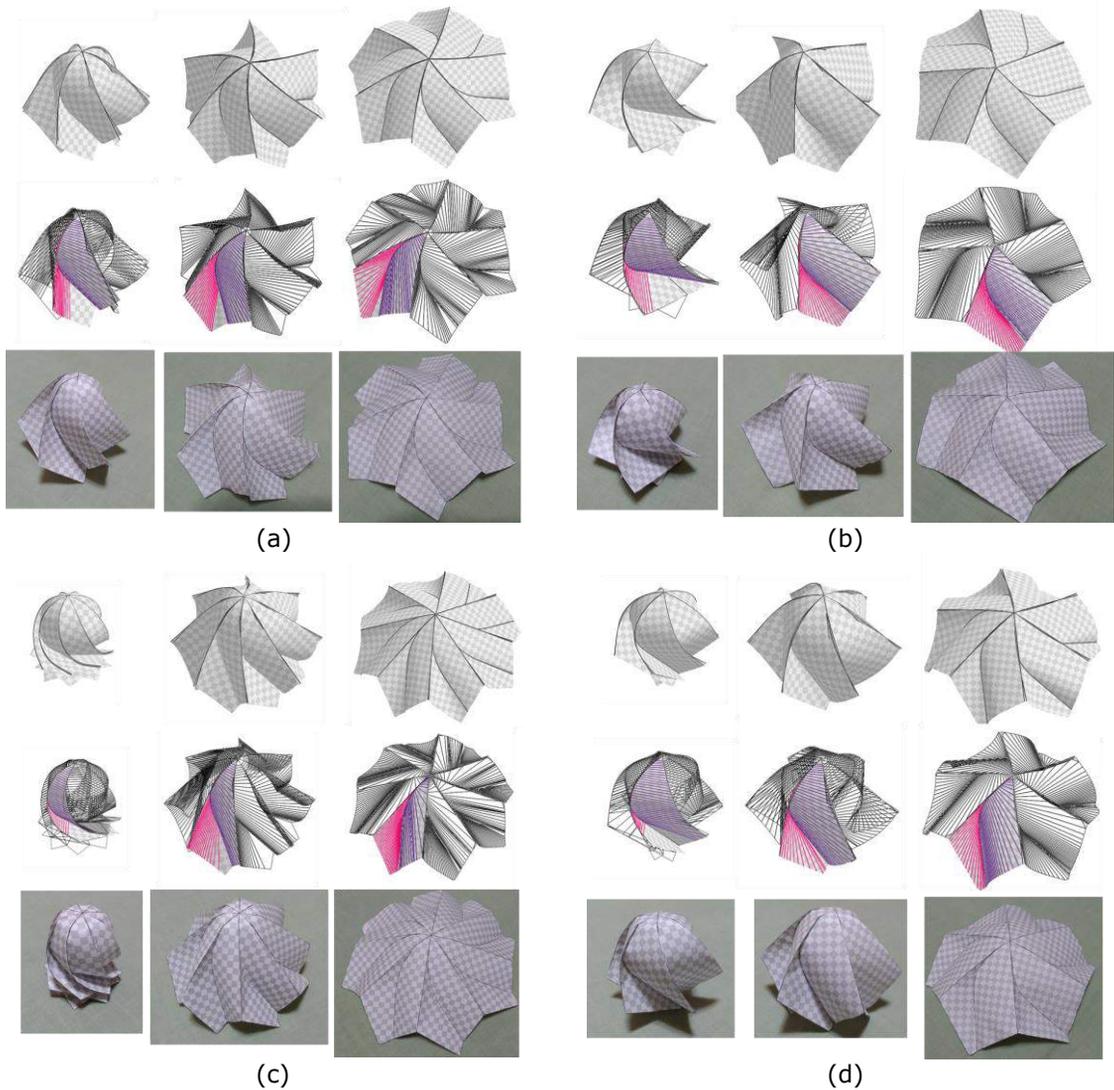


Figure 11: Comparison with real paper. (a) N6_k1. (b) N5. (c) N8. (d) N6_skew. For each 3D model, Top: rendered 3D models. Middle: wireframe models with one segment rendered with rulings. Bottom: photos of real paper.

4.4 Applications

As some examples of proposed applications, Figure 12 shows objects such as a bowl-shaped container, lamp shade, and unfoldable cup. They are designed as the paper shape of an origami-sphere in the intermediate state of the folding motion. The shapes of such containers may be controlled by the user, depending on the amounts they carry. For a lamp shade, by folding and unfolding, the luminous intensity may be controlled. The unfoldable cup is used in the folded state by fitting the deformable thin plastic to the frame and may be unfolded when washed or stored.

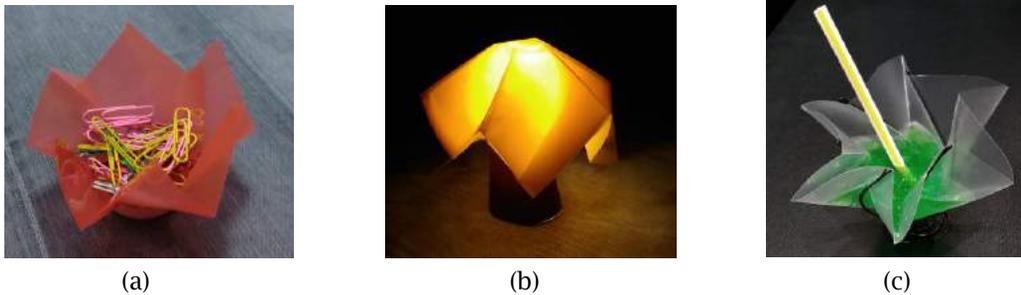


Figure 12: Some applications using 3D shapes of the intermediate frame of origami-sphere. (a) Container. (b) Lamp shade. (c) Unfoldable cup.

5 CONCLUSION

In this work, we proposed a GUI system to generate and visualize the folding motion of rotationally symmetric curved folding. The system models a sheet of paper with several curved creases, by having the user edit one segment and cloning the rest in accordance with the symmetry of the crease pattern. The evaluation shows that segments are modeled as developable surfaces, while there are small gaps between the segments. As mentioned in the introduction, mathematically, a straight crease line may have a curved shape in 3D space only in a flat or completely folded state. However, the behavior of the real paper shows a curved folding with straight creases, taking similar shapes with the 3D model generated by our method. This is thought to be realized by a small deformation of the paper, such as the compressions, stretching, and the shifting of the creases, which are similar to the model with small gaps between the segments.

As future work, the user interface to generate and modify the 3D shape of a segment should be improved, as well as having an optimization process added, because this process is still difficult and requires some user experiences. Evaluations with other crease shapes with more variation are also necessary. Moreover, as often observed with real paper, the system should be able to model the non-symmetric case of the folding motion, starting with the symmetric shape generated by this system.

Yuka Watanabe, <http://orcid.org/0000-0002-1362-1274>

Jun Mitani, <http://orcid.org/0000-0002-1596-844X>

REFERENCES

- [1] Epps, G.: Robofold and robots. IO, Made by Robots: Challenging architecture at a larger scale, Architectural Design, 2014, 68–69. <https://doi.org/10.1002/ad.1757>
- [2] Fuchs, D.; Tabachnikov, S.: More on paperfolding, The American Mathematical Monthly, 106(1), 1999, 27–35. <https://doi.org/10.2307/2589583>
- [3] Ghassaei, A.; Demaine E. D.; Gershenfeld, N.: Fast, interactive origami simulation using GPU computation, Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education, 2018, 1151–1166.
- [4] Kilian, M.; Flöry, S.; Chen, Z.; Mitra, N.; Sheffer, A.; Pottmann, H.: Curved folding, ACM Transactions on Graphics, 27(3), 2008, 75:1–9. <https://doi.org/10.1145/1360612.1360674>
- [5] Kilian, M.; Monszpart, M.; Mitra, N. J.: String actuated curved folded surfaces, ACM Transactions on Graphics, 36(3), 2017, 25:1–13. <https://doi.org/10.1145/3072959.3015460>
- [6] Narain, R.; Pfaff, T.; O'Brien, J. F.: Folding and crumpling adaptive sheets, ACM Transactions on Graphics, 32(4), 2013, 51:1–8. <https://doi.org/10.1145/2461912.2462010>
- [7] Mitani, J.: A design method for 3D origami based on rotational sweep, Computer-Aided Design & Applications 6(1), 2009, 69–79. <https://doi.org/10.3722/cadaps.2009.69-79>

- [8] Mitani, J.; Igarashi, T.: Interactive design of planar curved folding by reflection, Pacific Conference on Computer Graphics and Applications - Short Papers, 2011, 77–81.
- [9] Tachi, T.: Simulation of rigid origami, Origami 4: The Fourth International Conference on Origami in Science, Mathematics, and Education, 2009, 175–187.
- [10] Tachi, T.: One-DOF rigid foldable structures from space curves, in Proceedings of the IABSE-IASS Symposium, 2011.
- [11] Tachi, T.: Composite rigid-foldable curved origami structure, in Proceedings of Transformables, 2013.
- [12] Tang, C.; Bo, P.; Wallner, J.; Pottmann, H.: Interactive design of developable surfaces, ACM Transactions on Graphics, 35(2), 2016, 12:1-12. <https://doi.org/10.1145/2832906>
- [13] Rabinovich, M.; Hoffmann, M.; Sorkine, O.: Discrete geodesic nets for modeling developable surfaces, ACM Transactions on Graphics, 37(2), 2018. <https://doi.org/10.1145/3180494>
- [14] Rabinovich, M.; Hoffmann, M.; Sorkine, O.: The shape space of discrete orthogonal geodesic nets, ACM Transactions on Graphics, 37(6), 2018. <https://doi.org/10.1145/3272127.3275088>
- [15] Watanabe, Y.; Mitani, J.: Modelling the folding motions of a curved fold, Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education, 2018, 1135-1150.
- [16] Watanabe, Y.; Mitani, J.: Interactive modelling of curved folds with multiple creases considering folding motions, Computer-Aided Design & Applications, 16(3), 2018, 452-465. <https://doi.org/10.14733/cadaps.2019.452-465>