



Fitting Single Crease Curved-Fold Model to the User Specified Points

Yuka Watanabe¹  and Jun Mitani² 

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

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

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

Abstract. We propose an approach to realize an intuitive manipulation of a developable surface with a single curved fold. The surface shape is modelled as discrete planar quad strips, changing their geometries according to the ruling directions on the curve surfaces. The user manipulates the shape by specifying some control points on the curved surfaces and their target positions. The surface shape is then deformed to approximate the target positions using the ruling-based optimization method, minimizing the displacement of the surface control points to their corresponding target points while maintaining the developability. The evaluation shows that our method is effective in deforming the curved fold model to fit the target points.

Keywords: Curved Folding, Developable Surface Modeling, Optimization.

DOI: <https://doi.org/10.14733/cadaps.2022.387-404>

1 INTRODUCTION

Developable surfaces have an ideal property in manufacturing that its shape is formed from a flat sheet of material only by bending, without stretching nor contraction. Due to its industrial usefulness, there has been extensive research on the design of developable surfaces. Curved folding is also made from a flat sheet by folding it along the curved crease. However, controlling the shapes including a fold is a difficult problem, and much work remains to be done. While the direction of the surface bending may change in many ways, the two curved surfaces adjacent via the fold must be consistent with one another, sharing a boundary curve which must be flattened when unfolded. On the other hand, it is insufficient to model the shape as a fixed polygon model because the rulings, which are the straight lines on a developable surface, may change its directions during the folding motion, as shown in Figure 1 middle and bottom row. Other than the common paper folding, there are projects to fold a metal plate along a curved crease using machines, such as RoboFolds [1]. To deal with such materials and to automate the process, it is essential to understand, model, and to predict its folding motion.

To support such work, we have developed a sequence of interactive graphical modeling systems (Figure 2) [19],[20],[21]. Figure 2 shows a system from our previous works where the 3D polygon

is depicted in a 3D pane, and the 2D space representing a flattened sheet of paper, with the curved crease and the rulings projected, on the 2D pane. The user can edit the shape interactively by changing the parameters of the curve; the curvature, the torsion, and the folding angle, at the curve control points evenly placed on the curve, shown as small dots on the crease curve in Figure 2. These parameters are interpolated throughout the curve to define the shape of the curve, the ruling directions, and the shapes of the adjacent curved surfaces. While editing the parameters, the user can check the calculated 3D shape to make sure it is reshaped as intended. The user also needs to adjust the parameters so that the generated rulings are not intersecting in the 2D space. The ruling intersections are numerically possible but does not occur in the real world, and the user needs to avoid them manually. This interactive manipulation is not intuitive since the user cannot change the 3D shape directly, requiring some experience.

In our new system, we developed more intuitive and goal-oriented user interface which fits the curved surfaces to some target positions. Given the curved-fold model with a curved crease drawn on the 2D paper, folded in some initial state such as in Figure 2, the user controls the target 3D shape by specifying some control points on the surface and the corresponding 3D target points, as shown in Figure 3. The system reshapes the curved-folded surface by optimizing the torsion and the folding angle to minimize the distances between the surface control points and the target points, while the 2D curved crease on the paper is fixed. For efficiency, our optimization method is performed in two steps in every iteration. First, the directions of the rulings on the 2D paper are randomly changed from the current state by small amount and checked if there are any rulings intersecting. Then, only if there is no intersection, the torsion and the folding angles are calculated from the ruling directions. Finally, after some iterations, the parameters with the smallest distances are adopted. In this paper, we refer to this method as a *ruling-based optimization method*. Other than the intuitive manipulation, this new interface leads to many applications and extended features such as binding two or more models to shape a more complex combined model, as in Figure 1(b), and the point cloud fitting of the curved fold model.

In this paper, related work on the paper modeling and the interactive manipulation is introduced in section 2. Then the geometry and the procedure of our method are explained in section 3, followed by some applications introduced in section 4. In section 5, we discuss about the evaluation result of the ruling-based optimization method.

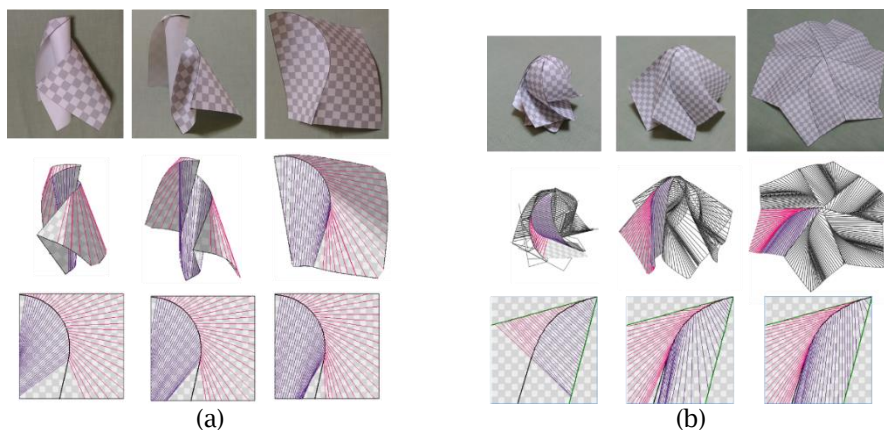


Figure 1: Curved fold models and the folding motions generated by our previous systems: (a) Curved folding with single crease [20], and (b) Rotational symmetric curved folding [21]. Top: photo of real paper. Middle: 3D model. Bottom: crease and rulings projected on the 2D space of flattened paper.

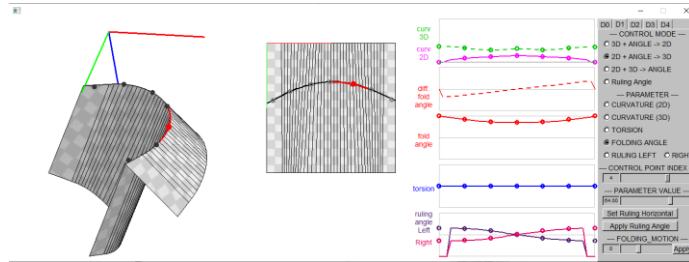


Figure 2: GUI system from our previous works, with four panes showing 3D model, 2D space of unfolded paper with a crease and rulings, curve charts of crease curve parameters, and the control panel to edit the parameters. The user edits the shape by changing the curve parameters and folding angles on the curve control points, shown as small dots on 2D and 3D crease curves.

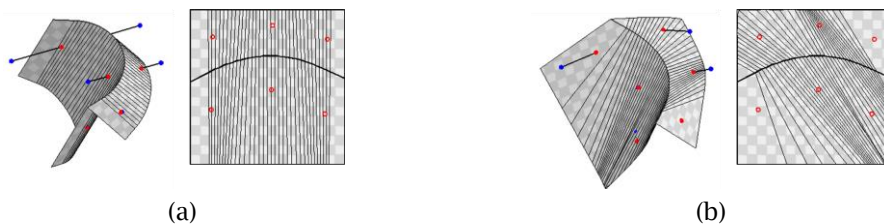


Figure 3: New user interface on 2D and 3D pane. Surface control points are shown as red dots. The target points are shown as blue dots connected to the corresponding surface control points in 3D pane. (a) Initial state. (b) Result of optimization.

2 RELATED WORK

In the area of modelling curved fold shapes, 3D surfaces have often been represented by a polygon mesh of fixed geometry composed of a group of quad strips and planar polygons. One early work of modeling a curved folded surface is the method proposed by Kilian et al. [4]. Their method reconstructs a developable 3D polygon mesh from input 3D data, such as point clouds, by initially fitting planar polygons on the 3D shape and then optimizing their shapes and orientations for its developability and the connectivity. While their method can generate complex shapes with many short creases branching and joining at their ends, it requires as an input a 3D shape that is nearly developable. In contrast, Mitani and Igarashi proposed a simple user interface to design multi-crease curved folding by folding the curved surface along planer curves [8]. As this tool uses the reflective principle of the rulings, only a simple shape is required as an input and the users can create new shapes freely with simple operations. Mitani has also introduced an interface to design 3D origami of rotational symmetric shapes with a simple user input [7]. Other than the discrete polygon mesh, Mundilova proposed an approach to mathematically describe the shape of a curved fold surface with one side of the crease forming a cone or a cylinder [9]. The generated shape is a smooth continuous surface with no approximation. These approaches and methods are successful in modelling developable surfaces as a static object but do not consider the folding motion.

In the field of paper folding simulation, Tachi developed software to simulate the folding motion of the paper using the rigid folding method [14],[15]. The method represents the paper by planar polygons of fixed geometry hinged at the edges, mainly targeting the paper folded on straight lines. He has also proposed some flat-foldable vault structures composed of flat-foldable tubes assembled by welding two sheets [16],[17]. The tubes are folded along the curve with the folding angles constant throughout the curve, which ensures the fixed rulings, enabling the surface to be modeled

as the rigidly folding object. To model more flexible motion with high degrees of freedom, Narain et al. proposed a method to simulate a sheet of thin materials by adaptive mesh refinement that could model irregular deformations such as crumpled paper [6]. Kilian et al. also take an approach of re-meshing the triangular mesh in accordance with the local surface curvatures in modeling the 3D shape of paper in string actuated folding motion [5]. Their methods successfully represent various shapes, but resulting mesh data may contain thousands of faces, not necessarily suitable for observing the rulings. Ghassaei et al. implemented a fast and interactive web application for origami simulation that works on a graphics processing unit (GPU) [3]. As a preprocess, the application triangulates the input crease pattern into fixed polygon model, allowing some stretching or shearing of the polygonal faces. The flexibility of the model enables the polygon shape to transit from the flat state to the folded state. To simulate the folding motion of curved folded papers using this application, Sasaki and Mitani proposed a method to triangulate the crease patterns along the approximated rulings of the curved folding [13]. Their work is successful in showing the folding motion freely with a software publicly accessible by anyone and with an easy user interface. But the generated shapes during the folding motion are not accurate nor developable. Taking a more refined approach, Tang et al. defined a curved surface by a pair of spline curves and solved the constraints for its global developability through iterative processes [18]. Rabinovich et al. introduced discrete orthogonal geodesic nets, which approximate a smooth developable surface by optimizing the corner angles of the quad mesh to be orthogonal [10]. In their subsequent work, the model was refined to represent a curved folding by having two smooth curved surfaces intersect on an identical curved crease [11]. In their latest work, their system allows the users to deform the shape interactively and intuitively, but taking different approach from ours in the modeling and the optimization method [12].

Among the researches of modelling and simulating the curved folded surface and its folding motion with some user interaction, our work allows intuitive user interface to manipulate curved folded surfaces modelled as quad strips based on the rulings. The system allows the ruling transition during the folding motion or other shape deformations, causing the geometry of the polygon to change. Our method of fitting the curved fold model to the target points is different from the previous researches in that it optimizes both the geometry and the pose of each polygon faces. Because the polygon shape is based on the geometry of the curved folding, the developability is guaranteed.

3 FITTING METHOD OF THE CURVED FOLDED SURFACE

Our method optimizes the torsions and the folding angles on the crease curve to minimize the displacement of the original shape to the target. The user can freely set the positions of surface control points on the 2D paper and their target points in 3D space, leaving the task of shape fitting to the optimization process. In free editing of the curved fold model, the surface control points should be set on the points that are to be moved, and the target points should be set in the directions of their movement, as if the user is choosing one point on surface and pulling it toward the target point. In an application where the shape of the model should meet some constraints, the control and the target point should be set according to the constraints. For example, in designing one part of complex shape composed of multiple parts, and some points of the surface must touch other parts with no gap, the joint part may be specified as the target point where the control points on the surface should meet. Depending on the user specified points, there may be some gap between the optimized shape and the target points. Nevertheless, due to the surface model based on the geometry of the curved folding introduced in subsection 3.1, the developability of the folded surface is guaranteed. The cost function used in the optimization process is the average distance between the pairs of the target points and their corresponding surface control points. The distance between the two points, instead of the distance between a point and the surface, is used to avoid the tangential drift. The tangential drift is a natural process in fitting a surface to some target points. But we also consider of applications where pairs of points must stick together without sliding along the surface, such as binding some points on a surface. To obtain the optimal torsions and the folding

angles, some popular optimization methods such as gradient descent would be inefficient because the parameter space has many local minima and invalid states. With seven curve control points on the crease curve, it optimizes 14 parameters, the torsions and the folding angles on each curve control point. Small changes in the parameters may cause large differences in the resulting shape and the ruling directions, easily falling into invalid states, such as ruling intersections, the ruling intersecting with the crease curve, or self-collisions. Instead of this direct optimization, we introduce a ruling-based optimization method which searches for the ruling directions that best fits the targets. In every optimization step, the method first tries some changes in the ruling directions and ensures the rulings to be nonintersecting. Then the torsions and a set of folding angles are calculated from the ruling angles, or the angles between the tangents of the curve and the ruling, as described in subsection 3.2. Finally, the rulings and the folding angles which generates the best fitting 3D model are adopted by the iteration process explained in subsection 3.3. The ruling-based optimization has 15 parameters; the ruling angles on the left and the right side of the seven curve control points, and the folding angle. The folding angle is 1-DoF as its derivative is calculated from the ruling angles. Given the ruling angle on one point of the curved crease, the folding angles on the rest of the points are calculated. Although having larger number of parameters, the optimization process is more stable as the output shape is less sensitive to the parameters. Also, because the rulings are changed by small amount sequentially from a valid state in the previous iteration, there are smaller probabilities of ruling intersections compared to setting new random parameters in every iteration. The geometry and the procedure are described in the following sections.

3.1 Geometry of Curved Folding

Our system uses the following equations introduced by Fuchs and Tabachnikov [2] to calculate the ruling angles from the 2D or 3D curvature, torsion, and the folding angle,

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

$$\cot \beta_L(s) = \frac{\alpha(s)' - \tau(s)}{k_{2D}(s) \tan \alpha(s)'} \quad (3.2)$$

$$\cot \beta_R(s) = \frac{-\alpha(s)' - \tau(s)}{k_{2D}(s) \tan \alpha(s)'} \quad (3.3)$$

where $k_{2D}(s)$ is the curvature of 2D crease curve, $k(s)$ and $\tau(s)$ are the curvature and the torsion of the 3D crease curve made by folding along the 2D crease, $\alpha(s)$ is the folding angle, and $\beta_L(s)$ and $\beta_R(s)$ are the ruling angles on the left and the right side of the crease curve, as illustrated in Figure 4. All parameters are represented as arc-length parametric function.

In our previous works, the user first chooses a mode to specify which parameters to be controlled or calculated, and then change the value of the parameters to deform the shape of 2D/3D crease curve and the generated 3D polygon shape [19],[20]. In designing the folding motion or deforming the 3D paper shape folded along a fixed 2D crease curve, the values of the torsion and the folding angle are the parameters to be changed by the user, while the 2D curvature is the given parameter [21]. The 3D shape of the curved folded surface is calculated by the following procedure:

1. Calculate the curvature $k(s)$ of the 3D crease curve from the 2D curvature $k_{2D}(s)$ and the folding angle $\alpha(s)$ using the Equation (3.1).
2. Build a 3D crease curve from the torsion $\tau(s)$ and the 3D curvature $k(s)$. Calculate tangential, normal, and binormal vectors; $\mathbf{T}(s)$, $\mathbf{N}(s)$, and $\mathbf{B}(s)$ on the 3D curve.
3. Calculate the rulings $\mathbf{r}_L(s)$, $\mathbf{r}_R(s)$ in the 3D space using the folding angle $\alpha(s)$ and the ruling angles $\beta_L(s)$, $\beta_R(s)$ based on the tangential, the normal, and the binormal vectors as follows,

$$\mathbf{r}_L(s) = \cos \beta_L(s) \mathbf{T}(s) + \sin \beta_L(s) \cos \alpha(s) \mathbf{N}(s) - \sin \beta_L(s) \sin \alpha(s) \mathbf{B}(s), \quad (3.4)$$

$$\mathbf{r}_R(s) = \cos \beta_R(s) \mathbf{T}(s) - \sin \beta_R(s) \cos \alpha(s) \mathbf{N}(s) - \sin \beta_R(s) \sin \alpha(s) \mathbf{B}(s). \quad (3.5)$$

4. The 3D shape of the curved fold model is then obtained as the quad strips connecting the rulings.

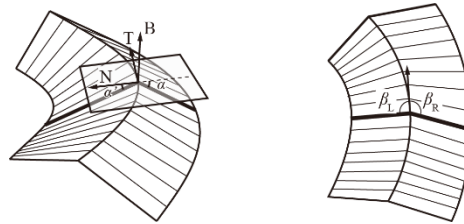


Figure 4: Parameters of a crease curve. Left: curved crease and rulings on the 3D model, with tangential, normal, and binormal vectors of 3D crease curve. Right: crease curve and rulings on the model flattened on to 2D plane.

3.2 Creating a Curved Fold Shape from the Rulings

In our new optimization method, ruling angles are given and the rest of the parameters are calculated. To derive the torsion $\tau(s)$ and the folding angle $\alpha(s)$ from the ruling angles $\beta_L(s)$, $\beta_R(s)$, Equation (3.2) and (3.3) are transformed as,

$$\tau(s)/\tan\alpha(s) = \frac{1}{2} k_{2D}(s)(\cot\beta_L(s) + \cot\beta_R(s)), \quad (3.6)$$

$$\alpha(s)'/\tan\alpha(s) = \frac{1}{2} k_{2D}(s)(\cot\beta_L(s) - \cot\beta_R(s)). \quad (3.7)$$

By expressing the parameters in the discrete form and setting a base folding angle α_0 on the starting point of the crease curve, the torsions $\{\tau_i\}$ and the folding angles $\{\alpha_i\}$ on the rest of the points are calculated sequentially by

$$\alpha_i = \alpha_{i-1} + d\alpha_{i-1}dx, \quad (3.8)$$

$$\tau_i = \frac{1}{2} k_{2Di}(\cot\beta_{Li} + \cot\beta_{Ri}) \tan\alpha_i, \quad (3.9)$$

$$d\alpha_i = \frac{1}{2} k_{2Di}(\cot\beta_{Li} - \cot\beta_{Ri}) \tan\alpha_i. \quad (3.10)$$

In our implementation, the ruling angles on the curve control points, shown as the dots on the curve in Figure 2, are randomly changed in the optimization process. The rulings on the rest of the curve are smoothly interpolated by the spline curve of degree-three. This follows the interpolation method of the curvatures, the torsion, and the folding angle in our previous works [19],[20],[21]. Because the spline interpolation of degree-three generates a curve of G2 continuity, the parameters are second-order differentiable, enabling smooth changes in the ruling directions along the curve. In calculating the torsion and the folding angle sequentially using Equations (3.8) to (3.10), we used the folding angle α_m on the center of the crease curve as the base folding angle, instead of the starting point. Then the sequential calculation is processed from the center toward the both ends. Setting the base folding angle at the center, the parameter value is more intuitive to the user, as in some shapes the folding angle at the end of the crease curve tends to become extreme such as 90 degrees. After the torsion and the folding angles are obtained for all points of the crease curve, the parameters, with the 2D curvatures, are used to derive the 3D shapes of the curved fold model, as described in subsection 3.1. The rulings are also recalculated from the parameters.

3.3 Optimization Process

In the ruling-based optimization process, firstly, a curved fold model is given as an initial state, created by the method in the previous subsections. The surface control points and their corresponding target points are given either by input file or from the user interactively through the GUI system. Here, the surface control points are the 2D positions on the flattened sheet of paper, whose 3D positions are obtained by projecting them on the surface of the 3D curved fold model. The target points are given as the 3D positions. With the 2D crease curve fixed, the parameters to define the 3D shape of the curved fold model are optimized by the following process:

1. Change the ruling angles at the curve control points by small amounts, and interpolate them with spline curves to obtain the ruling angles on the rest of the curve. The small changes in the

ruling angles are chosen from the table of displacements designed empirically, listed in Table 1. The table of displacements includes the uniform increase/decrease (Table 1 – row 1, 2), the gradational increase/decrease along the curve starting from zero on one end (Table 1 – row 3-6), and the increase/decrease on one control point (Table 1 – row 7-20). One of the rows in the table is randomly chosen and applied.

2. Check if there are any ruling intersections in the 2D space. If there are, go back to step 1.
3. Calculate the torsion and the folding angles sequentially using Equations (3.8) to (3.10). As there are more than one solution to the folding angle depending on the base folding angle α_m , calculate all possible solutions by changing the base folding angle by 2 degrees in the range $[-\pi/2, \pi/2]$, and confirm that the folding angles on all parts of the curve range in $[-\pi/2, \pi/2]$. If out of the range, discard the sample.
4. Recalculate the 3D crease curve and the rulings from the newly obtained torsion and the folding angles. Check if there are ruling intersections, as the recalculated rulings may have some small intersections due to the numerical error. If there are, discard the sample. Self-collisions and rulings intersecting with the crease curve are also checked at this step.
5. Calculate the cost function, which is the average distance between the target points and the corresponding surface control points. The base folding angles with the smallest cost are adopted.
6. Check if the cost of the adopted base folding angle is smaller than the previous iteration, derived from the previous ruling angles. If it is, update the ruling angles. Otherwise, go back to step 1.
7. Check if the cost is smaller than the predefined threshold. If it is, terminate the process. If not, go back to step 1.

The iteration is repeated until the cost is below the threshold or the number of iterations exceeds the predefined maximum number. In our implementation the threshold was set to be 0.3mm and the maximum number of iterations to be 500.

Index	control point #1	control point #2	control point #3	control point #4	control point #5	control point #6	control point #7
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1
3	0	0.0286	0.0571	0.0857	0.114	0.143	0.171
4	0	-0.0286	-0.0571	-0.0857	-0.114	-0.143	-0.171
5	0.2	0.171	0.143	0.114	0.0857	0.0571	0.0286
6	-0.2	-0.171	-0.143	-0.114	-0.0857	-0.0571	-0.0286
7	0.1	0	0	0	0	0	0
8	0	0.1	0	0	0	0	0
9	0	0	0.1	0	0	0	0
10	0	0	0	0.1	0	0	0
11	0	0	0	0	0.1	0	0
12	0	0	0	0	0	0.1	0
13	0	0	0	0	0	0	0.1
14	-0.1	0	0	0	0	0	0
15	0	-0.1	0	0	0	0	0
16	0	0	-0.1	0	0	0	0
17	0	0	0	-0.1	0	0	0
18	0	0	0	0	-0.1	0	0
19	0	0	0	0	0	-0.1	0
20	0	0	0	0	0	0	-0.1

Table 1: List of possible changes to be applied to the ruling angle. Units are in radian.

4 APPLICATIONS

Our new method enables applications such as (1) an intuitive user interface to manipulate a curved fold model and (2) combining curved fold models by stitching the boundaries of the two surface patches. The new interface also leads to many applications and extended features such as point cloud fitting and combining models by overlapping two curved surfaces.

4.1 Intuitive User Interface for a Single Crease Model Manipulation

We have implemented the new optimization process on our GUI system (Figure 2) for editing a curved folding with a single crease. An example input and its result are shown in Figure 3. The coordinates of the surface control points and the target points are given from an input file or by the mouse operation. The input file contains a list of 2D coordinates of the surface control points and the 3D positions of the corresponding target points. To activate the mouse input, the user first switches the editing mode on the control panel. Then, by left-clicking, dragging, or right-clicking in the 2D pane, a surface control point is added, moved, or deleted, specifying its position on the 2D surface. The corresponding target point in the 3D space is initially set on the surface at the user specified 2d position. Then, in the 3D pane, the 3D target points are dragged in the 3D space. In detail, the user drags the target point along the image plane on which the objects are rendered. To move the target point in other directions including a vector component orthogonal to the image plane, the user changes the viewpoint and then drag the target points along the new image plane. In our system, the viewpoint can be selected from x-up, y-up, or z-up coordinate system, or it can orbit around the origin by the mouse drag in another editing mode. By changing the viewpoint, the whole curved fold model with the surface control points and the target points is rotated. This part of the user interface to move the target points in 3D space is not the main content of our research, and other common methods may provide better usability, such as dragging along the normal vector or the local 3D coordinate system on the plane. After adjusting the control points and the target points to the user intended positions, the optimization is performed through the control panel operation. The user can check the result and, if necessary, re-adjust the positions of the target points for another optimization process until having an intended shape.

4.2 Stitch the Boundaries of Two Surface Patches

The feature is also implemented on the GUI system to design and visualize the rotational symmetric curved folding [21]. The ruling-based optimization is used to stitch the adjacent surface patches. Figure 5 shows a result of the stitching in one of the frames of the folding motion of the origami model shown in Figure 1(b). As the initial state, the six patches, each having one curved crease, are placed in rotational symmetry, rotated 60 degrees from the adjacent patch (Figure 5(a)). To minimize the gap between the patches, the poses of the patches are first optimized without changing the 3D shape of each patch, keeping the rotational symmetry (Figure 5(b)). Then the surface control points are placed evenly on the boundaries of the patches. The target points are set to be the midpoints of the pairs of surface control points on the adjacent patch boundaries. Finally, the torsion and the folding angle are optimized so that the two corresponding boundaries forms the same shape, approximating the series of the midpoints (Figure 5(c)). This process had been performed manually in the previous work, but the new feature enabled it to be semi-automated.

4.3 Other Applications and Features

Although not implemented, there are some other applications and extended features which may be realized by this optimization process. One possible feature is the fitting of the two overlapping curved surfaces. Stitching the boundaries is one way to combine two surface patches sparsely. By pasting or sticking together two curved surfaces, the two patches are combined continuously with the surface overlap. For example, by overlapping two curved surfaces on the left and the right sides of different surface patches, we get a surface model with two curved creases joining three smooth

developable surfaces, with the center surface composed of two overlapping surfaces. By adding more patches, a user could create more complex curved folded shapes.

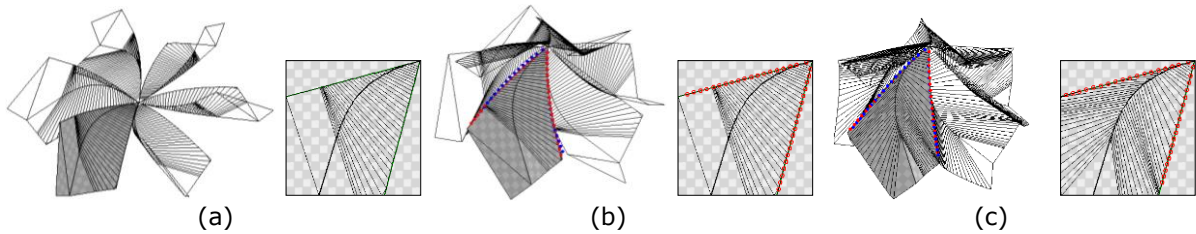


Figure 5: Stitching the boundaries of surface patches: (a) Initial state of the six patches, (b) After alignment, (c) After the optimization of torsion and folding angle. Left: the 3D model. Right: 2D crease and rulings of one patch.

Another feature is a point cloud fitting of the curved fold model. It may be useful in analyzing the real motion of a curved creased paper being folded with the support of 3D sensing. Using a 3D camera such as Kinect, RealSense, etc., partial point clouds of a sheet of paper are easily captured. With an RGB camera calibrated and synchronized with the 3D camera, color images are obtained and the point cloud can be colorized. By capturing a curved folded paper with textures such as markers and coded patterns, the 3D positions of the target points are detected from the color images and the point clouds. Then the curved fold model with the surface control points located identically to the markers or coded patterns are deformed to fit the target points, approximating the point clouds representing the partial 3D paper shape.

5 EVALUATION

We evaluated our method using synthetic data of the curved fold model with one crease, shown in Figure 2. The model is a 200mm square sheet with a fixed 2D curved crease. Various 3D shapes are created by changing the folding angle and the torsion of the crease. To test our method, a maximum of 81 surface control points are placed on the 2D surface in 9×9 lattice alignment. The target points are given as the 3D positions of the surface control points of the target shape, also created by changing the folding angle and the torsion of the same curved fold model. In the experiment, the model shape is optimized to fit a set of target points under different conditions: (a) initial shapes, (b) target shapes given as the positions of the target points, (c) the number and the alignment of the target points, and (d) the parameter optimization policy, or the method of changing parameter values in each iteration. We set the maximum number of iterations to be 500 and the threshold of the average gap to terminate the iteration to be 0.3 mm. The method is implemented in C++ using Visual Studio 2019, with the parallel processing enabled by OpenMP, and is processed on Windows 10 with Core i7-1165G7 @ 2.80GHz and 16 GB RAM. The results are evaluated by (A) the average gaps or distances between the target points and the surface control points on the optimized curved fold model and (B) the number of iterations. The results are also compared with the results of an optimization method based on the torsion and folding angle. The details of the conditions and their results are explained in the following subsections.

5.1 The Initial and the Target Shapes

We created 15 shapes of the curved fold model for the evaluation: (i) 5 shapes with the predefined rulings, (ii) 5 shapes with the random rulings, and (iii) 5 shapes with random torsions and folding angles, shown in Figure 6. The three groups of shapes are made by the following procedures:

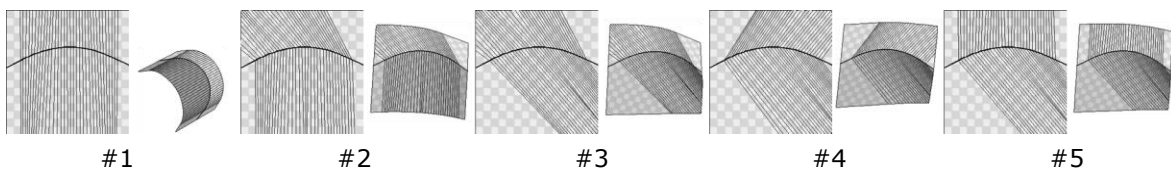
- i. The shapes with the predefined rulings are created by first manually adjusting the rulings to be parallel and orthogonal or about 45 degrees to the paper edge. Then the base folding angle at the center of the crease are set to be 30 degrees. The torsion and the folding angles on the

rest of the crease curve are calculated sequentially by the method described in section 3. The whole 3D shape is obtained according to the curve parameters.

- ii. The shapes with the random rulings are created by setting random ruling angles on the curve control points, interpolating them by spline interpolation, and checking the ruling intersections to decide whether to accept or to reject the shape. If accepted, then a random value is set as the base folding angle, and the rest is calculated sequentially. Confirming that the folding angles on all parts of the crease curve are in the range $[-\pi/2, \pi/2]$, the shape is adopted. Otherwise, another random folding angle is tested. The range of random parameter values are $[0, \pi]$ for the ruling angles and $[-\pi/2, \pi/2]$ for the folding angle.
- iii. The shapes with random torsion and folding angles are created by setting random torsions and the folding angles on the curve control points, interpolating them by the spline interpolation, and the shape calculated by the procedure in subsection 3.1. Then the shape is validated by checking the ruling intersections and the self-collisions. The range of random parameter values are $[-0.05, 0.05]$ for the torsion and $[0, \pi/2]$ for the folding angle.

By observing Figure 6 #6-15, the rulings from the random parameters tend to expand toward the paper edges, in another words, have wider intervals between the rulings at the area far from the crease curve. This is a natural result as the end points of the rulings are fixed on the crease curve while the other end points seek to stay apart from each other to avoid intersecting. Furthermore, the shapes with random torsion and folding angles (#11-15) often produces "extreme shapes" with partially large folding angle or torsion, and with a resulting ruling that is very acute at the ends of the curved crease, making the ruling expansion even larger. This is also empirically natural because when adjusting the torsion and the folding angles manually to resolve the ruling intersections, it is an easy solution to make such shapes with expanding rulings.

The overall results of the optimization are shown in Tables 2-6. The 15 shapes are used as both the initial shapes and the target shapes. Each result in the tables is the average result of the 10 different patterns of the target points (Figure 8) discussed in subsection 5.2. From the results, we could infer that our method is efficient in forming ruling-based shapes (#1-10) from another ruling-based shapes. Also, the number of iterations shows that with the initial shapes with the parallel rulings (#1-5) the process reached the goal faster. This result is fairly intuitive because, in the manual manipulation it is easier to change the rulings from parallel to randomly expanding and requires more effort to do the reverse manipulation. Compared to the above results, the process using the initial shapes with random torsion and folding angles (#11-15) results in larger average gaps, requiring more iterations to reach the goal. At last, for the target shapes with random torsion and folding angles, our method was unable to reach the threshold of the average gap, 0.3mm, within the maximum number of iterations, 500. Some result shapes of this case are shown in Figure 7. Comparing with the shapes in Figure 6 #11-15, we observed that the overall shapes are similar to the target shapes but the ruling angles on the edges of the crease curve are not as acute as the targets. It should be noted that the target shape #6 was reached by only a few iterations from all initial shapes. This is because the shape #6 is nearly flat so that any rulings are accepted by making the folding angles to be close to zero. As to the shape #12, no iteration was processed as the initial shape is somewhat extreme that any ruling displacement resulted in invalid shape. At last, the process time is summarized in Table 6. The average process time is 5.54 seconds, increasing up to 22 seconds in some difficult cases. They are closely related to the number of iterations, shown in Table 5, that we could estimate the process time from the number of iterations.



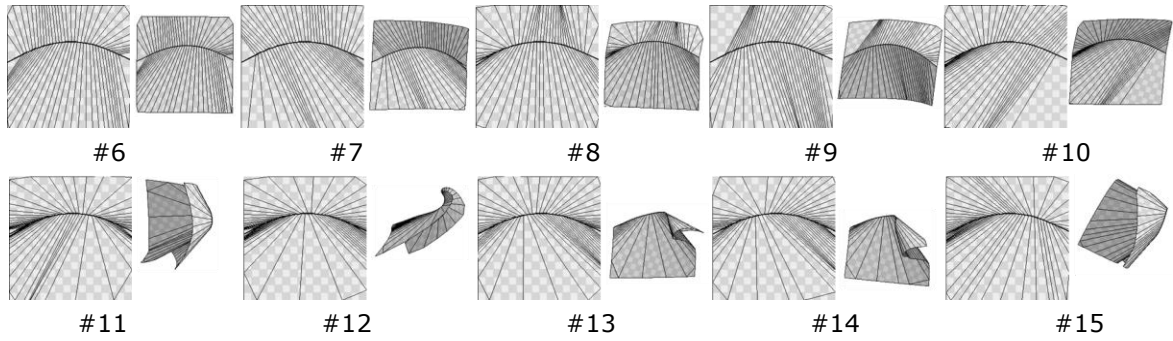


Figure 6: 3D shapes used for evaluation. Top row (#1-5): shapes with the predefined rulings, middle row (#6-10): shapes with the random rulings, bottom row (#11-15): shapes with random torsion and folding angles. Left: 2D curved crease and rulings, right: 3D shape.

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		0.27	0.29	0.25	0.23	0.07	0.26	0.30	0.29	0.27	3.62	2.37	1.96	3.16	2.57
	#2	0.28		0.37	0.34	0.78	0.08	0.24	0.30	0.28	0.27	3.08	2.05	2.05	3.15	2.12
	#3	0.28	0.26		0.33	0.25	0.13	0.31	0.30	0.30	0.55	3.98	2.51	2.06	3.57	2.69
	#4	0.27	0.25	0.32		0.26	0.16	0.28	0.29	0.29	0.59	8.09	2.26	2.16	3.74	2.95
	#5	0.25	0.25	0.28	0.28		0.14	0.25	0.31	0.30	0.61	4.62	2.34	2.13	3.45	3.20
	#6	0.28	0.30	0.40	0.42	0.28		0.27	0.29	0.29	0.27	2.40	1.68	1.53	2.80	1.27
	#7	0.28	0.30	0.29	0.29	0.25	0.12		0.28	0.30	0.33	3.51	1.71	1.20	2.39	2.18
	#8	0.30	0.29	0.32	0.32	0.30	0.06	0.30		0.33	0.33	4.39	2.35	1.56	3.97	2.39
	#9	0.28	0.29	0.32	0.28	0.28	0.10	0.27	0.30		0.33	2.87	2.17	1.25	3.47	2.26
	#10	0.30	0.28	0.58	1.05	1.11	0.11	0.34	0.29	0.41		2.36	1.86	1.96	3.64	1.83
	#11	0.35	0.53	0.73	0.75	0.47	0.08	0.34	0.31	0.68	0.29		1.44	1.61	6.75	1.11
	#12	7.73	7.83	7.31	8.18	7.19	0.44	6.43	5.77	9.13	3.67	24.20		17.30	22.03	19.15
	#13	0.44	0.45	0.50	0.49	0.39	0.11	0.30	0.37	0.56	0.47	4.02	1.20		1.52	3.47
	#14	0.43	0.56	0.57	0.51	0.42	0.12	0.30	0.32	0.55	0.34	4.75	1.39	0.59		1.93
	#15	0.30	0.34	0.58	1.21	0.95	0.11	0.36	0.37	0.49	0.27	1.51	1.77	1.61	4.57	

Table 2: Average gap between the target points and the surface control points. Units in mm.

5.2 The Number and the Alignment of the Surface Control Points

We tested our method using 10 different patterns of surface control points, shown in Figure 8. The 10 patterns include the 81 points aligned as the lattice of 9x9 (Figure 8-(a)), its subsets with control point on only one side of the crease (Figure 8-(c),(d)), and the subsets with 40, 10, and 5 control points, placed evenly or randomly (Figure 8-(b),(e)-(j)). We used the same data for the initial and for the target shapes as in the previous subsection, except for the initial shape of #12 which failed to be processed as described in subsection 5.1. The results, shown in Figure 9, are evaluated by (i) the average gap between the target and the control points used for the optimization, (ii) the average gap of all 81 pairs of points, shown in Figure 8-(a), used or not used for the optimization, and (iii) the number of iterations. (ii) The average gap of 81 points is introduced to approximate the gap between the whole surfaces of the target and the optimized shape, in contrast to the gaps on the control points only.

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		0.35	0.46	0.33	0.31	0.10	0.36	0.35	0.41	0.31	5.03	3.86	3.73	4.74	4.44
	#2	0.33		1.15	0.92	5.43	0.12	0.33	0.34	0.34	0.31	4.75	3.50	3.53	5.65	3.85
	#3	0.34	0.33		0.42	0.32	0.17	0.37	0.34	0.35	2.38	9.90	3.71	3.74	7.42	4.60
	#4	0.32	0.29	0.43		0.36	0.25	0.33	0.32	0.34	2.91	27.21	3.53	4.51	6.23	4.67
	#5	0.32	0.36	0.40	0.40		0.19	0.31	0.35	0.34	2.92	10.13	3.73	4.85	6.07	5.27
	#6	0.34	0.54	0.67	1.44	0.33		0.40	0.32	0.38	0.30	4.81	3.00	2.73	5.48	1.85
	#7	0.32	0.39	0.47	0.38	0.31	0.16		0.33	0.32	0.81	10.03	3.08	2.00	4.77	4.28
	#8	0.35	0.34	0.42	0.40	0.35	0.09	0.41		0.48	0.80	6.74	4.28	2.58	8.21	4.51
	#9	0.33	0.32	0.47	0.32	0.33	0.14	0.30	0.32		0.81	4.85	3.54	2.29	8.33	4.33
	#10	0.53	0.34	2.23	6.01	5.38	0.15	0.94	0.34	1.37		3.37	3.02	3.55	5.61	3.60
	#11	0.56	1.00	1.38	1.41	1.23	0.14	0.90	0.42	1.66	0.30		3.23	2.43	12.89	2.11
	#12	11.98	12.14	11.39	11.97	10.86	0.69	9.88	9.01	13.94	5.57	30.47		25.29	31.24	25.43
	#13	1.15	0.66	0.96	1.14	0.70	0.15	0.37	0.94	1.75	0.99	10.36	1.99		2.36	13.69
	#14	0.79	1.08	1.55	1.02	0.74	0.16	0.33	0.44	1.34	0.79	11.29	2.23	1.03		4.31
	#15	0.36	0.55	1.08	6.09	5.51	0.17	0.90	1.04	1.44	0.32	2.19	3.15	2.72	13.20	

Table 3: Maximum of average gap between the target points and the surface control points.

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		47	105	70	48	1	66	136	87	60	500	500	500	500	500
	#2	31		117	94	118	1	36	68	127	73	500	500	500	500	500
	#3	71	76		91	30	2	33	110	163	195	500	500	500	500	500
	#4	47	92	149		37	1	56	145	145	210	500	500	500	500	500
	#5	81	73	71	37		1	32	106	155	193	500	500	500	500	500
	#6	134	259	383	185	247		120	25	137	47	500	500	486	500	466
	#7	126	98	86	173	84	1		72	191	199	500	500	500	500	500
	#8	188	175	286	336	223	1	128		251	153	500	500	500	500	500
	#9	60	151	272	85	106	2	58	83		189	500	500	500	500	500
	#10	184	175	313	398	298	1	155	58	258		500	500	500	500	500
	#11	395	408	470	454	449	1	312	242	465	204		500	473	500	467
	#12	0	0	0	0	0	0	0	0	0	0	0		0	0	0
	#13	427	392	408	408	371	1	223	250	416	339	500	500		500	500
	#14	406	451	382	434	364	1	210	275	381	349	500	500	453		500
	#15	290	319	440	442	426	3	272	192	361	136	500	500	500	500	

Table 4: Average number of iterations.

This index may not necessarily be important depending on the applications. But it is useful in observing the effects of the control point alignments, that is, to check how well the optimized shape approximates the target shape using the set of surface control points. The result shows that (i) the average gap of the control points and (iii) the number of iterations is relatively large for the alignments (a), (e), (f), and (h) where the control points are distributed in all area of the 2D plane, and small for (b), (c), (d), and (j) where they are placed partially. On the other hand, (ii) the

average gap of 81 points was smaller for the uniformly distributed control points and larger for the partial distributions.

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		0.68	1.38	1.24	0.80	0.03	1.14	2.67	1.55	0.95	10.98	8.84	8.87	11.65	10.73
	#2	0.64		1.66	1.57	2.08	0.02	0.58	1.38	2.52	1.21	13.33	9.87	10.16	11.74	10.99
	#3	1.29	1.06		1.34	0.42	0.04	0.49	1.97	2.91	2.55	12.64	9.61	10.04	11.59	10.52
	#4	0.98	1.93	2.23		0.70	0.03	0.96	3.06	3.05	2.76	10.86	9.13	9.63	11.14	9.88
	#5	1.56	1.13	0.86	0.60		0.02	0.46	2.03	2.99	2.36	11.08	9.09	9.55	10.71	9.74
	#6	3.06	4.54	4.78	3.19	3.48		2.14	0.50	2.58	0.62	13.54	9.97	9.90	11.30	10.93
	#7	3.04	1.83	1.29	3.55	1.07	0.02		1.17	3.41	2.86	12.38	9.64	10.07	11.77	10.81
	#8	3.17	2.76	3.21	4.94	2.67	0.02	1.44		4.23	1.61	11.29	9.59	9.72	10.76	10.49
	#9	1.20	2.81	3.82	1.53	1.80	0.03	1.07	1.49		3.06	13.86	9.57	10.50	11.32	10.28
	#10	3.97	3.14	4.68	7.05	4.20	0.02	2.33	0.99	4.88		13.47	10.25	9.93	11.17	11.34
	#11	8.77	7.64	7.73	9.88	8.38	0.04	5.62	6.00	9.82	3.28		11.10	10.48	11.47	12.39
	#12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00
	#13	9.42	8.06	6.87	9.46	8.25	0.04	4.30	5.59	9.38	6.96	14.06	11.72		14.17	12.17
	#14	9.41	10.32	6.22	10.56	7.56	0.03	3.72	6.75	8.96	7.20	13.42	11.95	11.32		12.51
	#15	5.87	5.55	6.56	7.10	7.74	0.07	4.54	3.45	6.66	2.44	14.58	10.32	10.51	11.83	

Table 5: Average process time. Units in seconds.

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		2.02	3.16	2.21	1.36	0.04	4.94	10.92	6.06	1.76	15.17	11.93	11.66	16.74	14.33
	#2	1.61		7.16	3.41	9.89	0.02	1.42	3.10	6.92	1.82	19.50	13.15	13.78	15.38	14.57
	#3	2.48	2.38		2.60	0.81	0.09	1.99	5.15	8.05	6.38	16.55	13.20	13.19	14.65	14.19
	#4	2.12	3.71	6.86		1.64	0.06	1.70	9.82	10.85	6.29	14.43	12.36	11.91	15.68	13.22
	#5	7.44	1.92	2.66	1.18		0.04	1.49	5.01	9.92	4.02	16.17	11.91	13.12	13.33	14.07
	#6	13.08	10.40	8.07	7.65	5.56		6.01	1.71	8.82	1.31	17.60	12.50	13.38	15.35	15.00
	#7	8.83	3.55	7.60	11.01	4.12	0.02		2.03	9.41	6.34	16.57	12.77	13.70	14.51	13.84
	#8	6.94	6.52	6.20	9.07	5.10	0.03	2.36		10.64	3.98	15.81	13.14	13.19	13.05	13.78
	#9	3.71	6.64	8.01	3.60	3.84	0.09	2.47	2.36		7.33	22.26	12.35	14.59	13.96	14.83
	#10	14.10	9.25	10.61	12.59	7.30	0.03	5.88	2.68	13.82		20.25	14.42	12.90	15.08	16.13
	#11	15.29	14.68	11.00	15.55	12.47	0.09	10.07	15.00	15.73	6.93		14.39	16.32	14.48	16.68
	#12	0.01	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00		0.01	0.00	0.00
	#13	17.89	12.59	14.94	14.95	14.32	0.05	9.17	13.35	14.76	12.57	19.54	15.09		18.09	15.87
	#14	15.54	15.81	11.61	16.37	13.99	0.05	8.27	14.62	14.63	13.04	19.41	15.04	15.87		17.03
	#15	11.32	11.91	10.23	11.12	12.06	0.13	10.93	8.57	12.70	5.62	19.41	13.68	13.21	16.16	

Table 6: Maximum process time. Units in seconds.

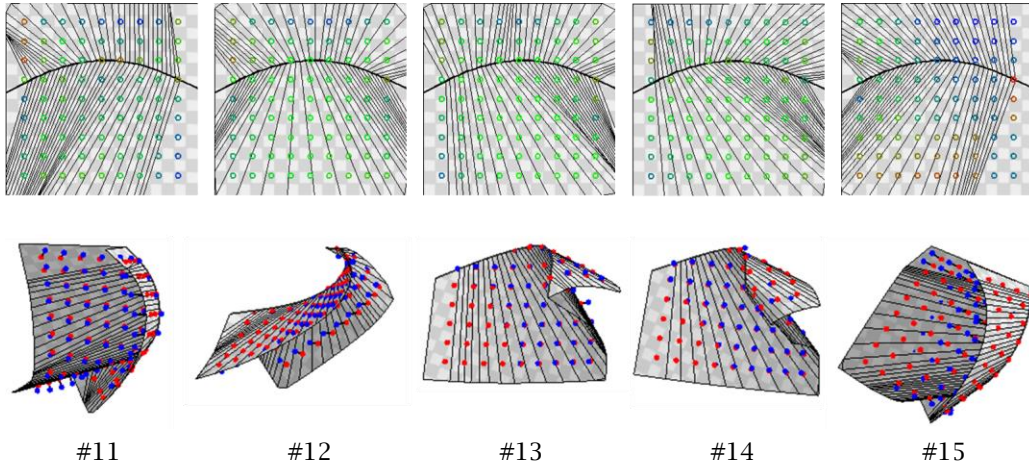


Figure 7: Results of optimization with initial shape #1 and target shapes #11-15. Top row: 2D crease curve, rulings, and the control points shown in colors where green indicating zero distance to the target points, red and blue showing positive and negative displacement from the surface. Bottom row: 3D shapes with control points shown in red and target points shown in blue.

The number of control points had less impact compared to the distribution. From above observation, the control points should be placed uniformly if the whole shape is to be controlled. They should be placed partially with no redundant points if it only needs to meet the partial constraints, as in the case of matching the boundaries of the surfaces with no care about the shape of non-boundary area.



Figure 8: Target point distributions. '1' indicates where the surface control point is placed and '0' is where it is absent. (a) Full points. (b) Left half only. (c) Convex surface only. (d) Concave surface only. (e) 40 points evenly placed. (f) 10 points evenly placed. (g) 5 points evenly placed. (h) 40 points randomly placed. (i) 20 points randomly placed. (j) 5 points randomly placed.

5.3 Parameter Optimization Policy

In changing the ruling angles during the optimization process, we used the table of displacements, shown in Table 1, for efficiency. To validate the effect, we tested the optimization process which does not use the table, for comparison. Instead, each of the 14 ruling angles are changed randomly and independently. The range of random values are [-5, 5] in the unit of degrees, close to 0.1 radian which appears many times in Table 1. Figure 10 shows the frequency distributions of (a) the average gap, (b) the number of iterations, (c) the average number of iterations in obtaining the valid rulings (step 1 and 2 in subsection 3.3), and (d) the process time, with and without the displacement table. The number of trials for each distribution is 2100, including the 15 initial shapes, the 14 target shapes, and the 10 alignments of the surface control points. The widths of the classes are 0.1 mm, 20 times, 0.25 times, and 0.5 seconds for (a), (b), (c), and (d).

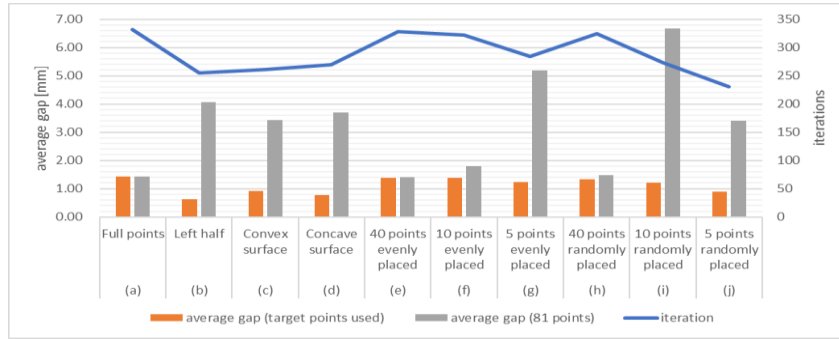


Figure 9: The results of different target point distributions. (i) The average gaps between the target and the control points used for the optimization are shown in orange bars. (ii) The average gaps of all 81 pairs of points are shown in gray bars.

Although the number of iterations in obtaining the valid rulings are smaller using the table (Figure 10(c)), the average gap and the number of iterations show similar distributions for the processes with and without the table (Figure 10(a), (d)). From this result, we confirmed that the displacement table are effective in searching valid rulings. But its effect on the final result is small and did not contribute to the overall improvement. We analyze that this is because the process time of searching the rulings is small relative to other parts of the process. It implies that the ruling-based optimization is reasonable in that it searches for the valid rulings thoroughly in small process time, leaving only one more parameter to be optimized, the base folding angle.

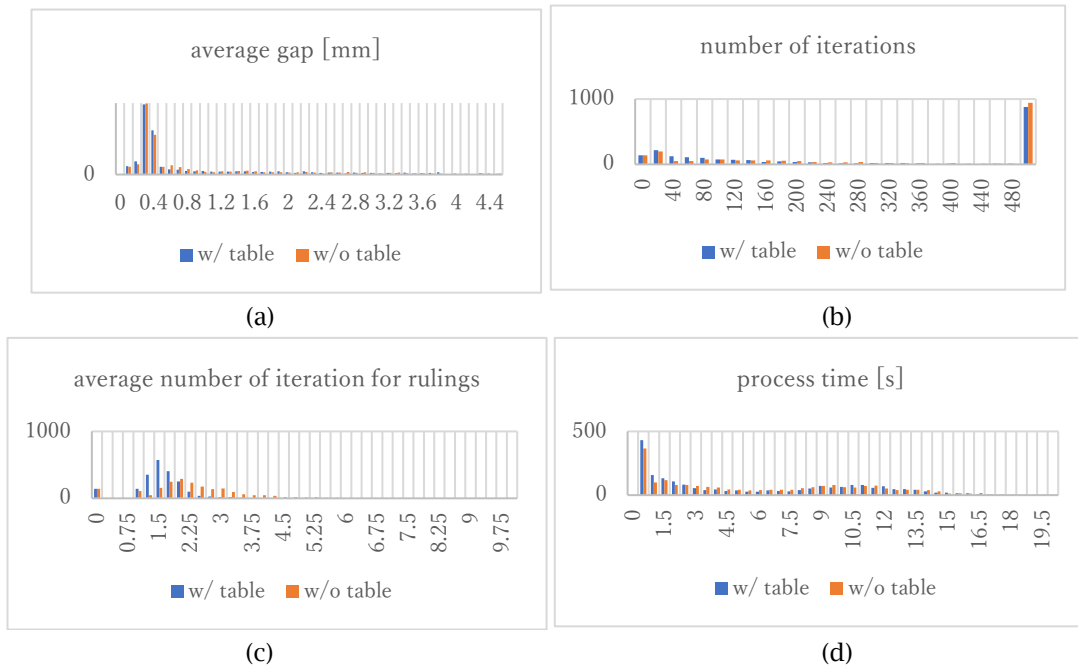


Figure 10: Frequency distributions of four indices: (a) average gap, (b) number of iterations, (c) average number of iterations in obtaining valid rulings, (d) process time, with and without displacement table. Horizontal axis shows the value of the indices. Vertical axis shows the

frequency. The range of horizontal axis is set to show the difference of two distributions, including more than 90 percent of the data. The high peak in (b) at 500 shows the number of trials which reached the maximum number of iterations.

5.4 Comparison with Optimization Method based on Torsions and Folding Angles

For comparison, we have also tested a naive implementation to directly search for better torsions and the folding angles on the curve control points. In every iteration, the 14 parameters, which are the torsion and the folding angles on the seven curve control points, are changed randomly by small amount and evaluated using the cost function. The parameters are updated if the result shape does not contain a ruling intersection and the cost is decreased. We used this naive method while there are many efficient computational optimization methods which converges faster with a larger amount of parameter shift per iteration using the gradients. This is because we assumed that the parameter space of the torsions and the folding angle has many local minima and invalid states that, with a large shift of parameters, the model is difficult to keep valid state. The comparison with other optimization methods is left to be the future work. For the evaluation, the same initial and the target shapes as subsection 5.1 are used with the 10 patterns of surface control points alignments described in subsection 5.2. The threshold of the average gap to terminate the iteration was set to be 0.3 mm, same as the evaluation of our ruling-based optimization. We set the maximum number of iterations to be 10000, 20 times larger than the ruling-based optimization. This is because the ruling-based optimization tries multiple folding angles in one iteration. For fair comparison, we set the maximum iteration large enough for the parameter values to converge. Although this search method often converges in local minima, the re-initialization of the parameters is not introduced because it is not effective due to the sparse localization of valid state. That is, randomly sampled parameters have very small probability of generating valid shapes, easily causing ruling intersections. The result is shown in Table 7. As a whole, the ruling-based method shown in Table 2 shows better, or smaller average gaps, than this result.

One reason for conducting this experiment was to check if this method is effective for the target shapes with random torsion and folding angles (Figure 6 #11-15). In subsection 5.1, the results were not as good as other conditions for this group of data. We analyzed that this may be due to the inconsistency of the shape generation method, that the parameters to be optimized are different from the random parameters to create the target shapes, and assumed that the direct optimization of the torsion and the folding angle may work better. Against our expectation, for the target shapes with the random torsions and folding angles (#11-15) and the initial shape from the other groups (#1-10), the average gaps were larger than that of Table 2. But the results were somewhat better in some cases where the target and the initial shapes are both from the group of random torsions and folding angles (#11-15) having similar shapes. From this observation, we considered that the naive optimization method based on the torsion and the folding angle is effective in refining the shape in the final step while it is difficult to reach the goal starting from an initial state far from the target. Table 8 shows the average gaps of three methods: (i) the ruling-based method, (ii) torsion and folding angle-based method, and (iii) the ruling-based optimization followed by the refinement step by the torsion and folding angle-based method. The figures in the table show the average of all results described in subsection 5.1 and 5.2: the initial shapes excluding #12, the target shapes, and the 10 patterns of control points alignments, grouped by the types of the initial and the target shapes. The result shows that the refinement step contributes in decreasing the average gap especially for the target shapes with random torsions and folding angles (#11-15).

		target shape														
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
initial shape	#1		1.72	3.28	3.70	2.95	8.08	11.25	2.19	2.34	9.78	20.00	10.92	11.80	17.39	14.56
	#2	1.69		2.88	5.14	3.76	8.16	11.12	2.62	3.59	9.93	19.64	10.89	12.20	17.71	13.95
	#3	2.58	2.45		3.57	1.74	7.58	10.96	2.67	3.29	8.75	21.22	10.79	11.42	17.09	15.35
	#4	7.78	8.82	8.04		6.72	11.83	14.87	8.43	6.85	12.90	24.20	14.38	13.96	18.87	19.15

#5	2.94	3.66	2.05	1.66		7.03	11.06	2.86	3.04	8.92	21.22	10.97	11.15	15.69	16.43
#6	8.08	8.18	7.65	8.49	7.53		6.07	6.15	9.47	3.29	24.31	16.38	17.50	22.15	19.30
#7	10.19	9.63	9.48	10.29	9.85	3.30		8.66	11.04	1.63	18.98	13.01	15.87	19.11	15.96
#8	0.67	0.71	1.33	1.84	1.10	2.50	7.64		0.78	5.17	10.15	3.94	4.94	7.66	5.50
#9	0.96	3.10	2.68	1.58	1.98	6.09	10.79	2.60		9.45	14.96	7.80	6.06	13.90	11.38
#10	9.47	9.50	8.87	9.12	8.35	2.28	2.12	7.62	10.28		19.87	14.63	15.57	17.45	15.83
#11	6.42	5.83	6.22	8.36	7.65	11.04	12.39	6.97	5.93	12.64		0.39	2.55	7.79	1.29
#12	4.75	4.52	4.73	5.33	5.32	10.50	12.07	4.98	4.05	11.71	3.64		0.80	3.54	2.68
#13	2.54	3.08	2.58	2.08	2.34	8.13	10.67	2.91	1.91	9.31	3.98	0.48		1.60	2.30
#14	7.31	8.44	8.11	6.78	7.30	12.90	15.01	9.14	6.26	13.75	9.75	1.44	0.70		5.71
#15	5.37	4.55	6.18	7.64	6.84	10.39	11.44	6.24	5.36	11.28	1.82	1.35	3.29	8.53	

Table 7: Average gap between the target points and the surface control points obtained by the optimization method based on torsion and folding angle. Units in mm.

initial shape	target shape	(i) ruling-based	(ii) torsion and folding angle-based	(iii) ruling-based + refinement step
#1-5	#1-5	0.31	3.86	0.27
#1-5	#6-10	0.29	7.60	0.29
#1-5	#11-15	3.04	15.64	2.44
#6-10	#1-5	0.38	6.02	0.39
#6-10	#6-10	0.26	5.85	0.26
#6-10	#11-15	2.36	14.09	1.88
#11-15	#1-5	0.55	5.78	0.53
#11-15	#6-10	0.34	9.18	0.30
#11-15	#11-15	2.45	3.31	1.86

Table 8: Average gap between the target points and the surface control points for (i) ruling-based method, (ii) torsion and folding angle-based method, and (iii) ruling-based method followed by the refinement step with the torsion and folding angle-based method. Units in mm.

6 CONCLUSIONS

We developed an interactive manipulation method of a curved-folded shape with a given 2D crease by optimizing the curve parameters to minimize the displacement of the surface control points to the target points. To avoid the inefficiency caused by the invalid states with rulings intersections, our method takes two steps: trials to obtain the rulings without the intersections, and the derivation of the torsions and the folding angles which are consistent with the given ruling angles and best approximates the target points. The method is applied as new features of the interactive graphical modeling systems to design the shape of curved fold model, to manipulate the model shape and the stitching of two curved-folded surface patches, with more possible applications. The evaluation using synthetic data shows that, as a whole, the ruling-based optimization well approximates the target shapes. The accuracy, or the average displacement of the control point to the target, was less than 1mm for most of the moderate shapes, and up to 8mm for some extreme shapes, also depending on the distribution of the control points. The process time was 5.54 seconds on average and up to 22 seconds for the most difficult cases.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number JP19K12677, Japan.

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

REFERENCES

- [1] Epps, G.: Robofold and robots. IO, Made by Robots: Challenging architecture at a larger scale, *Architectural Design*, 2014, 68–69. <https://doi.org/10.1002/ad.1757>
- [2] Fuchs, D. Tabachnikov, S.: More on paperfolding, *The American Mathematical Monthly*, 106(1), 1999, 27-35. <https://doi.org/10.2307/2589583>
- [3] Ghassaei, A.; Demaine E. D.; Gershenfeld, N.: Fast, interactive origami simulation using GPU computation, *Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education*, 2018, 1151-1166.
- [4] Kilian, M.; Flöry, S.; Chen, Z.; Mitra, N.; Sheffer, A.; Pottmann, H.: Curved folding, *ACM Transactions on Graphics*, 27(3), 2008, 75:1-9. <https://doi.org/10.1145/1360612.1360674>
- [5] Kilian, M.; Monszpart, M.; Mitra, N. J.: String actuated curved folded surfaces, *ACM Transactions on Graphics*, 36(3), 2017, 25:1-13. <https://doi.org/10.1145/3072959.3015460>
- [6] Narain, R.; Pfaff, T.; O'Brien, J. F.: Folding and crumpling adaptive sheets, *ACM Transactions on Graphics*, 32(4), 2013, 51:1–8. <https://doi.org/10.1145/2461912.2462010>
- [7] Mitani, J.: A design method for 3D origami based on rotational sweep, *Computer-Aided Design & Applications* 6(1), 2009, 69-79. <https://doi.org/10.3722/cadaps.2009.69-79>
- [8] Mitani, J.; Igarashi, T.: Interactive design of planar curved folding by reflection, *Pacific Conference on Computer Graphics and Applications - Short Papers*, 2011, 77–81.
- [9] Muidilova, C.: On mathematical folding of curved crease origami: Sliding developables and parametrizations of folds into cylinders and cones, *Computer-Aided Design*, 115, 2019, 34-41. <https://doi.org/10.1016/j.cad.2019.05.026>
- [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.: The shape space of discrete orthogonal geodesic nets, *ACM Transactions on Graphics*, 37(6), 2018. <https://doi.org/10.1145/3272127.3275088>
- [12] 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>
- [13] Sasaki, K.; Mitani, J.: Simple simulation of curved folds based on ruling-aware triangulation, In *Pacific Graphics Short Papers, Posters, and Work-in-Progress Papers*. The Eurographics Association, 2020.
- [14] Tachi, T.: Generalization of Rigid-Foldable Quadrilateral-Mesh Origami, *Journal of the International Association for Shell and Spatial Structures (IASS)*, 50(3), 2009, 173-179.
- [15] Tachi, T.: Simulation of rigid origami, *Origami 4: The Fourth International Conference on Origami in Science, Mathematics, and Education*, 2009, 175–187.
- [16] Tachi, T.: One-DOF rigid foldable structures from space curves, in *Proceedings of the IABSE-IASS Symposium*, 2011.
- [17] Tachi, T.: Composite rigid-foldable curved origami structure, in *Proceedings of Transformables*, 2013.
- [18] 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>
- [19] 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>
- [20] Watanabe, Y.; Mitani, J.: Modelling the Folding Motions of a Curved Fold, *Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education*, 2018, 1135-1150.
- [21] Watanabe, Y.; Mitani, J.: Visualization of Folding Motion of Rotationally Symmetric Curved Folding, *Computer-Aided Design & Applications*, 17(3), 2019, 513-524. <https://doi.org/10.14733/cadaps.2020.513-524>