Title:
Interactive Curved Fold Modeling using a Handle Curve

Authors:
Jun Mitani, mitani@cs.tsukuba.ac.jp, University of Tsukuba
Kaoru Ohashi, s2020574@s.tsukuba.ac.jp, University of Tsukuba

Keywords:
Curved Folds, Developable Surface Modeling, User Interface

DOI: 10.14733/cadconfP.2022.56-60

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, 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.

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, 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.

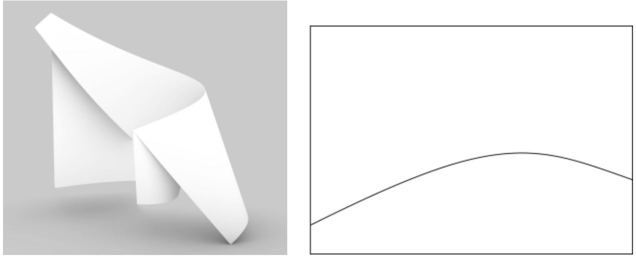


Fig. 1: A developable surface containing a single curved fold.

The user provides a three-dimensional curve that represents a curved fold as an input, which is referred to as a *crease curve*. 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 through 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.

Geometry of a developable surface containing a curved fold:

The geometry of a developable surface with a curved fold was formulated by Fuchs and Tabachnikov [1] and it was organized by Tachi [2] as follows. The curvature $\boldsymbol{k}(\boldsymbol{s})$ and torsion $\boldsymbol{\tau}(\boldsymbol{s})$ of the crease curve in three-dimensional space is defined as follows:

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

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

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 α(s) denote the folding angle at position s on the crease, and the ruling will ride on the plane where the osculating plane is rotated by angle α(s) around the tangent vector. The fold angle α(s) is the same on both sides of the crease (Figure 2). There is the following relationship between the curvature k_{2D}(s) of the 2D curve that is the crease on the development and the curvature k(s) of the crease curve.

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

With this equation, the 2D curve on the development is calculated with the 3D crease curve and the fold angle. The angles β_L and β_R formed by the left and right rulings and the tangent vector on the development can be derived by the following equations respectively.

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

Then, the ruling direction vectors $\boldsymbol{r}_L$ and $\boldsymbol{r}_R$ in 3D on the left and right sides of the crease are expressed by the followings.

$$\boldsymbol{r}_L = \cos\beta_L\boldsymbol{T} - \sin\beta_L\cos\alpha\boldsymbol{N} + \sin\beta_L\sin\alpha\boldsymbol{B} \tag{5}$$

$$\boldsymbol{r}_R = \cos\beta_R\boldsymbol{T} + \sin\beta_R\cos\alpha\boldsymbol{N} + \sin\beta_R\sin\alpha\boldsymbol{B} \tag{6}$$

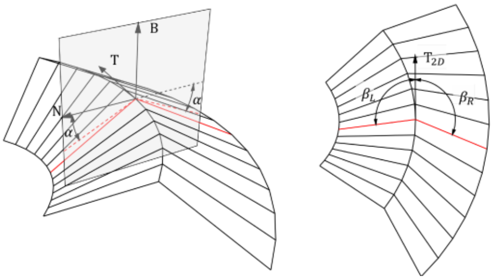


Fig. 2: Parameters of crease curve. Folded state (left) and the unfolded state (right).

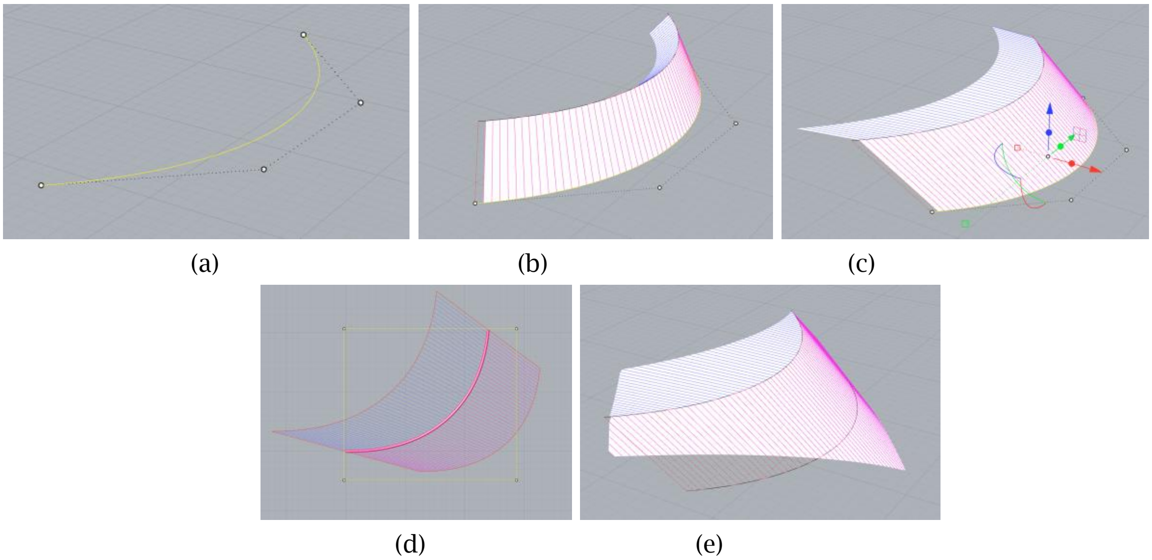
Based on the above relationships, the shape of the developable surface with a curved fold is determined by any of the following combinations:
(1) a pair of the fold angle function α(s) and the crease curve (3D)
(2) a pair of the fold angle function α(s) and the mapped crease curve (2D)
(3) a pair of 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 α(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. Watanabe and Mitani [3] proposed a method to visualize the rulings and obtain the shape by manipulating the elements, but it was still difficult to avoid the intersection of rulings.

Proposed method:

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. In addition to this, an operation to specify the border of the surface is required for the user.

The overall flow is as follows. First, the user inputs the crease curve as a three-dimensional Bézier curve (Fig. 3a). Then, the user inputs the handle curve (Fig. 3b). To simplify the operation, it can be done by duplicating the crease curve. At this point, a developable surface is generated. The surface has the specified curved fold, and the handle curve lies on it. The user manipulates the developable surface by moving or deforming the handle curve (Fig. 3c). The user specifies the border of the surface by referring to the curve which is the crease curve mapped on the plane, i.e. the development (Fig. 3d). Then the final shape is generated by applying the border (Fig. 3e). The user can continue to edit the shape by modifying the crease curve, the handle curve, and the border at any time.


(a) (b) (c)

(d) (e)

Fig. 3: Flow of the operation for generating a developable surface with a curved fold.

The discretized developable surface with a curved fold is represented by two adjusted strips of quads (Quad-strip). The edges of each quad consist of two rulings, a part of the crease curve and a part of the border. 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. In the following, we describe the procedure to generate a developable surface with a curved fold along with the curve C.

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. 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. 4) is obtained by the following equation,

$$\boldsymbol{v}_i = \boldsymbol{u}_i - < \boldsymbol{u}_i, \boldsymbol{T}_i > \boldsymbol{T}_i. \tag{7}$$

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

$$\alpha_i = \arccos < \boldsymbol{N}_i, \frac{\boldsymbol{v}_i}{|\boldsymbol{v}_i|} > . \tag{8}$$

By using this value, the two rulings for both sides and the right side of the crease curve extending from point $\boldsymbol{p}_i$ that satisfies the geometrical constraints are obtained from equations (5) and (6).

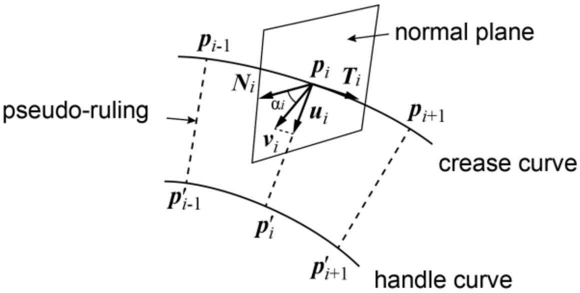


Fig. 4: Relationship between the crease curve, handle curve and pseudo-rulings.

Result:

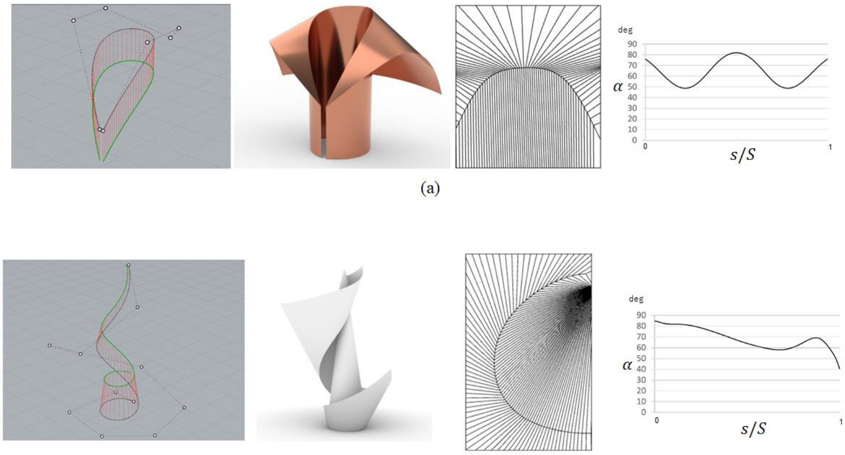
We implemented the proposed method on Rhinoceros, a popular 3DCAD software, with Grasshopper, its plug-in. The basic operations, such as inputting curves and moving and deforming them, were done using the interface provided by Rhinoceros. The algorithm of generating a developable surface with a curved fold was implemented by Grasshopper.

Figure 5 shows the results generated with the implemented system; both of the two examples took only a few minutes to create. Figure 5a 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 5b is another example where the crease curve and the handle curve are similar. The design is a reproduction of the origami artwork in the origami book [4]. From left to right in Fig. 5, each figure shows the user input (white circles are control points of the Bézier curve,the black curve are 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. 5a) or similar (Fig. 5b) 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. 5b 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.

Further, 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. For all the evaluated models, the maximum error for developability was less than the criterion if the number of divisions is 60 or more, and the maximum error for flatness was less than the criterion, regardless of the number of divisions. Since the crease curves are generally divided into hundreds to obtain a smooth appearance the resulting shapes can be seen as sufficiently developable.

A user test was also conducted to evaluate the usability of the system. The subjects were asked to create shapes similar to two shapes presented as examples, and to create another shape freely. 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. Although statistical evaluation is not possible due to the small number of subjects, the results showed that it is easy to understand the operation, but it is not easy to create the intended shape until one becomes accustomed to it. 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. 5.



(a)

(b)

Fig. 5: Developable surfaces with a curved fold generated by the proposed method.

Conclusions:

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. Resolving the problems identified as a result of user testing is a future task.

References:

[1] Fuchs, D.; Tabachnikov, S.: More on paperfolding. The American Mathematical Monthly, Vol. 106, No. 1, pp. 27–35, 1999.

[2] Tachi, T.: One-dof rigid foldable structures from space curves, In Proceedings of the IABSE-IASS Symposium, 2011, 20–23.

[3] 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.

[4] Mitani, J.: Curved-Folding Origami Design, CRC Press, 2019.