Computer-Aided Design and Applications www.cadanda.com

Interactive Curved Fold Modeling using a Handle Curve

Jun Mitani[1] and Kaoru Ohashi[2]

[1]University of Tsukuba, mitani@cs.tsukuba.ac.jp
[2]University of Tsukuba, s2020574@s.tsukuba.ac.jp

Corresponding author: Jun Mitani, mitani@cs.tsukuba.ac.jp

Abstract. A developable surface has a useful engineering property of being formed by bending a non-stretchable material, and various methods for designing its shape have been proposed. However, the shape design of developable surfaces with curved folds is still a difficult problem. In this paper, we propose a user interface that enables the interactive design of a developable surface with a curved fold along a user-specified space curve, by using an auxiliary curve called a "handle curve". The proposed interface is similar to the loft command, which generates a developable surface by using two curves implemented in common 3DCAD software. The proposed approach has an important property that it can avoid the collision of rulings, which is a major problem when generating a developable surface.

Keywords: Curved Folds, Developable Surface Modeling, User Interface
DOI: https://doi.org/10.14733/cadaps.2023.275-289

1 INTRODUCTION

A developable surface is a surface that can be made by bending non-stretchable material and is useful for engineering purposes. The design of ship hulls is one of the most common examples of the application of developable surfaces [9], and they are also widely used in architecture, machine parts, clothing design, etc. While smooth developable surfaces are limited to those made from a combination of conical, cylindrical, tangent developable, and planar surfaces, the addition of folds allows a greater variety of shapes to be expressed. The design of shapes with folds has been studied in the field of origami, where a single sheet of paper is folded to create various shapes. Although it is difficult to fold a thick material such as a metal sheet along a curve, from an engineering point of view, the technology to create shapes with curved folds is becoming more familiar as seen in the case of Robofold [2]. The challenge that needs to be taken now is to provide a suitable design tool for interactively designing such a developable surface with curved folds.

A developable surface is composed of a sequence of linear elements called *rulings*. Although its geometrical components are simple, interactively manipulating its shape is still difficult because of the geometric constraint that the Gaussian curvature is zero at every point on the surface. The fact that there should be no collision of rulings is another critical factor that makes the problem difficult. The geometric constraints on the developable surface with a curved fold are even more complex (detailed in Section 3), and the interactive design of such a shape is still an open problem. In this paper, we resolve this problem by proposing a novel user interface for interactively designing such shapes. Figure 1 shows an example of the shape covered in this paper, which is a developable surface with a single fold along a curve. We assume that the curve is smooth and does not have an inflection point.

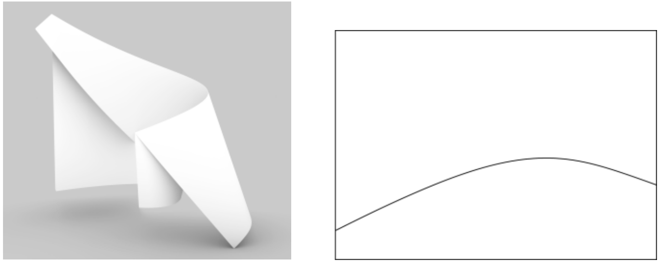


Figure 1: A developable surface with a single curved fold.

The user provides a three-dimensional curve that represents a curved fold as an input, which is referred to as a *crease curve* hereinafter. Following this, the user inputs an auxiliary curve called a *handle curve*, which is a curve used to manipulate the shape. By manipulating the position and scale of this handle curve, the user can interactively deform the developable surface with a fold along the crease curve.

The main features of the interface proposed in this paper are as follows.

- The user can directly specify the crease curve. This is a suitable interface when the shape and the location of the fold are important.
- The user can interactively manipulate the shape of the developable surface with the specified fold by manipulating an auxiliary curve called the handle curve, which allows quick trial-and-error operations.
- The user does not need to explicitly specify the angle of the fold or the curve of the fold mapped to the plane.
- The user does not have to worry about ruling collisions, which is an annoying problem when dealing with developable surfaces.
- The idea is simple and easy to implement.
- The operation is simple and intuitive, as it only requires the input of two curves in three-dimensional space and the border of the surface on the two-dimensional map.

The remainder of this paper is organized as follows: Section 2 introduces related work, Section 3 describes the geometry of developable surfaces with a curved fold, Section 4 describes the proposed method, and Section 5 presents results and evaluations. Finally, conclusions are given in Section 6.

2 RELATED WORK

There have been several studies related to the modeling technique of developable surfaces. This section introduces related studies on developable surfaces without folds and with folds in Section 2.1 and Section 2.2, respectively.

2.1 Developable Surface Modeling

Several approaches have been proposed to generate developable surfaces. One of the most common methods, called "loft", is implemented in most CAD software to generate an approximate developable surface between two curves. Such a method is useful, for example, when designing a ship hull [9]. A method has been proposed to maintain the developability by adding inner vertices and folds on the surface when there exists ruling collision when generating a developable surface between two curves [4]. As a different approach, a method generates a developable surface such that its geodesic coincides with a given curve was proposed [1]. Once the geodesic curve is determined, the rulings that make up the developable surface can be calculated. The geodesic curve is represented as a Bézier curve, and the user can manipulate the shape of the developable surface through the control points of the curve. Although this interface is intuitive, since it is difficult to avoid ruling collisions, the shapes that can be created are limited to strips along the geodesic curve with limited width. As an approach that does not use ruling in the first place, a surface representation called DOG net was proposed [10], in which the surface is represented with a quad net. The model is able to approximate a developable surface by adjusting the angles of the corners of the quads. The user can interactively manipulate the shape by specifying the position of points on the surface.

2.2 Developable Surface with Curved Folds

Due to the strict geometric constraints of developable surfaces with curved folds, it is difficult to design them into the intended shape. A method for reconstructing such geometric models from scanned data of real three-dimensional objects was proposed in [5]. The paper describes in detail a method for discrete representation of developable surfaces with quad strips. The geometry used in our method is similar to this method. More details are given in Section 3.

In the field of origami design, some methods for designing surfaces with curved folds had been proposed [6][7]. However, the shapes generated by the method in [6] are limited to rotationally symmetric ones. The method proposed in [7] of applying a mirror inversion to an existing developable surface to add curved folds limits the curves planner. A method for specifying a three-dimensional crease curve has also been proposed [15]. The user specifies the parameters for the crease curve, the angle function of the fold along the curve, and the mapped curve on the plane. This simple approach revealed that it is difficult to obtain the intended shape through controlling the parameters, and it is also difficult to avoid collisions of the rulings. An approach of extending the DOG net to handle shapes with creases was intuitive [11]. However, the crease curves cannot be directly specified.

The equations for calculating rulings based on the geometric constraints of a developable surface with a single fold was formalized by [13] proposing a method to simulate 1-DOF rigid-body kinematics using a discretized model. The equations are used in this paper for shape generation. A method for simulating the folding motion from the development with curved folds with mountain-valley assignment was proposed in [12]. Although the method can generate a shape with curved folds, it is based on a spring-math physical simulation, which allows for small expansions and contractions of the edges that make up the model.

All methods introduced here have their advantages and disadvantages, and no method can interactively generate a shape with specified folds as proposed in this paper.

3 GEOMETRY OF A DEVELOPABLE SURFACE WITH A CURVED FOLD

The geometry of a developable surface with a curved fold was formulated by Fuchs and Tabachnikov [3] and it was organized by Tachi [13] as follows. The curvature $k(s)$ and torsion $\tau(s)$ of the crease curve in three-dimensional space is defined as follows:

$$\boldsymbol{k}(\boldsymbol{s}) = |\boldsymbol{T}'(\boldsymbol{s})| \tag{1}$$

$$\boldsymbol{\tau}(\boldsymbol{s}) = -< \boldsymbol{B}'(\boldsymbol{s}), \boldsymbol{N}(\boldsymbol{s}) > \tag{2}$$

where s is the arc length parameter, and $\boldsymbol{T}(s)$, $\boldsymbol{N}(s)$, and $\boldsymbol{B}(s)$ represent the tangent vector, the normal vector, and the binormal vector, respectively. Let $\alpha(s)$ denote the folding angle at position s on the crease curve, and the ruling passing through the position lies on the plane where the osculating plane is rotated by angle $\alpha(s)$ around the tangent vector. The fold angle $\alpha(s)$ takes the same value on both sides of the crease (Fig. 2 left). There is the following relationship between $k(s)$

and $k_{2D}(s)$ which is the curvature of the crease curve mapped on to the unfolded state of the developable surface (Fig. 2 right).

$$k_{2D}(s) = k(s)\cos\alpha(s) \tag{3}$$

With this equation, the crease curve mapped on the development is calculated with the curvature of the crease curve and the fold angle along the curve. The angles β_L and β_R formed by the left and right rulings and the tangent vector in the development can be derived by the following equations, respectively.

$$\cot\beta_L(s) = \frac{-\alpha(s)' + \tau(s)}{k(s)\sin\alpha(s)} \tag{4}$$

$$\cot\beta_R(s) = \frac{\alpha(s)' + \tau(s)}{k(s)\sin\alpha(s)} \tag{5}$$

Then, the ruling direction vectors $\boldsymbol{r}_L$ and $\boldsymbol{r}_R$ in three-dimensional space on the left and right sides of the crease curve are expressed by the followings.

$$\boldsymbol{r}_L = \cos\beta_L\boldsymbol{T} - \sin\beta_L\cos\alpha\boldsymbol{N} + \sin\beta_L\sin\alpha\boldsymbol{B} \tag{6}$$

$$\boldsymbol{r}_R = \cos\beta_R\boldsymbol{T} + \sin\beta_R\cos\alpha\boldsymbol{N} + \sin\beta_R\sin\alpha\boldsymbol{B} \tag{7}$$

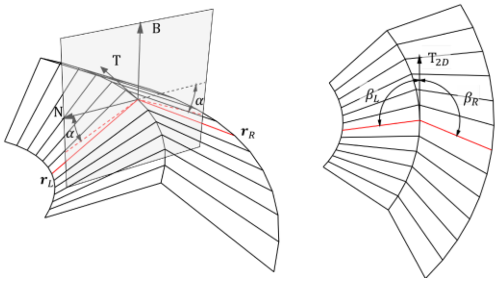


Figure 2: Parameters of the crease curve. Folded state (left) and the unfolded state (right).

Based on the above relationships, the shape of the developable surface with a curved fold is determined by any of the following combinations:

(1) a pair of the fold angle function $\alpha(s)$ and the crease curve (3D)

(2) a pair of the fold angle function $\alpha(s)$ and the mapped crease curve (2D)

(3) a pair of the mapped crease curve (2D) and the crease curve (3D)

A shape can be uniquely determined by specifying any of these pairs. However, it is difficult to specify the angle function $\alpha(s)$ manually, and the relationship between the crease curve (3D) and the mapped crease curve (2D) is not intuitive. This makes it difficult to obtain the intended shape by specifying them. In addition, it is especially difficult to avoid the collision of the rulings because it is hard to predict how the rulings will be placed beforehand. Further, even small changes in the fold angle and crease curves can affect the orientation of rulings drastically. Although a method for controlling these elements and visualizing the rulings was proposed in [15], it is still difficult to avoid the collision of rulings.

4 PROPOSED METHOD

In this section, the proposed method is explained. First, the flow of shape modeling is shown. Then, the details of the method are described and the relationship between the crease curve, the handle curve, and the resulting shapes are discussed.

4.1 Flow of the Shape Modeling

The shape modeling by the proposed method is done by inputting two curves. One is the crease curve in three-dimensional space that represents a curved fold, and the other one is the handle curve, which is used to manipulate the shape of the developable surface. This is a simple interface similar to the common loft command used to generate a surface between two curves. In addition to this, an operation to specify the border of the surface is required for the user.

Figure 3 illustrates the overall flow. First, the user inputs the crease curve (labeled C) as a three-dimensional Bézier curve (Fig. 3a). Then, the user inputs the handle curve (labeled C' in Fig. 3b). At this point, a developable surface with the curved fold is generated with predefined width. The user manipulates the developable surface by moving or deforming the handle curve (Fig. 3c). The handle curve can be placed on either the concave or convex side of the crease curve. The user specifies the border of the surface by referring to the crease curve mapped on the plane, i.e. the development (the yellow rectangle in Fig. 3d and 3e). Then the final shape is generated by applying the border (Fig. 3f). The user can continue to edit the shape by modifying the crease curve, the handle curve, and the border at any time.

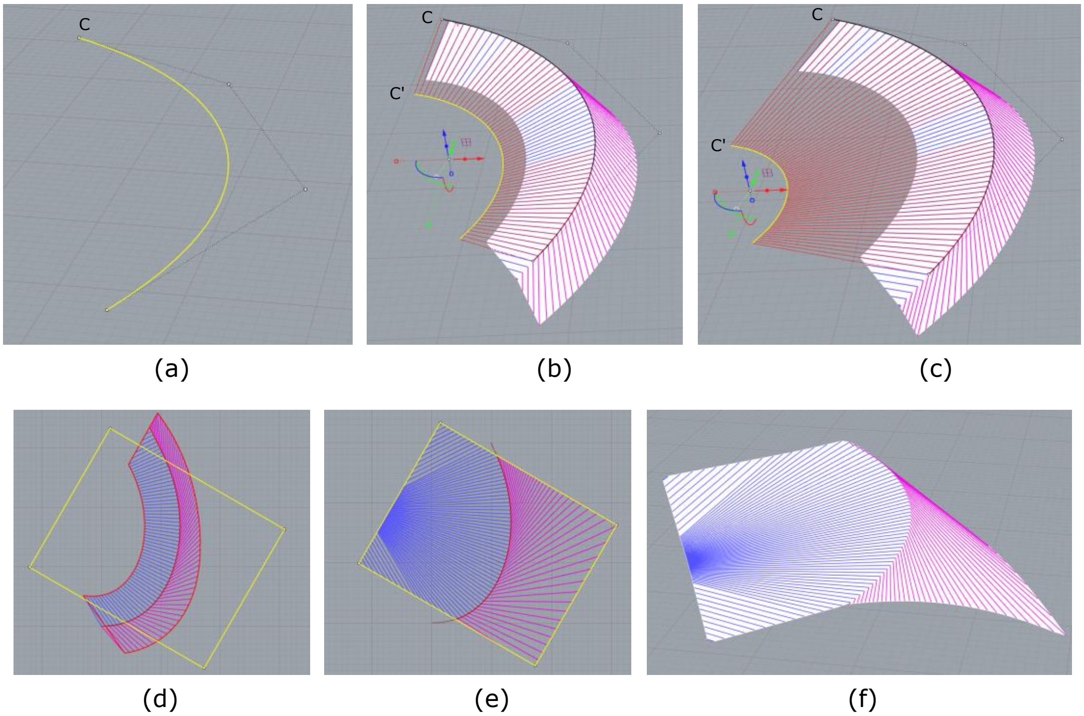


Figure 3: Flow of the operation for generating a developable surface with a curved fold. (a) The crease curve. (b) The handle curve was input. (c) The handle curve was moved and modified. (d) The border of the surface was drawn on the mapped pattern. (e) The border of the surface was updated. (f) The generated surface.

4.2 Model Generation

The discretized developable surface with a curved fold is represented by two adjusted strips of quads (Quad-strip) as in the approach of [5]. The edges of each quad consist of two rulings, a part of the crease curve and a part of the border. As illustrated in Fig. 4, the user-specified crease curve is denoted as C and the handle curve as C' hereinafter. Both C and C' are Bézier curves. The degree of the Bézier curve was set to be six (the reason will be discussed in Section 5). In the following, we describe the procedure to generate a developable surface with a curved fold along with the curve C.

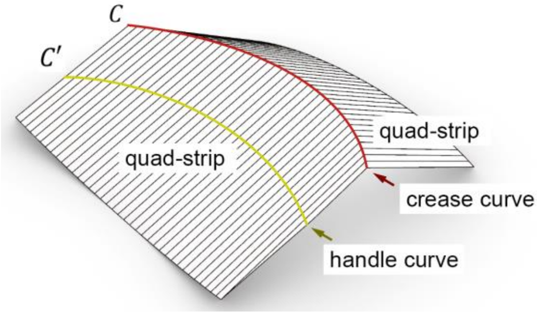


Figure 4: Representation of the model in the proposed method.

First, C is sampled at regular arc length intervals, and each sampled point is denoted as $\boldsymbol{p}_i$. Then C' is equally sampled by the same number of the sampled points of C, and each point is denoted as $\boldsymbol{p}'_i$. Next, a line, connecting $\boldsymbol{p}_i$ and $\boldsymbol{p}'_i$, is generated for all i. The lines are called *pseudo-rulings*. The reason for the prefix "pseudo" is that we want to generate a developable surface with these lines as the rulings, but in practice, they do not necessarily satisfy the geometric constraints described in Section 3. Therefore, we calculate the ruling consistent with the equations, passing through $\boldsymbol{p}_i$ and near $\boldsymbol{p}'_i$ for all i by referring to the pseudo-rulings. First, we obtain the unit vector $\boldsymbol{u}_i = \frac{\boldsymbol{p}'_i - \boldsymbol{p}_i}{|\boldsymbol{p}'_i - \boldsymbol{p}_i|}$, which is the direction vector of the pseudo-ruling. Then, the point $\boldsymbol{v}_i$, which is the projection of $\boldsymbol{u}_i$ onto the normal plane (Fig. 5) is obtained by the following equation,

$$\boldsymbol{v}_i = \boldsymbol{u}_i - < \boldsymbol{u}_i, \boldsymbol{T}_i > \boldsymbol{T}_i. \tag{8}$$

The angle α_i of the fold at $\boldsymbol{p}_i$ is calculated as

$$\alpha_i = \arccos < \boldsymbol{N}_i, \frac{\boldsymbol{v}_i}{|\boldsymbol{v}_i|} >. \tag{9}$$

By using this value, the two rulings for both sides of the crease curve extending from point $\boldsymbol{p}_i$ that satisfies the geometrical constraints are obtained from equations (6) and (7).

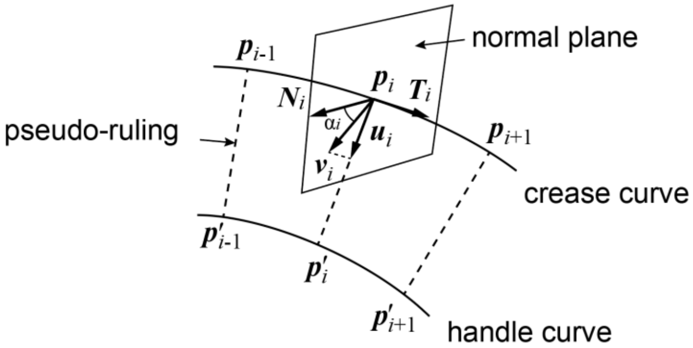


Figure 5: Relationship between the crease curve, handle curve and pseudo-rulings.

As the result, each pseudo-ruling through $\boldsymbol{p}_i$ rides on a tangent plane at $\boldsymbol{p}_i$ (Fig. 6). Therefore, the pseudo-ruling can be regarded as a line that determines the orientation of the tangent plane of the surface at $\boldsymbol{p}_i$.

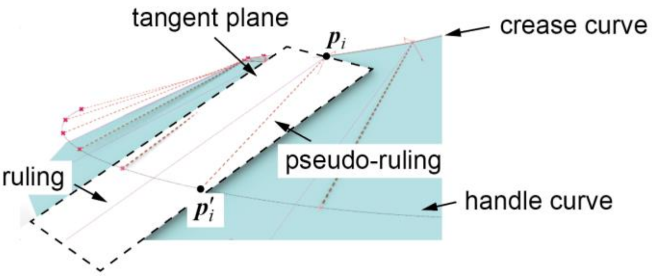


Figure 6: The pseudo-ruling defines the tangent plane on which the ruling passes through $\boldsymbol{p}_i$ lies.

4.3 Relationship Between the Handle Curve and the Resulting Surface

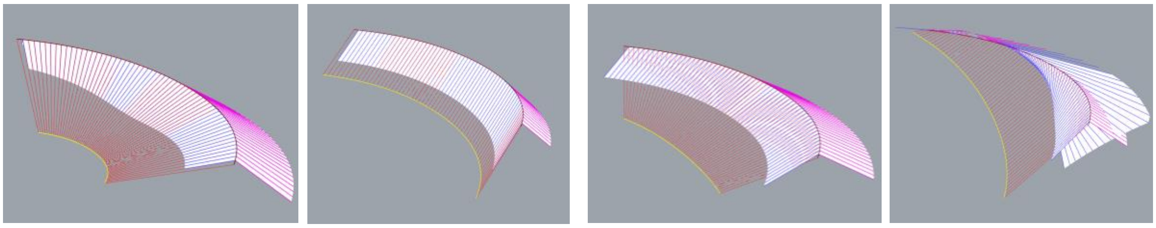
In general, pseudo-rulings and rulings which satisfy the geometric constraints do not coincide. However, they do coincide when certain conditions are met, at which point the pseudo-rulings can be considered as the exact rulings.

If the shape of the handle curve is similar to the crease curve with a scale factor s ($\neq 1$), that is, point $\boldsymbol{p}_i$ on C and point $\boldsymbol{p}'_i$ on C' for any i are represented by the relation

$$\boldsymbol{p}'_i = s\boldsymbol{p}_i + \boldsymbol{t} \qquad (s \neq 1), \tag{10}$$

where $\boldsymbol{t}$ is a three-dimensional vector, then the pseudo-rulings intersect at a point. In this case, the pseudo-rulings can be the exact rulings, and C' exactly lies on the surface (Fig. 7a) because the pseudo-rulings can be considered as the rulings of a conical surface. When the handle curve is congruent with the crease curve ($s = 1$ in equation (10)), the pseudo-rulings are parallel and form a cylindrical surface. Even in this case, the pseudo-rulings can be exact rulings, and C' lies exactly on the cylindrical surface (Fig. 7b). The conditions under which pseudo-rulings can be exact rulings are summarized as the crease curve and the handle curve lie on a common "conical" or "cylindrical" surface.

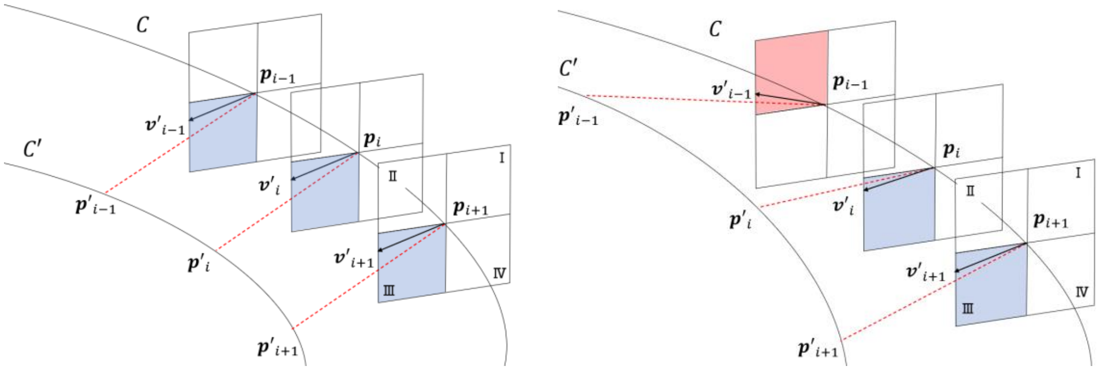
Other than the above cases, i.e. the handle curve is not similar to the crease curve, or if it is rotated, the pseudo-rulings do not coincide with the rulings (Fig. 7c). In this case, appropriate surface may not be generated depending on the shape of the handle curve (Fig. 7d).



(a) (b) (c) (d)

Figure 7: Generated shapes. (a) A conical surface generated with the handle curve similar to the crease curve. (b) A cylindrical surface generated with the handle curve congruent with the crease curve. (c) A tangent developable surface generated in other cases. (d) Depending on the shape of the handle curve, appropriate surface may not be generated.

The angle of the fold can change continuously along the crease curve, but the direction of the fold cannot change from a mountain fold to a valley fold or vice versa in the middle of the curve. If the crease curve is a mountain fold, the angle of the fold is limited to $0 < \alpha(s) < \pi/2$ for any s, and if the fold is a valley fold, it is limited to $\pi/2 < \alpha(s) < 0$ for any s. Therefore, when the pseudo-ruling is projected onto the normal plane, the quadrants to which the projected line belongs must all be the same (Fig. 8a). However, if there are projected lines belonging to different quadrants as shown in Fig. 8b, the appropriate surface cannot be generated as shown in Fig. 7d. This should be noted when the user manipulates the handle curve. This problem does not arise when the handle curve is similar to the crease curve and the positional relationship is parallel shifted, as expressed in eq. (10).



(a) (b)

Figure 8: The case where all the lines projecting the pseudo-ruling to the normal plane are in the same quadrant (a) and the case where they are not (b).

5 RESULT

The proposed method was implemented, and the shapes generated by the system are shown in this section. The geometry of the shapes was evaluated in terms of their developability, and the planarity of each quad face that constitutes the Quad-strips. A user test was also conducted. In addition, examples of extensions to the proposed method to broaden the range of forms that can be designed will be presented.

5.1 Implementation

We implemented the proposed method on Rhinoceros, a popular 3DCAD software, with Grasshopper, its plug-in. The basic operations, such as inputting curves and moving and deforming them, were done using the interface provided by Rhinoceros. The algorithm that takes the crease curve, the handle curve, and the border of the surface created in Rhinoceros and generates a developable surface with a curved fold was implemented in Grasshopper. The flow of operations by the user is as described in Section 3.1.

The type of curves, such as Bézier or NURBS, and their degree affect the resulting shapes. As a preliminary experiment, we used the B-Spline curve, which is a standard curve in the Rhinoceros, to investigate the effect of its degree on the shape. The experimental results are shown in Fig. 9. Although the degree must be at least three, in order to calculate the torsion of the curve, the result indicates that the degree should be more than five, in order to make the change of the torsion along the curve smooth and to suppress the ruling collision. For this reason, we decided to use 6-degree Bézier curves for both the crease curve and the handle curve. The curve input by the user is converted to a 6-degree Bézier curve while keeping the location of the endpoints.

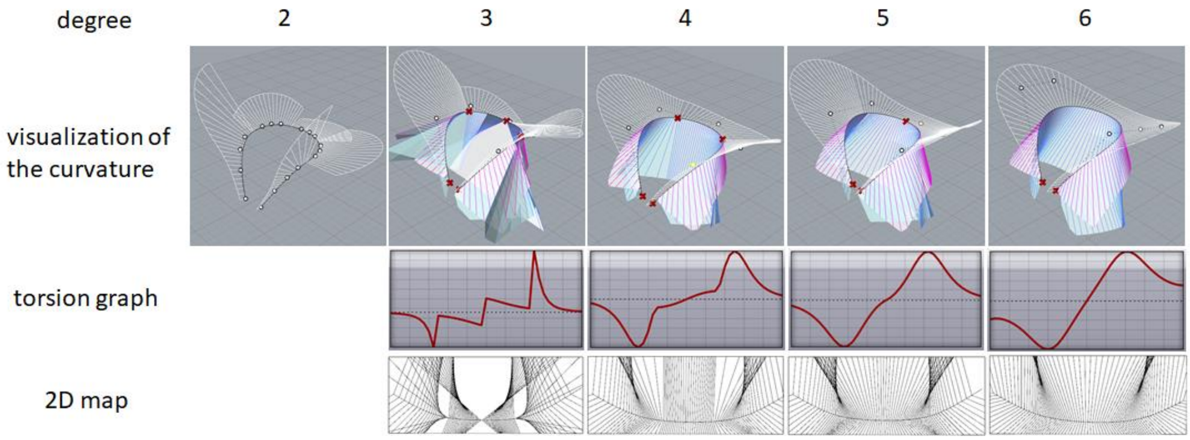


Figure 9: Effect of the degree of a curve on curvature, torsion and map of rulings. Red X marks show locations that are not G3 continuous.

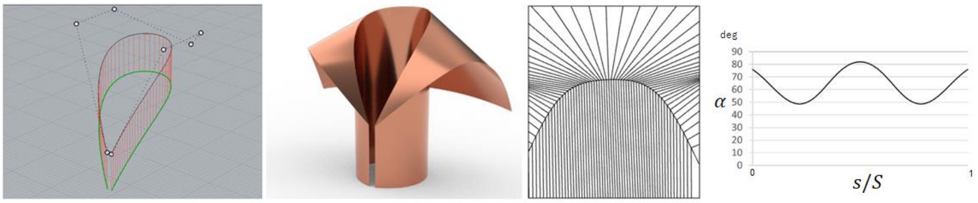
Figure 10 shows the examples generated with the implemented design system. The user interactively manipulated the handle curve to generate the shape while viewing the surface generated by the proposed system. The geometry can be generated at interactive speed time because the proposed method does not require any optimization process. Figure 10a is an example where the crease curve and the handle curve are congruent. The design was aimed at the shape of a component used in packaging machinery. It is shaped as a vertical cylindrical surface that is folded back to the opposite side by a crease. Figure 10b is another example where the crease curve and the handle curve are similar. The design is a reproduction of the origami artwork in the book [8]. From left to right in Fig. 10, each figure shows the user input (white circles are control points of the Bézier curve, the black curve is the crease curve, and the green one is the handle curve), the generated 3D model, the development with projected rulings, and the graph of the angle function where s is the arc length parameter and S is the length of the curve. Since the crease curve and the handle curve are congruent (Fig. 10a) or similar (Fig. 10b) and are in parallel positions, the pseudo-ruling and the ruling have coincided. This makes the user be able to manipulate the shape intuitively. In addition, no effort was needed to avoid the collision of rulings. In particular, the angle function in Fig. 10b has a complex shape and would be nearly impossible to generate using the existing method of manual specification by the user. Because of this, the effectiveness of the proposed method has been demonstrated.

5.2 Evaluation of the Developability

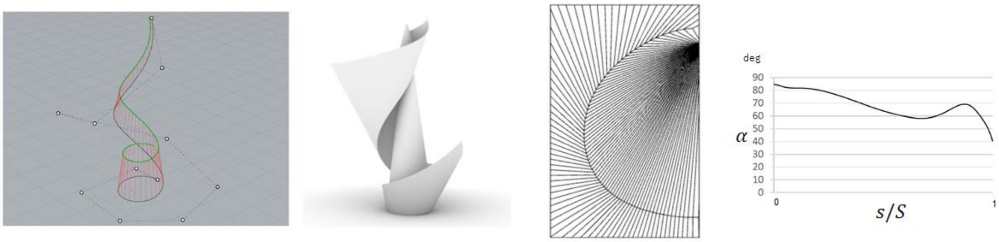
Two numerical evaluations of the validity of the geometries generated by the proposed method were performed: one is the evaluation of the developability, and the other is the evaluation of the flatness of the quads that compose the quad-strips. Since the sum of the angles around each inner vertex should be 2π in the discretized model of a developable surface, the developability can be evaluated by the difference between the sum of angles around each vertex and the value of 2π. Specifically, the error e_i of the inner vertex i is expressed by

$$e_i = \left|2\pi - \theta_{sum_i}\right|, \tag{11}$$

where θ_{sum_i} is the total sum of the angles around i. Since this is a local value, the average (e_{ave}), and the maximum (e_{max}) of this value among all inner vertices are used to evaluate the overall developability.



(a)



(b)

Figure 10: Developable surfaces with a curved fold generated by the proposed method. (a) The handle curve is congruent to the crease curve. (b) The handle curve is similar to the crease curve.

For both indices, the smaller the value, the better. When quad-strips are used to represent a developable surface, each quad must be flat. The flatness is evaluated by the ratio of the shortest distance d_i between two diagonals to the average l_{ave_i} of the lengths of the two diagonals, as used in [14]. The values of d_i and l_{ave_i} are obtained by the following equations (Fig. 11),

$$d_i = \left| \frac{< (\boldsymbol{y}_{i+1} - \boldsymbol{x}_i) \times (\boldsymbol{y}_i - \boldsymbol{x}_{i+1}), \boldsymbol{x}_{i+1} - \boldsymbol{x}_i >}{|(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i) \times (\boldsymbol{y}_i - \boldsymbol{x}_{i+1})|} \right| \tag{12}$$

$$l_{avg_i} = \frac{|\boldsymbol{y}_{i+1} - \boldsymbol{x}_i| + |\boldsymbol{y}_i - \boldsymbol{x}_{i+1}|}{2} \tag{13}$$

and the flatness of the quad i is represented by

$$\frac{d_i}{l_{ave_i}}. \tag{14}$$

The flatness was evaluated by calculating the average (f_{ave}) and maximum (f_{max}) of the value among all quads. The smaller both of these values are, the better.

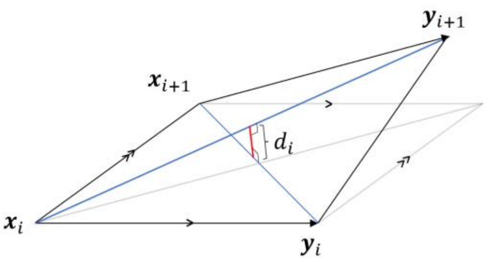


Figure 11: Evaluation of the flatness of a quadrangle.

These values were evaluated by the three examples shown in Fig. 12. All of these examples are reproductions of origami artworks included in the book [8]. To evaluate the effect of the number of divisions on discretizing the crease curve, both the developability and the flatness were calculated while changing the number of divisions by 20 in the range of 20 to 100.

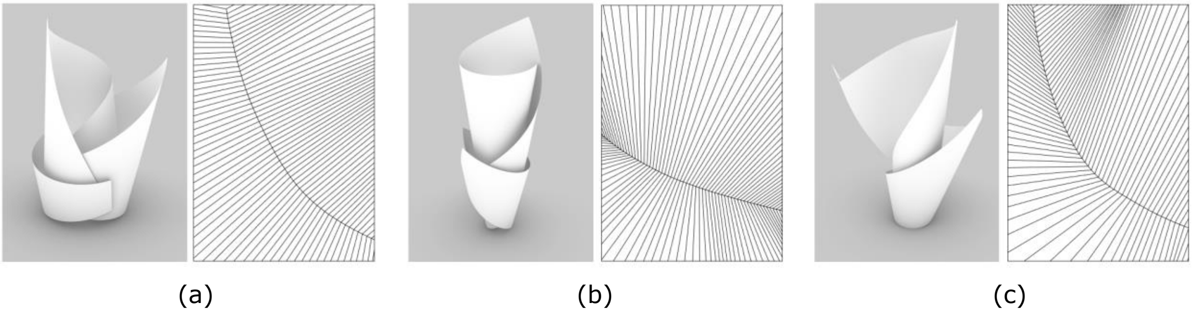


Figure 12: The three shapes used in the evaluation. The number of divisions is set to 500 for the models in the figures, and set to 60 for the maps.

The average and maximum errors for developability (e_{ave} and e_{max}) and for flatness (f_{ave} and f_{max}) are shown in the graphs in Fig. 13. The criterion for determining that the shape is sufficiently developable was set at 0.05 degrees or 8.7×10^{-4} rad. For flatness, a value within 1.0×10^{-2} is considered to be sufficiently flat as described in [14]. These values are indicated by the yellow horizontal lines marked with a star in the graphs. For all the models, the maximum error for developability (e_{max}) is less than this criterion if the number of divisions is 60 or more, and the maximum error for flatness (f_{max}) is less than this criterion, regardless of the number of divisions. Since the crease curves are generally divided into hundreds to obtain a smooth appearance, as shown in Fig. 12, the resulting shape can be seen as sufficiently developable.

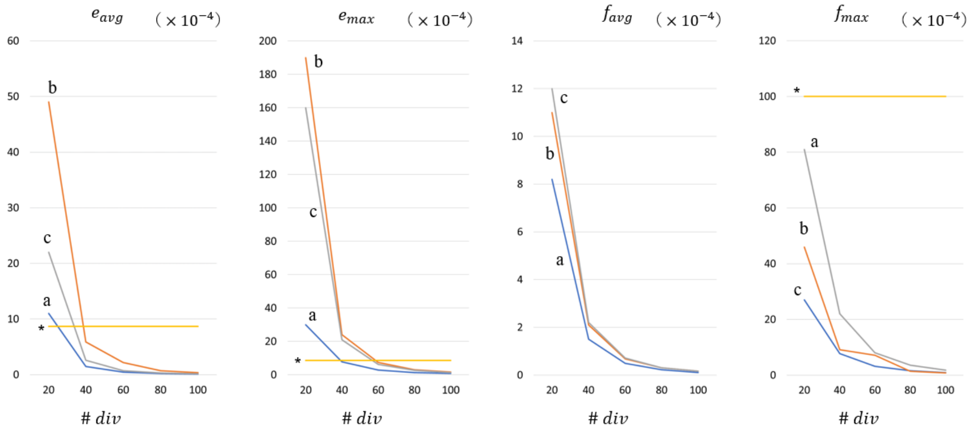


Figure 13: Graphs of the values of e_{ave}, e_{max}, f_{ave}, and f_{max} against the number of divisions of the crease curve. Labels a, b, and c correspond to the shapes in Fig. 12.

5.3 User Test

A user test was conducted to evaluate the usability of the system implementing the proposed method. The subjects were three undergraduate students and three graduate students in the field of computer science, who were not very proficient in using 3DCAD. After 20 minutes of explanation and practice of the system, the subjects were asked to create shapes similar to two shapes presented as examples, and to create another shape freely. The resulting shapes are shown in Fig. 15. In (a) and (b), the models on the left are examples, and the others are the models created by the subjects. Each took approximately 10 minutes of work. In (c), the models were made free by the subjects. Then, a questionnaire based on the five-point Likert scale was administered to the subjects regarding the following two questions:

Q1: It was easy to understand how to operate.

Q2: It was easy to create the intended shape.

The choices are as follow:

1. Strongly disagree; 2. Disagree; 3. Neither agree nor disagree; 4. Agree; 5. Strongly agree.

The graphs summarizing the results of the questionnaires are shown in Fig. 14, with the above scores on the horizontal axis and the number of respondents on the vertical axis. Although statistical evaluation is not possible due to the small number of subjects, it can be seen that it is easy to understand the operation, but it is not easy to create the intended shape until one becomes accustomed to it.

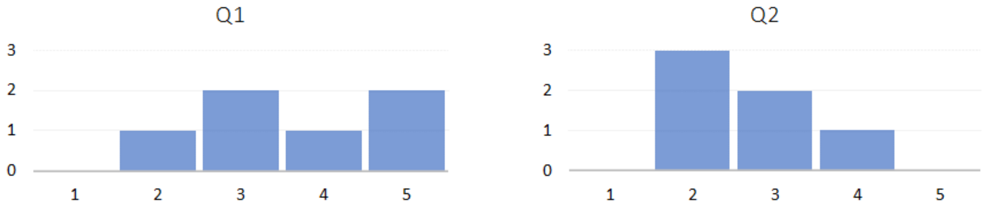


Figure 14: The graphs summarizing the results of the questionnaires.

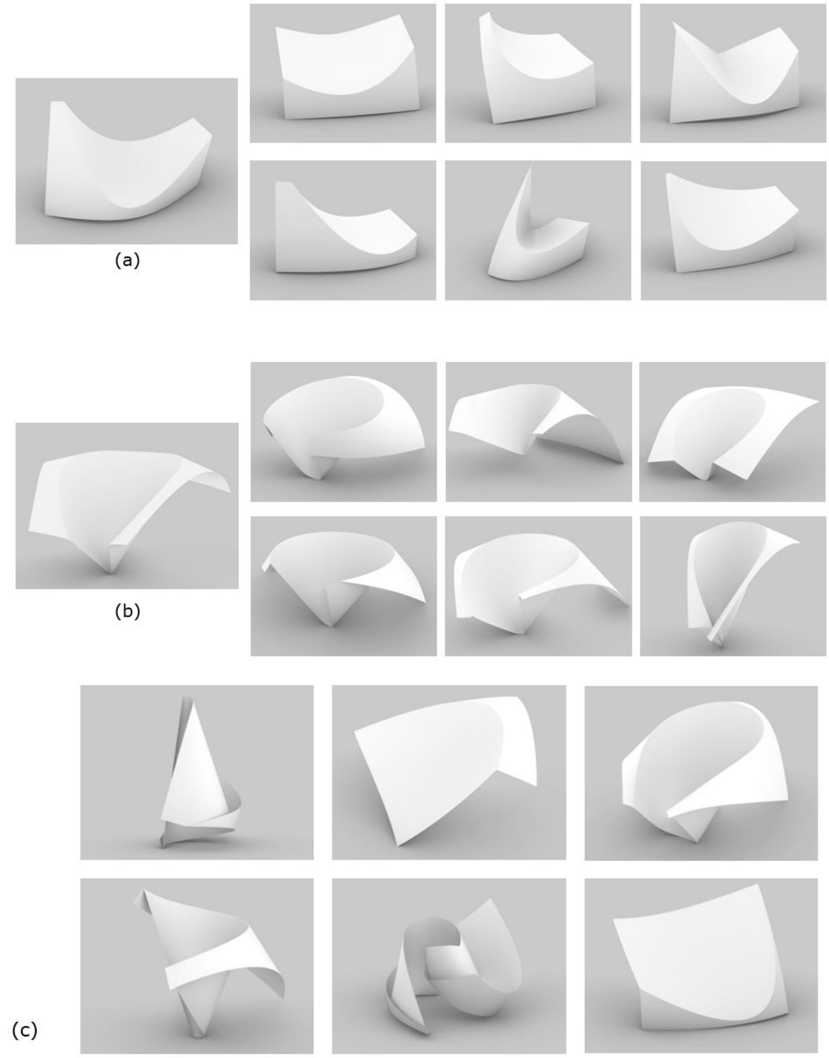
There was a positive comment in the free-text portion of the questionnaire, such as "By looking at the example, I was able to make the shape as I expected. I felt it was easy to use with no failures." On the other hand, some said, "It was difficult to make the surface border rectangular," and "It was difficult to put in the curves as intended." Although the border can be freely specified as a polygon with a mouse click on the 2d map, it was difficult to predict how it would be reflected in the three-dimensional shape. Also, the problem is that freely specifying spatial curves is difficult unless one is proficient in 3DCAD in the first place.

Although there are still issues to be addressed, it is significant that we have proposed a new design tool, because without such a system, it would not be possible to create the shapes of developable surfaces with a curved fold, as shown in Fig. 15.

5.4 Extension of the Proposed Method

A slight extension of the proposed method increases the convenience of shape modeling. For example, an existing 3D shape such as a cone can be used as a guide to input the crease curve and the handle curve. By placing both the crease curve and the handle curve on the surface of the cone object, a folded shape of the existing cone surface can be obtained (Fig. 16). Figure 17 shows an example of modeling a strip with a fold by this approach by using a helix curve drawn on a cone.

One or more folds can be added to the model obtained by the proposed method. Adding a crease curve on an already existing developable surface determines both the 3D shape and the 2D mapped shape of the crease curve (the case of (3) below Figure 2), so the shape after folding along it is uniquely determined. Figure 18 shows an example of adding two crease curves to the shape generated by the proposed method (only with the center crease). When adding new folds, the user needs to take care to avoid collision of rulings and self-intersection of the surface.

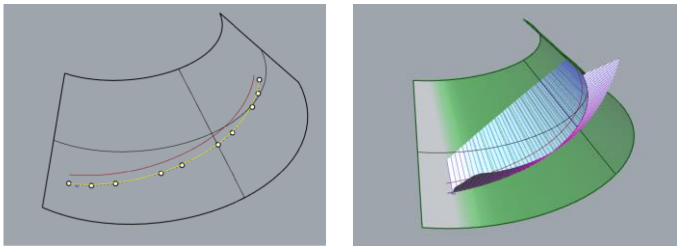


(a)

(b)

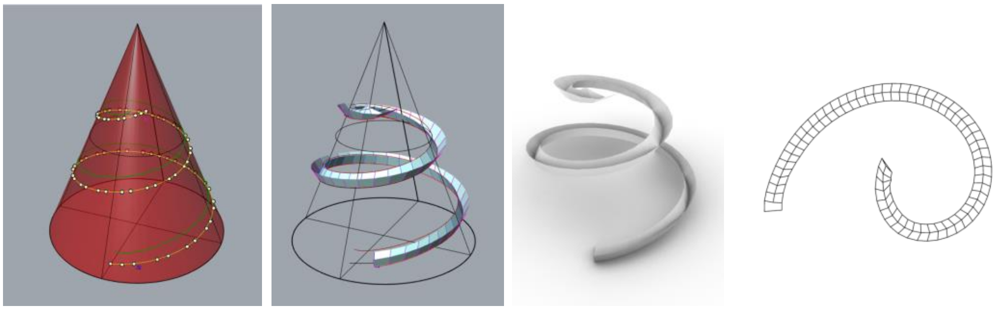
(c)

Figure 15: 3D models designed in the user test.



(a) (b)

Figure 16: Example of shape modeling with existing surfaces. (a) The crease curve (yellow) and the handle curve (red). (b) Generated shape.



(a) (b) (c) (d)

Figure 17: A strip with a crease curve drawn on a cone. (a) The crease curve (yellow) and the handle curve (green) are generated by adding an offset to the crease curve. (b) The shape is generated by the proposed method. (c) Rendered image. (d) Development.

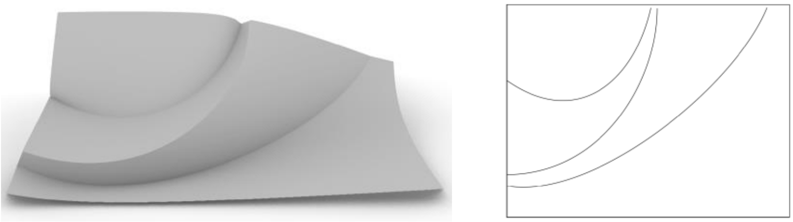


Figure 18: Shape with three curved folds (left) and its development (right). Two folds were added after generating the shape with the central fold by the proposed method.

6 CONCLUSION

We proposed a novel user interface for designing a developable surface with a curved fold with intuitive operations. With the proposed method, the crease curve, the handle curve, and the contour shape are used to define the final shape. The user can manipulate the shape interactively and intuitively by operating the handle curve. It also has the advantage of avoiding the collision of rulings. The actual implementation of the system confirms that the proposed method has the advantages summarized at the end of the first chapter through evaluations described in Section 5. Despite the inherent difficulty in inputting the intended 3D curves, the results of user testing indicate that further improvements in user usability are in the future work

ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR1911, Japan.

Jun Mitani, https://orcid.org/0000-0002-1596-844X
Kaoru Ohashi, https://orcid.org/0000-0002-0565-309X

REFERENCES

[1] Bo, P.; Wang, W.: Geodesic-Controlled Developable Surfaces for Modeling Paper Bending, Computer Graphics Forum, 26(3), 2007, 365-374. http://dx.doi.org/10.1111/j.1467-8659.2007.01059.x

[2] Epps, G.: Robofold and robots. IO, Made by Robots: Challenging architecture at a larger scale, Architectural Design, 84, 2014, 68–69. https://doi.org/10.1002/ad.1757
[3] Fuchs, D.; Tabachnikov, S.: More on Paperfolding, The American Mathematical Monthly, 106(1), 1999, 27-35. https://doi.org/10.2307/2589583
[4] Kergosien, Y.L.; Gotoda, H.; Kunii, T.L.: Bending and creasing virtual paper, IEEE Computer Graphics and Applications, 14(1), 1994, 40-48. https://doi.org/10.1109/38.250917
[5] 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
[6] 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
[7] Mitani, J.; Igarashi, T.: Interactive design of planar curved folding by reflection, Pacific Conference on Computer Graphics and Applications - Short Papers, 2011, 77–81.
[8] Mitani, J.: Curved-Folding Origami Design, A K Peters/CRC Press, 2019.
[9] Pérez, F.; Suárez, J.A.: Quasi-developable B-spline surfaces in ship hull design, Computer-Aided Design, 39(10), 2007, 853-862. https://doi.org/10.1016/j.cad.2007.04.004.
[10] 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
[11] Rabinovich, M.; Hoffmann, M.; Sorkine, O.: Modeling curved folding with freeform deformations, ACM Transactions on Graphics, 38(6), 2019. https://doi.org/10.1145/3355089.3356531
[12] Sasaki, K.; Mitani, J.: Simple implementation and low computational cost simulation of curved folds based on ruling-aware triangulation, Computers & Graphics, 102, 2021, 213-219. https://doi.org/10.1016/j.cag.2021.09.012.
[13] Tachi, T.: One-DOF Rigid Foldable Structures from Space Curves, in Proceedings of the IABSEIASS Symposium, 2011.
[14] Verhoeven, F.; Vaxman, A.; Hoffmann, T.; Sorkine, O.: Dev2PQ: Planar Quadrilateral Strip Remeshing of Developable Surfaces, ACM Transactions on Graphics, 41(3), 2022, Article No. 29. https://doi.org/10.1145/3510002
[15] Watanabe, Y.; Mitani. J.: Interactive Modelling of Curved Folds with Multiple Creases Considering Folding Motions, Computer-Aided Design & Applications, 16(3), 2019, 452-465. https://doi.org/10.14733/cadaps.2019.452-465