Interactive Modelling of Curved Folds with Multiple Creases Considering Folding Motions

Yuka Watanabe[1] and Jun Mitani[1]

[1]University of Tsukuba, yukakohno@hotmail.com
[1]University of Tsukuba, mitani@cs.tsukuba.ac.jp

Corresponding author: Yuka Watanabe, yukakohno@hotmail.com

ABSTRACT

A framework of a user interface system to design a curved folding and to simulate its folding motion is proposed. The system simulates the folding motion of a sheet of paper while supporting the ruling transition caused by the change in the 3D shapes of the curved creases. The creases consists of one primary crease curve and one or more additional crease curves. The primary crease curve is first designed given the curvature, the torsion, and the folding angles. Then the additional crease curves are added by drawing free form curve, approximated by B-spine curves, and refined by adjusting its control points. In this adjustment process, the crossing of the rulings are resolved and a curved fold surface is obtained. The system was evaluated visually and by numerical verification, to confirm its usability and reliability.

Keywords: Curved folding, developable surface modeling, folding motions
DOI: https://doi.org/10.14733/cadaps.2019.452-465

1 INTRODUCTION

Curved folding is one of the most popular fields of origami, which attracts people by its beautiful appearance. To support the design of such shapes, we propose a framework of the user interface system to model and visualize curved folded surfaces and their folding motions. Figure 1 shows an example of a curved fold and its folding motion in 3D and 2D space. The 3D space shows the shape of the curved fold paper approximated by a polygon model. The 2D space represents the crease pattern on a sheet of flat paper. In both 3D and 2D, rulings, which are the straight lines on the curved surface, are depicted explicitly to show the bending direction of the paper. As shown in the figures, even with a fixed crease curve, the rulings change their directions, while the paper is being folded along the folding curves. The rulings also change if the crease curve is curled or twisted in 3D space, creating various 3D shapes from the same 2D crease, under the restriction of developability. Because of this flexibility, it is not easy to predict the folded state or the folding motion of a curved crease pattern. Understanding the movement of the rulings using our user interface system helps the user to learn about the behavior of the paper, which is necessary in designing a curved fold.

The method introduced in this paper follows our previously proposed system which designs one crease curve on a sheet of paper, supporting the rulings transition [9]. The proposed prototype system allows the user to add some creases on the original curved fold and to adjust its shape to make an intended 3D model through a graphical user interface (GUI). Thus the user can design a curved fold with multiple curved creases, with a limitation that there are no intersection between creases. As in the previous system, folded paper is represented as quad strips between the fold curves and/or paper edges. When a crease is added, the rulings are divided into two by the crease, and the rulings on the folded side is updated. Then a new quad strip is generated from the updated rulings. Here, the geometry of the folded paper is derived according to the folding curve input by the user. The ruling angles are calculated from the parameters derived from the 2D and 3D shape of the curve, which are the curvature and the torsion, and the folding angle, as described in section 3.1.

However, it is sometimes difficult for a user to adjust the shape of the curve to generate existable rulings. The rulings derived from inappropriate parameter values may be crossings, which does not occur in physical paper (Fig. 2). This type of failure often happens as the rulings are easily affected by a small change in parameter values. Furthermore, the values of different parameters or a parameter in different points does affect each other. A small unintended shift on the curve affects the torsion, curvature, and the folding angles of that point, which causes a large shift in ruling directions calculated from them. Drawing a free-form curve manually is an intuitive method to design a crease curve, but it often end up in severe crossings of rulings. It is especially difficult to add a new crease to a curved surface without having the rulings crossing.

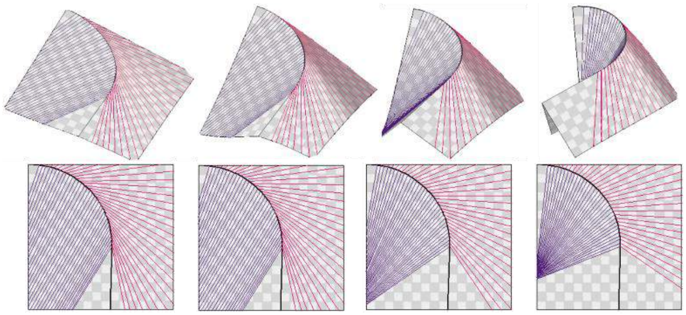
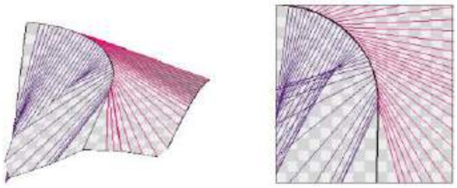
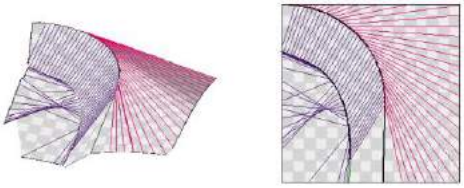


Figure 1: Transition of rulings during fold and twist motion. (top: 3D space, bottom: 2D space).



(a) Crossing caused by one crease



(b) Crossing caused by the added crease

Figure 2: Crossing of rulings. (Left: 3D space, Right: 2D space).

To support the design of developable and consistent curved surface, our method allows the user to add and adjust the curved crease while observing the calculated rulings and the paper shape instantly. The system also provides functions to restrict the movement of the user-adjusted curve, or to optimize the shape of the curve, according to some cost functions, to reach user intended shape effectively.

2 RELATED WORKS

In recent studies on paper modelling and simulation, 3D surfaces generated by curved folding are often represented by a polygon model of fixed geometry composed of a group of quad strips and planar polygons. Kilian et al. proposed a method to reconstruct a 3D polygon mesh of a developable surface from 3D data of a curved folding as an input, by initially fitting planar polygons on the 3D shape and then optimizing their shapes and orientations [2]. Their method is successful in generating the developable surface of a user-intended shape but does not design nor consider its folding motion. Tang et al. introduced an interactive design system of a curved-creased origami, which defines each surface by a pair of spline curves and solves the constraints for its global developability through iterative process [8]. Contrary to their robust but complex method, 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 [4]. As the folding curve is defined as the intersection of the surface and a plane specified by the user, their method is limited to the curved fold composed of plane curves where the rulings do not change before and after the fold.

As to the folding motion, Tachi developed a software to simulate the folding motion of the paper using the rigid folding method. His method represents paper by planar polygons with fixed geometry and mainly deals with straight fold lines [5]. He have also proposed a design method for flat-foldable vault structures composed of flat-foldable tubes assembled by welding two sheets [6][7]. While dealing with the folding motion of the curved crease, this structure restricts the folding angles to be constant throughout the curve, which makes the rulings to be fixed, and the model is folded rigidly. It does not support deformation where the directions of the rulings change while the paper is being folded. Another method to simulate a sheet of thin materials by adaptive mesh refinement [3] models irregular deformations of paper such as crumpled paper or ruling transitions, but it results in complicated mesh data composed of thousands of faces.

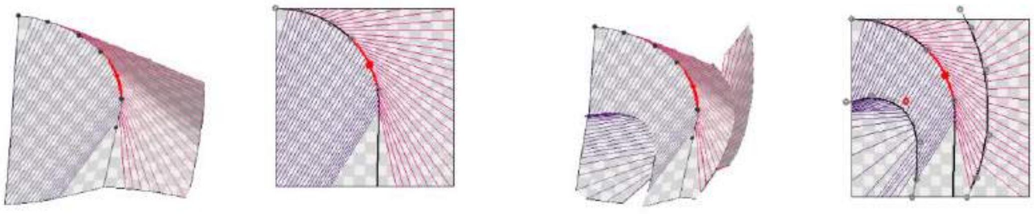
In our method, the paper is modelled as a quad strip with relatively small number of faces. In interactive design and the simulation of folding motion, the 3D shape of the paper is calculated instantly according to the shape and the folding angles of the curve, with no iterative process. Unlike in the surface subdivision method, our approach allows the deformation of paper by updating the ruling directions, which makes the number of quads and their connectivity to be consistent. In adding a crease curve, the user can input the new crease without any restrictions, such as planar curves, and then adjust its shape with the feedback of the system.

3 DESIGNING CURVED FOLDS

3.1 Main Idea

3.1.1 User input

Our system generates the 3D shape of the curved folded paper from one primary crease curve and some additional crease curves which is input by the user through a GUI (Fig. 3). The shape of the primary crease curve is defined according to the curvature in 2D and 3D, the torsion, and the folding angles on the curve. The user inputs and adjusts the values of the parameters on some control points on the crease curve, which is then interpolated throughout the curve to make an arc-length parametric function. With the primary crease curve, a 3D developable surface with a single curved fold is generated. Then, the user places some additional crease curves by first drawing the curves on the 2D space and then adjusting the position of the control points of the B-spine curve approximating the curve drawn. Note that the "control points" indicate different type of points for primary and additional crease curves. The user may put more than one additional curves on either side of the primary curve, but the one needs to make sure that there is no intersection of crease curves as our system does not support such case.



(a) Primary curved crease (b) Additional curves

Figure 3: The primary and the additional curves. (Left: 3D space, Right: 2D space).

3.1.2 Geometry of the curved fold

The geometry of curved folds and surfaces, or sets of quad strips, are defined by the shape of the crease curve, folding angle, and the rulings. They are calculated based on the following formula, with some of the parameters and vectors indicated in Fig. 4. The ruling angles $\beta_L(s)$, $\beta_R(s)$ are the angles between the tangent vector $\mathbf{T}(\boldsymbol{s})$ and the ruling vectors on the left and the right side. The ruling vectors $\boldsymbol{r}_L(s)$, $\boldsymbol{r}_R(s)$ indicates the orientations of the rulings in 3D space, which is part of the quad strips. The folding angle $\alpha(s)$ is the angle between the osculating plane of the curve and the ruling vectors projected on the normal plane of the curve. The tangent vector $\mathbf{T}(\boldsymbol{s})$, the normal vector $\mathbf{N}(\boldsymbol{s})$, and the binormal vector $\mathbf{B}(\boldsymbol{s})$ are calculated by (3.1) using the 3D curve $\mathbf{X}(s)$, with s indicating the arc-length parameter.

$$\begin{bmatrix} \boldsymbol{T}(s) \\ \boldsymbol{N}(s) \\ \boldsymbol{B}(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}'(s) \\ \boldsymbol{X}''(s)/k(s) \\ \boldsymbol{X}'(s) \times \boldsymbol{X}''(s)/k(s) \end{bmatrix}. \tag{3.1}$$

The relation between the vectors $\mathbf{T}(\boldsymbol{s})$, $\mathbf{N}(\boldsymbol{s})$, $\mathbf{B}(\boldsymbol{s})$ and the parameters, the curvature $k(s)$ and the torsion $\tau(s)$ of the 3D curve, are given by (3.2), which is known as Frenet–Serret formula.

$$\begin{bmatrix} \boldsymbol{T}'(s) \\ \boldsymbol{N}'(s) \\ \boldsymbol{B}'(s) \end{bmatrix} = \begin{bmatrix} 0 & k(s) & 0 \\ -k(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{T}(s) \\ \boldsymbol{N}(s) \\ \boldsymbol{B}(s) \end{bmatrix}. \tag{3.2}$$

Given the parameters, ruling angles $\beta_L(s)$, $\beta_R(s)$ and the folding angles $\alpha(s)$ are calculated by the following equations, induced by Fuchs and Tabachnikov and introduced by Tachi in his work [1][7].

$$k_{2D}(s) = k(s)\cos\alpha(s), \tag{3.3}$$

$$\cot\beta_L(s) = \frac{-\alpha(s)' + \tau(s)}{k(s)\sin\alpha(s)}, \tag{3.4}$$

$$\cot\beta_R(s) = \frac{\alpha(s)' + \tau(s)}{k(s)\ \sin\alpha(s)}, \tag{3.5}$$

where $k_{2D}(s)$ is the curvature of 2D crease curve. Note that the signs of some parameters are different from that of the references, as the definition of the vectors are in opposite orientation. Finally the ruling vectors $\boldsymbol{r}_L(s)$, $\boldsymbol{r}_R(s)$ in 3D space is calculated as follows.

$$\boldsymbol{r}_L(s) = \cos\beta_L(s)\,\boldsymbol{T}(s) - \sin\beta_L(s)\cos\alpha(s)\,\boldsymbol{N}(s) + \sin\beta_L(s)\sin\alpha(s)\,\boldsymbol{B}(s), \tag{3.6}$$

$$\boldsymbol{r}_R(s) = \cos\beta_R(s)\,\boldsymbol{T}(s) + \sin\beta_R(s)\cos\alpha(s)\,\boldsymbol{N}(s) + \sin\beta_R(s)\sin\alpha(s)\,\boldsymbol{B}(s). \tag{3.7}$$

As for the additional crease curves, folded surface is defined by the shape of the 3D crease curve, folding angle, and the ruling vectors on the new surface. The 3D shapes of the curves are obtained as the 2D curves projected on to the 3D curved surface. Let the vectors, the parameters, and the folding angles of the j-th additional crease curve on the left side of the primary crease curve be denoted as $\mathbf{T}^{Lj}(s)$, $\mathbf{N}^{Lj}(s)$, $\mathbf{B}^{Lj}(s)$, $k^{Lj}(s)$, ${k_{2D}}^{Lj}(s)$, $\tau^{Lj}(s)$, and $\alpha^{Lj}(s)$. The folding angle $\alpha^{Lj}(s)$ is calculated as the angle between the normal vector $\mathbf{N}^{Lj}(s)$ and the ruling vector ${\boldsymbol{r}_R}^{Lj}(s)$ projected to the normal plane of the additional crease curve. ${\boldsymbol{r}_R}^{Lj}(s)$ is identical to $-\boldsymbol{r}_L(s)$ as they lie in the same ruling but defined as the opposite orientation. The new ruling vector ${\boldsymbol{r}_L}^{Lj}(s)$ on the other side of the additional crease curve is calculated using equations (3.4)-(3.7). The same method is applied to the additional crease curve on the right side of the primary curve, using vectors and parameters indicated as $\mathbf{T}^{Rj}(s)$, $\mathbf{N}^{Rj}(s)$, $\mathbf{B}^{Rj}(s)$, $k^{Rj}(s)$, ${k_{2D}}^{Rj}(s)$, $\tau^{Rj}(s)$, and $\alpha^{Rj}(s)$.

To interpret the formula, equation (3.3) shows that the curvature in 2D $k_{2D}(s)$, the curvature in 3D $k(s)$, or the folding angle $\alpha(s)$ could be calculated given other two parameters. So, the user needs to set only two of the three parameters and the last parameter is calculated automatically. Equation (3.4) and (3.5) indicates that

- If the folding angle $\alpha(s)$ is constant and torsion $\tau(s)$ is zero, the ruling angles $\beta_L(s)$, $\beta_R(s)$ are $\pi/2$, or perpendicular to the tangent vector. Otherwise, larger value for torsion $\tau(s)$ leads the ruling angles $\beta_L(s)$, $\beta_R(s)$ to slant in the same direction, and the larger increase or decrease in folding angle $\alpha'(s)$ makes the rulings in the left and right side slant in opposite directions.
- The larger value in 3D curvature $k(s)$ or the folding angle $\alpha(s)$, which range between 0 to $\pi/2$, leads the ruling angles to be close to $\pi/2$.

These features are difficult to understand intuitively, but in our system, the user can control the parameter values empirically by adjusting the curve shape according to the instant visual feedback.

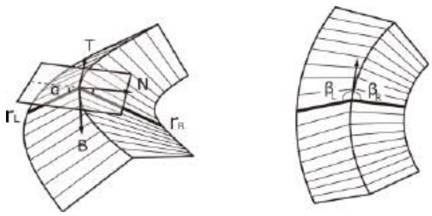


Figure 4: The definitions of folding angle α, ruling angles β_L, β_R, ruling vectors $\boldsymbol{r}_L(s)$, $\boldsymbol{r}_R(s)$, tangent vector $\mathbf{T}$, normal vector $\mathbf{N}$, and binormal vector $\mathbf{B}$. (Left: 3D space, Right: 2D space).

In the actual calculation process of our system, a curve is replaced with a series of vertices as the curved folded surface is represented as quad strips. The discrete version of the geometry calculation described above is now explained. Instead of arc-length parametric functions, the vectors, the parameters, and the folding angles are defined on the vertices $\{\boldsymbol{X}_i\}$ instead of the 3D curve $\mathbf{X}(s)$, and are indicated as $\{\boldsymbol{T}_i\}$, $\{\boldsymbol{N}_i\}$, $\{\boldsymbol{B}_i\}$, $\{k_i\}$, $\{{k_{2D}}_i\}$, $\{\tau_i\}$, and $\{\alpha_i\}$. The derivative of $\boldsymbol{X}_i$, or $d\boldsymbol{X}_i$, used in equation (3.1), is calculated as follows, using three consecutive vertices $\{\boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \boldsymbol{X}_{i+1}\}$,

$$d\boldsymbol{X}_i = \frac{1}{2}\left(\frac{\boldsymbol{dX}_{i,i-1}}{|\boldsymbol{dX}_{i,i-1}|} + \frac{d\boldsymbol{X}_{i+1,i}}{|d\boldsymbol{X}_{i+1,i}|}\right), \tag{3.8}$$

where $\boldsymbol{dX}_{i,i-1} = \boldsymbol{X}_i - \boldsymbol{X}_{i-1}$ is a vector connecting two vertices. Similarly, derivatives of the vectors $d\boldsymbol{T}_i$, $d\boldsymbol{N}_i$, and $d\boldsymbol{B}_i$, used in equation (3.2), are calculated as

$$\begin{cases} d\boldsymbol{T}_i = \dfrac{1}{2}\left(\dfrac{\boldsymbol{T}_i - \boldsymbol{T}_{i-1}}{|\boldsymbol{dX}_{i,i-1}|} + \dfrac{\boldsymbol{T}_{i+1} - \boldsymbol{T}_i}{|d\boldsymbol{X}_{i+1,i}|}\right) \\ d\boldsymbol{N}_i = \dfrac{1}{2}\left(\dfrac{\boldsymbol{N}_i - \boldsymbol{N}_{i-1}}{|\boldsymbol{dX}_{i,i-1}|} + \dfrac{\boldsymbol{N}_{i+1} - \boldsymbol{N}_i}{|d\boldsymbol{X}_{i+1,i}|}\right), \\ d\boldsymbol{B}_i = \dfrac{1}{2}\left(\dfrac{\boldsymbol{B}_i - \boldsymbol{B}_{i-1}}{|\boldsymbol{dX}_{i,i-1}|} + \dfrac{\boldsymbol{B}_{i+1} - \boldsymbol{B}_i}{|d\boldsymbol{X}_{i+1,i}|}\right) \end{cases} \tag{3.9}$$

which is an average of two differences between the consecutive vertices normalized by the distance. This normalization term is especially important for the additional crease curve whose vertices are not spaced evenly. As in the continuous form, vectors $\boldsymbol{T}_i$, $\boldsymbol{N}_i$, $\boldsymbol{B}_i$, curvature k_i, and torsion τ_i are calculated from the discrete 3D curve $\{\boldsymbol{X}_i\}$, using the following equations. According to equation (3.1) and the fact that $\boldsymbol{T}_i$, $\boldsymbol{N}_i$, $\boldsymbol{B}_i$ are unit vectors,

$$\begin{cases} \boldsymbol{T}_i = \dfrac{d\boldsymbol{X}_i}{|d\boldsymbol{X}_i|} \\ \boldsymbol{N}_i = \dfrac{d\boldsymbol{T}_i}{|d\boldsymbol{T}_i|} \\ \boldsymbol{B}_i = \boldsymbol{T}_i \times \boldsymbol{N}_i \end{cases} . \tag{3.10}$$

Derived from equation (3.2), or $d\boldsymbol{T}_i = k\boldsymbol{N}_i$, $d\boldsymbol{N}_i = -k\boldsymbol{T}_i + \tau\boldsymbol{B}_i$, $d\boldsymbol{B}_i = -\tau\boldsymbol{N}_i$ in discrete form, and that $\boldsymbol{N}_i$ and $\boldsymbol{B}_i$ are unit vectors, curvature k_i and torsion τ_i are calculated by

$$k_i = |d\boldsymbol{T}_i|, \tag{3.11}$$
$$\tau_i = |d\boldsymbol{N}_i + k_i\boldsymbol{T}_i| = |-d\boldsymbol{B}_i|. \tag{3.12}$$

3.2 The Design Procedure

This subsection explains the procedures of designing curved folds. The primary curve is first designed to generate a developable surface with one crease curve [9]. Then additional curves are added to the curved surfaces to generate new creases. The shapes of the additional crease curves are adjusted by the user with the support of the system to resolve the ruling crossing.

3.2.1 Design of primary crease curve

The primary crease curve is defined by the user by specifying the curvature $k(s)$, $k_{2D}(s)$, torsion $\tau(s)$ and/or folding angle $\alpha(s)$ at some control points on the curve. An example of primary crease curve with the GUI is shown in Fig. 5. In our method, seven control points are placed on the primary crease curve, two on the ends of the curve and the rest of the points placed evenly between the two ends. The number of the control points may be changed. The position and the orientation of the starting point of the curve, initially set to be on the top left corner of the paper and in the direction of left to right, may be changed by the user through mouse drag. To design the curve shape, the user first selects "CONTROL MODE" and "PARAMETER" to choose which parameters should be controlled by the user, and which one to be calculated according to the other parameters. Because the parameters are mutually dependent, the change in one parameter causes the other parameters to change so that it meets equation (3.3). The user then select a control point by the scroll bar "CONTOL POINT INDEX" and adjusts the parameter value of that point by the scroll bar "PARAMETER VALUE". Internally, the input parameter values at the control points are interpolated by the spline curve of degree three to all vertices $\{\boldsymbol{X}_i\}$ on the curve, and then the other parameters are calculated on the vertices. Using the parameters, the ruling angles are obtained by equation (3.4)-(3.7). With the 3D curve $\{\boldsymbol{X}_i\}$ and the ruling vectors in 3D space $\{\boldsymbol{r}_{L_i}\}$, $\{\boldsymbol{r}_{R_i}\}$, the 3D shape of the curved fold surface is obtained, which means the pose and the position of each quad in the quad strips is defined. The 2D and 3D surfaces composed of the rulings are shown to the user as a prompt feedback. In this step the rulings are calculated as it is, according to the user input parameters, which means they may intersect with other rulings. If needed, the user adjusts the parameter values of different control points and different parameters alternately to design the desired shape and to resolve the rulings crossing.

3.2.2 Design of additional crease curve

Additional crease curves are initially drawn by the user as free-form curves on the 2D space (Fig. 6 (a)). This operation is to be done in curve adding mode which is activated by checking the check box "FOLD". Now, we will explain the internal process using the denotation of the crease on the right side of the primary crease curve. To make the curve smooth, each curve is approximated by the B-spline curve of degree-four, with six control points and the knot vector $\{0,0,0,0,1,2,3,3,3,3\}$. It is then discretized by plotting the intersections of the rulings and the curve as the vertices (Fig. 6 (b)). By projecting the vertices onto the 3D ruling vectors $\{\boldsymbol{r}_{\boldsymbol{R}_i}\}$, a 3D space curve of the additional crease curve $\{\mathbf{X}^{Rj}{}_i\}$ is obtained. The curvature $\{k^{Rj}{}_i\}$, the torsion $\{\tau^{Rj}{}_i\}$, the tangent vector $\{\mathbf{T}^{Rj}{}_i\}$, the normal vector $\{\mathbf{N}^{Rj}{}_i\}$, and the binormal vector $\{\mathbf{B}^{Rj}{}_i\}$ are then calculated from the 3D curve by equation (3.8)-(3.12). The folding angles $\{\alpha^{Rj}{}_i\}$ are calculated as the angle between the normal vector $\mathbf{N}^{Rj}{}_i$ and the ruling $\boldsymbol{r_L}^{Rj}{}_i$ projected to the normal plane. As described in 3.1.2, the ruling $\boldsymbol{r_L}^{Rj}{}_i$ is obtained as $-\boldsymbol{r}_{R_i}$ because the ruling on the left side of the additional crease curve $\{\mathbf{X}^{Rj}{}_i\}$ is same as the rulings on the right side of the primary crease curve with the opposite orientation. At last, the ruling angles $\{\beta_R{}^{Rj}{}_i\}$ on the new surface on the other side of the additional crease curve is calculated using equations (3.4) and (3.5), and the 3D ruling vectors $\{\boldsymbol{r_R}^{Rj}{}_i\}$ by equations (3.6) and (3.7). In this step, the new rulings are likely to have crossings (Fig. 6 (c)), which are to be eliminated in the next step. If two or more additional crease curves are input, the creases are processed one by one starting from the closest one to the primary curve.

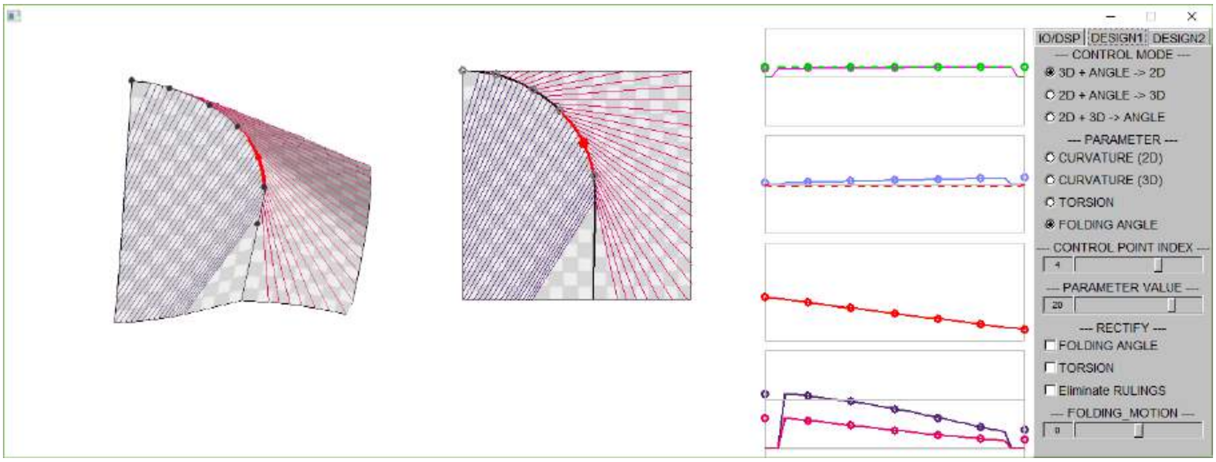
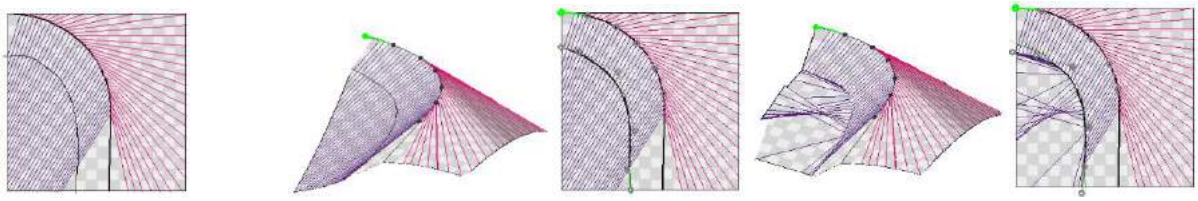


Figure 5: Design of primary crease curve. (Left: 3D space, Center: 2D space, Right: Graph of parameter values on the control points and along the curve) The red dot in 3D space and crease pattern indicates the control point whose parameter being adjusted, and gray dots shows the other control points. The graphs are the curvature in 3D and 2D (top row, pink line and green dotted line), the torsion and the differentiation of folding angle (second row, blue line and red dotted line), folding angle (third row, red line), and ruling angles (fourth row, in pink and purple, the same color as the rulings in 2D and 3D space).



(a) User drawn free form curve (b) B-spline curve approximation (c) Additional crease curve

Figure 6: Design procedure of additional curve. (Left: 3D space, Right: 2D space).

3.2.3 Adjustment of additional crease curve

To resolve the crossing of the rulings, the shape of the additional curve is adjusted by moving the control points of the B-spline curve. Figure 7 shows an example of the additional curve in initial state and the user adjusted state. The user can select and drag the control points in the display window, so that the curved crease and the rulings are in the intended shape. While the control points are moved, the parameters of the curve and the rulings are re-calculated and the 2D and 3D shape are shown to the user promptly. We propose following three editing methods to help the user adjust the control points effectively.

(a) The user edit the control points with no system support.
The user can move the control points on 2D space freely with no restriction. An inappropriate movement could make the state worse and cause the rulings to cross, but this may be good enough for an experienced user.

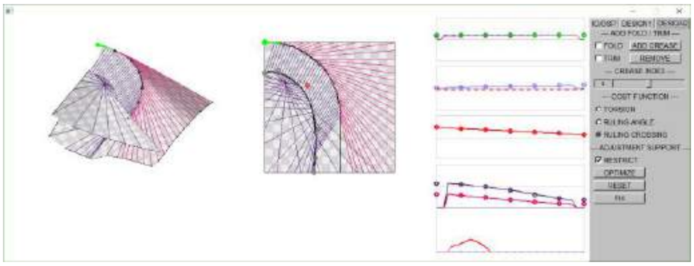
(b) The user edit the control points with system support to encourage better state.
The user can move the control points with some restrictions on the movement. The restriction is based on the cost functions explained in the next subsection. The control points can be moved according to the user's mouse drag only if the new position results in smaller cost than the previous state. This mode becomes valid with the checkbox "RESTRICT" being checked.

(c) The optimization process based on the cost function.
When an additional crease is chosen on the scroll bar "CREASE INDEX" and the button "OPTIMIZE" is pressed, the optimization process is executed. Each control point of the curve is shifted by small amount in different directions and evaluated by the cost function. The control point positions with the smallest cost are chosen. This process is iterated until the control point positions are converged or the maximum number of iteration is carried out.

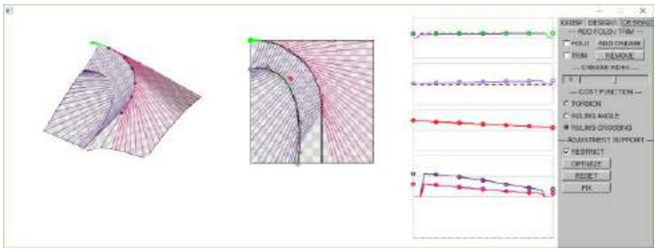
While moving the control points in (a) and (b), the system shows the user the 2D and 3D curved folds with rulings, the graphs of the parameters, and the current cost distribution along the curve. One may also check the folding motion, as written in subsection 3.2.4, and then come back to this step again. The user may also trim the paper to eliminate the rulings crossing instead of resolving them. This is done by choosing "TRIM" instead of "FOLD" and drawing a curve (Fig. 8).

3.2.4 Simulation of folding motion

After the design and the adjustment of the curved fold, the folding motion of the paper is simulated. Figure 9 shows three states of a curved fold with an additional crease on each side of the primary curved crease, which are in different folding angles during the folding motion. In the folding motion, the folding angle $\{\alpha_i\}$ of the primary crease curve are changed linearly from its original value to $\pi/2$, where the paper is completely folded. On additional crease curves, the 3D curves, the folding angles, and the rulings on the other side are recalculated as described in section 3.2.2. In the unfolding motion, the folding angle $\{\alpha_i\}$ and the torsion $\{\tau_i\}$ of the primary crease curve are changed linearly from their original value to 0, the flat state. Given appropriate additional creases by the user, the folding and unfolding motion with ruling transition are simulated successfully to some extent. The crossing of the rulings may occur at some point of the motion, as a result of ruling transition. In the folding motion, because the curved crease could never be fold completely, it must be stopped at some point of the motion. This is where the folding angles becomes the maximum for the given crease pattern.



(a) Additional curve in initial state



(b) Additional curve in the user adjusted state

Figure 7: Adjustment of additional curve in GUI. (Left: 3D space, Center: 2D space, Right: Graphs of parameter values along the additional crease curve, control panel) The graph in red on the bottom row are the current costs along the curve.

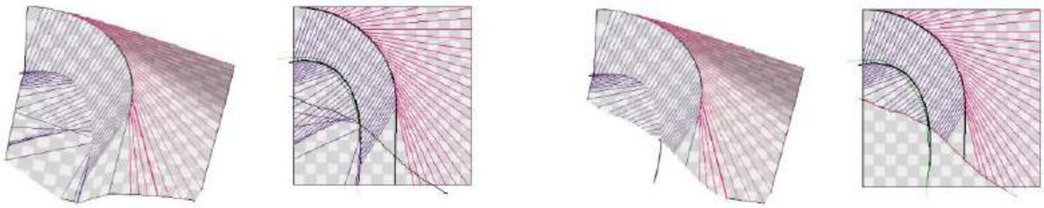
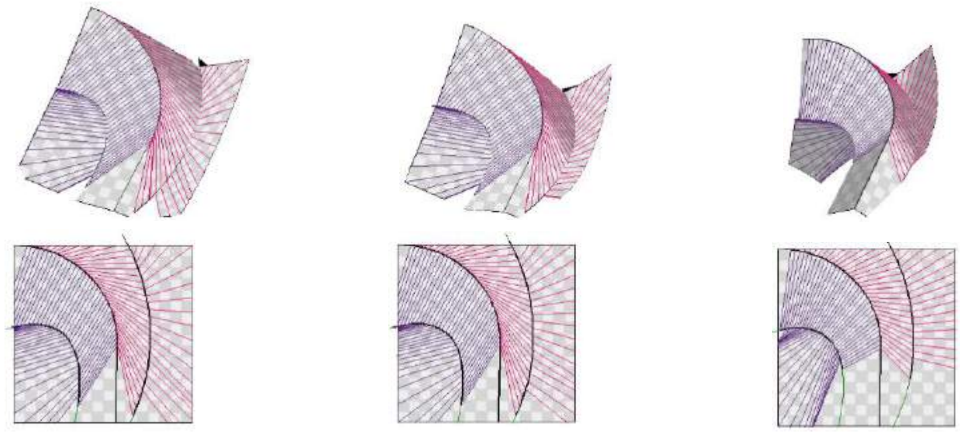


Figure 8: Trimming of the Paper. (Left: 3D space, Right: 2D space)



Folding angle: (a) 7 to 26 degrees (b) 11 to 40 degrees (c) 38 to 57 degrees

Figure 9: Folding motion with additional crease curve. (top: 3D space, bottom: 2D space).

3.3 Cost Functions

Our system provides three types of cost functions used in the editing method (b) and (c) of section 3.2.3. These cost functions are designed empirically, using the indices described below.

i. The difference of torsion between the primary crease curve and the additional crease curve, calculated on the vertices between starting point *si* and the ending point *ei*.

$$Cost_{\tau} = \frac{1}{ei - si} \sum_{i=si}^{i<ei} \left| \tau_i - \tau^{Rj}{}_i \right|. \tag{3.13}$$

ii. The difference of left and right ruling angles.

$$Cost_{\beta} = \frac{1}{ei - si} \sum_{i=si}^{i<ei} \left| \beta_L{}^{Rj}{}_i - \beta_R{}^{Rj}{}_i \right|. \tag{3.14}$$

iii. The total area on 2D crease pattern where the projection to the 3D space is non-injective due to ruling crossing, as shown in Fig. 2.

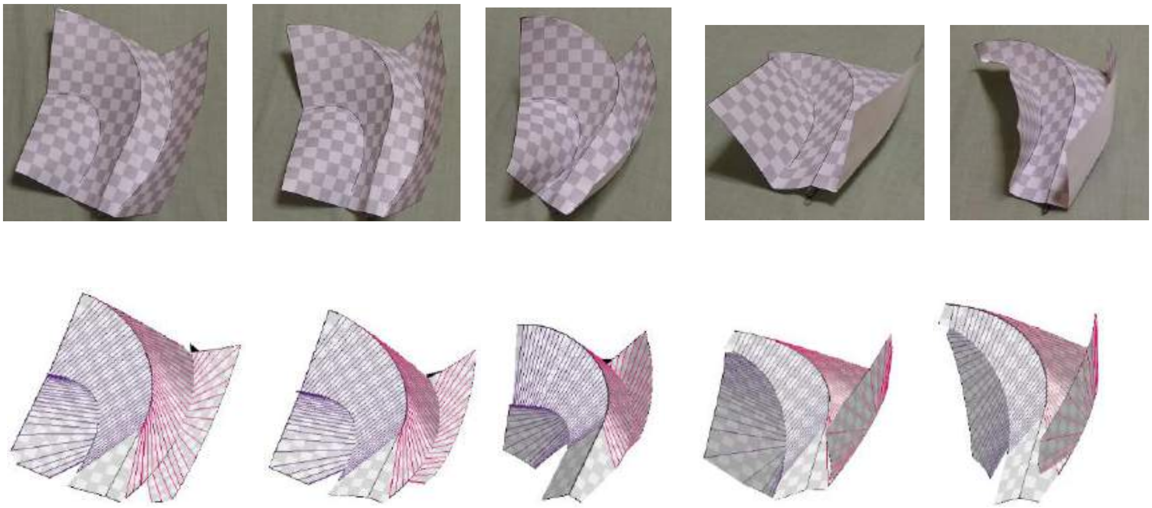
The user can select the cost function through the GUI anytime during the adjustment process. The user may want to change the cost function alternately as well as the editing mode. In that case, the current smallest cost is managed and updated separately for each cost function.

4 RESULTS AND DISCUSSIONS

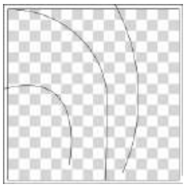
As an evaluation of our system, visual comparison with real paper and the numerical check of its developability were performed. Then some examples of curved folding, designed and simulated by our system, are presented.

4.1 Visual Comparison

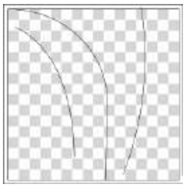
The results of visual comparison are shown in Fig. 10. (a)-(c) are the shapes introduced in Fig. 9 and share the same crease pattern shown in (f). The crease pattern of (d) and (e) is shown in (g). The curved fold of the real paper, in the top row, is made by folding the crease pattern and fixing the shape of primary curve using a piece of wire on the back of the paper. The 3D model in the second row are missing some faces where the rulings are not defined. Overall result seems acceptable to say the 3D model represents the developable surface.



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



(f) Crease Pattern for (a)-(c)



(g) Crease Pattern for (d),(e)

Figure 10: Visual comparison with real paper.

4.2 Numerical Evaluation

To evaluate the developability of the model, we measured (A) the sum of corner angles adjacent to each vertex on the curves and (B) the flatness of each quad composing the curved surface. As for (A), each corner angles was calculated as the angle between the ruling and the tangent vector of the curve in 3D space. The angle sum was evaluated by the difference between 2π, as the angle sum should always be 2π for perfectly developable surface. (B) was calculated as the distance between one vertex of a quad and a plane passing the rest of the vertices. The results are shown in Tab. 1 and Tab. 2.

With small errors, we could still say that the model is developable and is able to represent a sheet of paper with curved folds.

Fold. Ang.	*(a) 7 to 26 degrees*		*(b) 11 to 40 degrees*		*(c) 38 to 57 degrees*	
Curved Crease	*Left*	*Right*	*Left*	*Right*	*Left*	*Right*
Average	0.101	0.000782	0.00044	0.106	0.0398	0.000426
Maximum	2.12	0.0159	0.0012	2.11	1.14	0.00189

Table 1: The Difference between Sum of Corner Angles and 2π. (Units in Degrees).

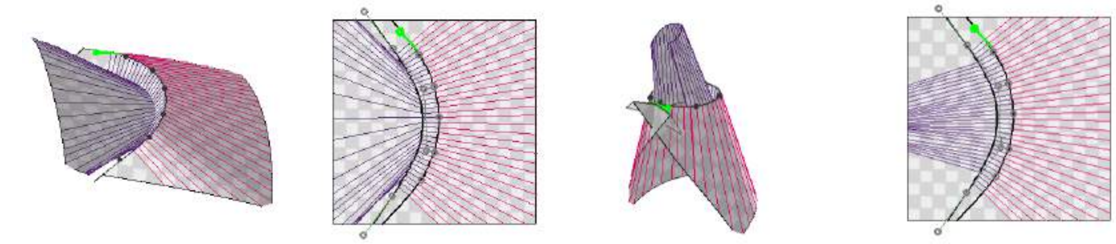
Fold. Ang.		*(a) 7 to 26 degrees*		*(b) 11 to 40 degrees*		*(c) 38 to 57 degrees*	
Quads|Curved Crease		*Left*	*Right*	*Left*	*Right*	*Left*	*Right*
Left	Average	0.00657	0.00166	0.00888	0.00254	0.0393	0.00233
	Maximum	0.0268	0.00208	0.0366	0.00322	0.566	0.00383
Right	Average	0.000119	0.00545	0.00014	0.00542	0.00102	0.00934
	Maximum	0.000311	0.049	0.000318	0.0199	0.0194	0.0353

Table 2: Flatness of quads. (Unit in mm. Note that the length of the paper edge being 200mm).

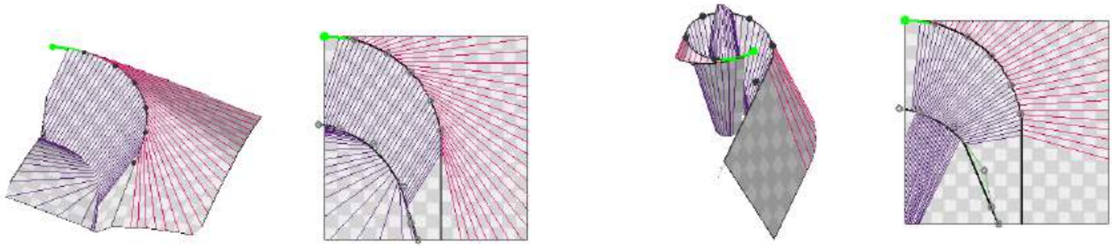
4.3 Example results

Some examples of curved folds designed by our system are shown in Fig.11 and Fig. 12. The examples include the curved folds with (a) a primary crease curve having zero torsion and constant folding angle and (b) a primary crease curve with non-zero torsion and increasing or decreasing folding angle along the curve. For each example, two steps of the folding motion are shown, with ruling angles changing while the 2D crease curves are consistent. As shown in Fig.11, the results with one additional crease curve are likely to be successful. But, as in Fig. 12, with more than one additional creases on one side of the primary curve, the problem becomes much more difficult for the second curve, even with the support of the system. In case of Fig. 12 (b), after a great effort to add and refine the second additional crease curve, the surface quickly collapsed during the folding motion.

Figure 13 demonstrates example results of the optimization process using three types of cost functions introduced in section 3.3. The original curve drawn by the user causes irregular crossings of the rulings on the newly folded surface (Fig. 13 (a)). After the convergence of the optimization process, each result shows the rulings aligned according to the cost function (Fig. 13 (b)-(d)). In this case, only (d) gives an existable surface. But other two types of cost functions are also useful in the intermediate stage of resolving the ruling crossing, to change the ruling angles to take in the user intensions or to escape from a local minimum state.

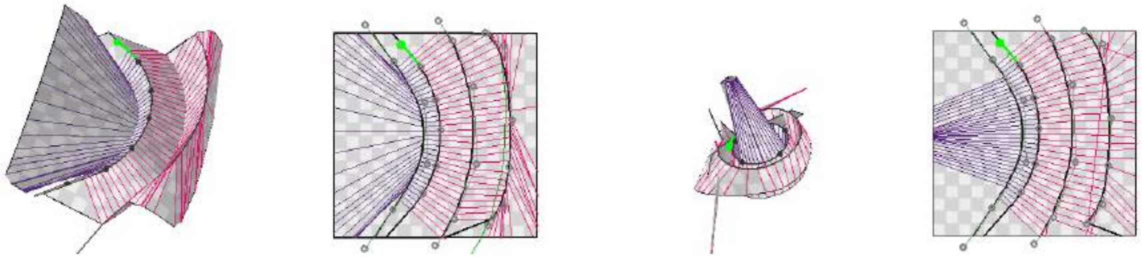


(a) Primary crease curve with zero torsion and constant folding angle.

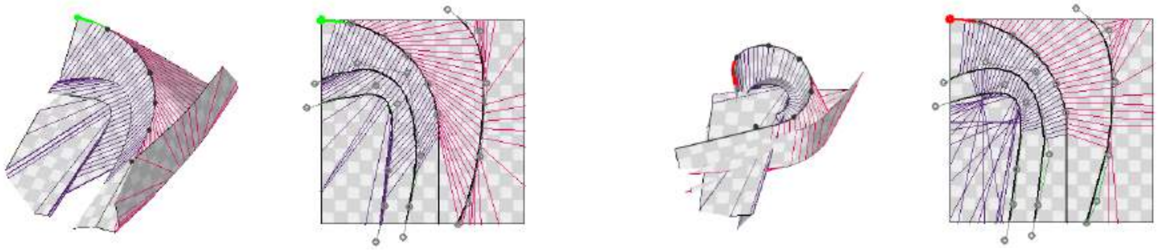


(b) Primary crease curve with non-zero torsion and increasing or decreasing folding angle.

Figure 11: Examples of curved folding.

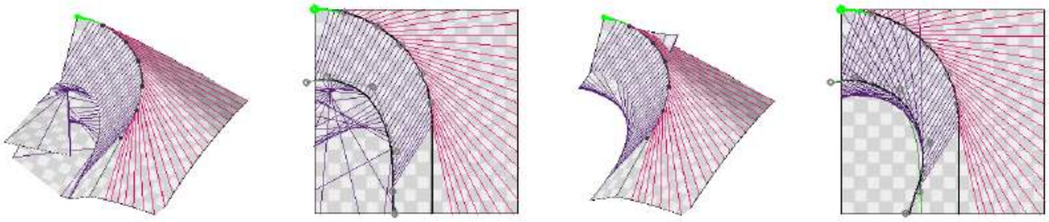


(a) Primary crease curve with zero torsion and constant folding angle.

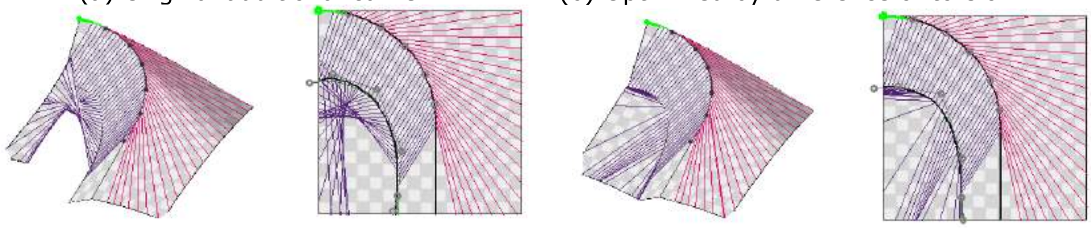


(b) Primary crease curve with non-zero torsion and increasing or decreasing folding angle.

Figure 12: Curved folding with multiple additional curves.



(a) Original additional curve (b) Optimized by difference of torsion



(c) Optimized by the difference of ruling angles (d) Optimized by total area of ruling crossing

Figure 13: Results of optimization.

5 CONCLUSION

In this work, we proposed a framework for designing a surface with multiple curved folds which allows the transition of rulings during the fold motion. To resolve the crossing of the rulings caused by the additional crease curves, the system provides the GUI to adjust the control points of the B-spline curve approximating the 2D curve. The system also supports the input restriction and optimization process guided by three types of cost functions, calculated using torsion, ruling angles, and the area of rulings crossing. The 3D shapes generated by the method were evaluated by the visual comparison and some indices to show its developability.

As future work, there is room for improvement in user interface to help users design intended and consistent surface more easily and effectively. As for improvements in the internal process, better cost functions should be investigated, along with the algorithm to refine the curve shape according to the rulings on all parts of the paper, instead of just the ones adjacent to the curve. At last, the evaluation of the GUI system should be done by subject experiment.

Yuka Watanabe, http://orcid.org/0000-0002-1362-1274
Jun Mitani, http://orcid.org/0000-0002-1596-844X

REFERENCES

[1] Fuchs, D.; Tabachnikov, S.: More on Paperfolding, The American Mathematical Monthly, 106(1), 1999, 27-35. https://doi.org/10.2307/2589583

[2] 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

[3] 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

[4] Mitani, J.; Igarashi, T.: Interactive Design of Planar Curved Folding by Reflection, Pacific Conference on Computer Graphics and Applications - Short Papers, 2011, 77–81.

[5] Tachi, T.: Simulation of rigid origami, Origami 4: The Fourth International Conference on Origami in Science, Mathematics, and Education, 2009, 175–187.

[6] Tachi, T.: One-DOF Rigid Foldable Structures from Space Curves, in Proceedings of the IABSE-IASS Symposium, 2011.

[7] Tachi, T: Composite Rigid-Foldable Curved Origami Structure, in Proceedings of Transformables, 2013.
[8] 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
[9] Watanabe, Y.; Mitani, J.: Modelling the Folding Motions of a Curved Fold, The 7th International Meeting on Origami in Science, Mathematics and Education (7OSME), 2018, to appear.