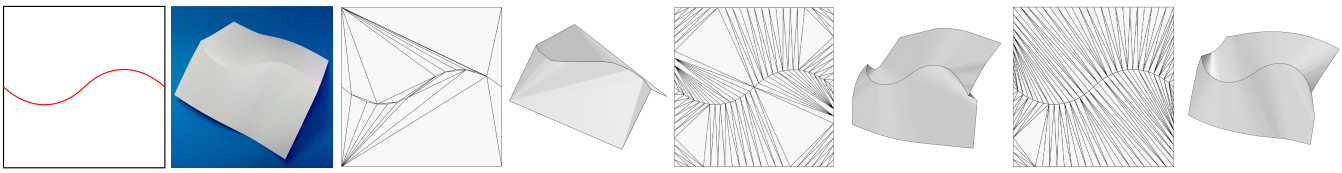
Pacific Graphics (2020)
E. Eisemann, A. Jacobson, and F.-L Zhang (Editors)

Simple Simulation of Curved Folds Based on Ruling-aware Triangulation

K. Sasaki[1] and J. Mitani[1]

[1]University of Tsukuba, Japan



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

Figure 1: *(a) A crease pattern and a photograph of the folded paper. Triangulated crease patterns and simulated shapes with (b) the Origami Simulator, (c) constrained Delaunay triangulation, and (d) proposed triangulation.*

Abstract

Folding a thin sheet material such as paper along curves creates a developable surface composed of ruled surface patches. When using such surfaces in design, designers often repeat a process of folding along curves drawn on a sheet and checking the folded shape. Although several methods for constructing such shapes on a computer have been proposed, it is still difficult to check the folded shapes instantly from the crease patterns.In this paper, we propose a simple method that approximately realizes a simulation of curved folds with a triangular mesh from its crease pattern. The proposed method first approximates curves in a crease pattern with polylines and then generates a triangular mesh. In order to construct the discretized developable surface, the edges in the mesh are rearranged so that they align with the estimated rulings. The proposed method is characterized by its simplicity and is implemented on an existing origami simulator that runs in a web browser.

CCS Concepts

• ***Computing methodologies*** → ***Mesh models; Mesh geometry models;***

1. Introduction

Folding a thin sheet material such as paper along curves creates a developable surface composed of ruled surface patches. The shapes are often used in the design of origami works, boxes for packaging, leather products, and architecture. When making a prototype model of such a shape, designers often repeat a process of folding along curves drawn on a sheet and checking the folded shape until the desired shape is obtained. Time and effort are required in order to repeat the trial and error processes because the folded shape is uncertain at the stage of drawing a *crease pattern*, which is a set of lines or curves to be folded. Although several methods for constructing such shapes on a computer have been proposed [KFC*08, Mit12, WM19, RHSH19], it is still difficult to obtain the folded shapes instantly from the crease patterns with curves because of the limitation of the target shapes or the requirement of a large amount of user interaction. In the present paper, we propose a simple method that approximately realizes a simulation of curved folds with a triangular mesh from its crease pattern. The input is a crease pattern, which may have curves, with a mountain-valley assignment. The output is a triangular mesh model representing the shape expected by folding the crease pattern. The proposed method is characterized by its simplicity and ease of implementation. It is implemented on an existing origami simulator that is written in JavaScript and runs in a web browser on standard PC and smartphones. The shape obtained from a given crease pattern is not uniquely determined but can be varied depending on the folding angle of the creases. In addition, when a physical sheet of paper is folded, the obtained shape is slightly different from the ideal developable surface due to small distortions caused by tiny buckling of the material near the creases [LYG18]. Furthermore, the shape can drastically deform without changing its crease pattern if the curved creases are twisted in space. For these reasons, generating a folded

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.

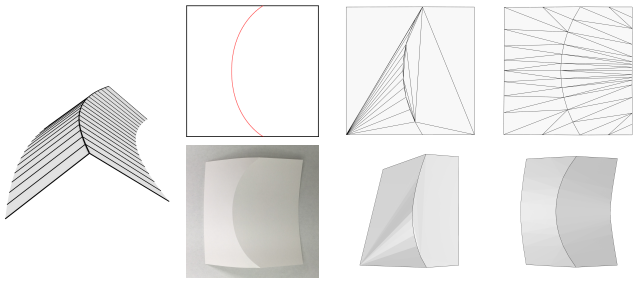


Figure 2: *(a) A curved fold and rulings. (b) A crease pattern and a photograph of the folded shape. (c) A triangle mesh generated without considering rulings and the simulation result. (d) A triangle mesh that includes rulings at edges and the simulation result.*

shape from only a crease pattern is not a trivial problem. The goal is to provide an instant tool that enables designers to grasp a rough shape quickly. We assume the natural state of the folded paper as it is placed on a table without any additional force or gluing. Based on observations of the folded paper, the following preconditions were made for the shapes targeted in the proposed method: 1) the torsion of the folded curves is likely to be zero, and 2) the folding angles tend to remain constant.

The proposed method is based on the Origami Simulator developed by Ghassaei et al. [GDG18], which runs on a web browser and can show the shape obtained by folding an input crease pattern.The limitation of the Origami Simulator is that only straight creases are supported. The simulator generates a triangular mesh by dividing the area surrounded by creases using the ear clipping method. The simulator then deforms a mesh using a simple mass-spring-damper model to simulate the folding motion. The meshes generated by this method are not suitable for curved folds (Figure 1(b)). A curved surface formed by bending non-elastic material such as paper is called a developable surface and is represented as a trajectory of a line element called a *ruling* (Figure 2). Since a shape created by curved folds is composed of ruled surface patches, only a mesh with triangles that include rulings on their edges can represent a discretized developable surface (Figure 2(d)). However, determining the arrangement of rulings is difficult or impossible when a folded shape is unknown. The proposed method uses two empirical rules to estimate the initial placement of rulings from a crease pattern: 1) rulings on a smooth developable surface tend not to align conically (Section 3.2), and 2) rulings and the curved creases tend to align orthogonally on the unfolded pattern (Section 3.3). By considering these rules, the proposed method generates a triangular mesh suitable for simulating curved folds while keeping the mesh data size small. In order to confirm the effectiveness of the proposed method, we implemented it on the Origami Simulator and performed experiments.

The contributions of this paper are summarized as follows:

- A simple method to simulate curved folds from a crease pattern is proposed based on a triangulation of the crease pattern with an estimation of the ruling arrangement.
- The proposed method can generate simulation results similar to shapes of actual folded paper with various crease patterns.
- It is confirmed that the proposed method has advantages in terms of mesh size, simulation cost, ease of implementation, and robustness.

2. Related work

The Origami Simulator developed by Ghassaei et al. [GDG18] is a simple mass-spring-damper based simulator on which we implemented the proposed method. The crease pattern to be input to the simulator is a set of line segments with a predefined mountain-valley assignment and folding angles. The springs are placed between vertices in order to maintain the isometric properties of a non-elastic material, and a hinge-spring is placed at each crease to make it folded at a specified angle. The simulator deforms the mesh by calculating the force on each vertex at each time step. Before the simulation, the simulator triangulates the area surrounded by the creases or borderlines in the input. The edges added by this triangulation are restricted to be as flat as possible during the simulation. Although curved creases can be input to the simulator by discretizing them with polylines, improper triangulation will not yield smooth surfaces. Kilian et al. proposed the PQStrip representation for discretized developable surfaces [KFC*08]. The PQStrip is a strip of planar quads. Each quad contains two rulings on the edges facing each other. In order to discretize the developable surfaces in this manner, it is important to align the edges with the rulings. The concept is the same, although the elements are not quads but triangles.

A recent study by Rabinovich et al. [RHSH19] shares our objectives. They proposed a novel approach for simulating curved folds using discrete orthogonal geodesic nets called DOG nets. A unique feature of their method is that rulings do not need to be considered. Although this approach does not require a mountain-valley assignment, users need to place and manipulate the control points for the deformation. Solomon et al. [SVWG12] proposed a method for generating smooth surfaces by deforming and subdividing the folded lines or rulings that the user inputs so that their developability is maintained. This method can reproduce the shape obtained by curved folding, but requires the user to input both the fold lines and the approximate rulings. Zhu et al. [ZIM13] introduced the concept of soft folding. Their method generates shapes of folded thick and soft materials. A user specifies sharp and soft folds in a crease pattern, along with the folding angle and sharpness of each fold. The pattern is divided into fine meshes, and a globally folded shape is obtained by assembling the locally folded small patches. By specifying sharp folds, the shapes that we want in the present paper can be generated, but implementing their method appears to be very difficult, and a fine mesh model is not suitable for quick simulation. Furthermore, their method generates shapes that cannot actually be made when inappropriate crease patterns are given. Watanabe et al. [WM19] proposed a method for generating and visualizing the folding motion from a crease pattern and its folded state. This method is only compatible with crease patterns in which the creases are placed rotationally symmetrically.

In terms of differential geometry, Fuchs and Tabachnikov [FT99] described the relationship among a curved crease and rulings in

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.

the flatten state (2D), their spatial state (3D), and the folding angle. Demaine et al. [DDH*15] proposed a method for obtaining the ruling configuration from a specific crease pattern by clarifying the properties between curved creases and rulings. Demaine et al. [DDH*18] revealed that the configuration of rulings can be decided uniquely from a crease pattern under some specific conditions. Furthermore, at the same time, they also clarified that there exist many crease patterns in which the mathematically appropriate configuration of rulings does not exist. Since the proposed method uses a rough estimation of the configuration of rulings for triangulation, the mathematical correctness is not guaranteed but is robust.

3. Proposed method

The proposed method is for generating a triangular mesh for simulation on the Origami Simulator [GDG18]. As the first step, the curved creases in the crease pattern are discretized to polylines (Section 3.1). Next, in order to include triangles with the estimated ruling on the edges in the mesh, constrained Delaunay triangulation is used to generate the initial state of the mesh from the input crease pattern (Section 3.2). Then, the edges in the mesh are rearranged based on the relationship between the curved creases and the estimated ruling configuration (Section 3.3).

3.1. Discretization of an input crease pattern

Since the Origami Simulator runs with a triangular mesh, curves in the crease pattern need to be approximated by polylines. First, a polyline is created by placing vertices on each curve at equal intervals. In our implementation, the curves are represented by cubic Bézier curves and are equally divided in the parametric space. Then, if the distance between a line segment connecting two adjacent vertices and the midpoint of the curve between the two points is greater than the specified value, a new vertex at this midpoint is inserted to the polyline. This process is repeated until no addition is needed. Borderlines are also divided into segments at equal intervals. The effect of the distance intervals for the vertices on the results is discussed in Section 3.2. The mountain-valley flags are assigned at this stage to each line segment.

3.2. Triangulation

When triangulating the input crease pattern, the pattern should include triangles with edges along with the ruling. Demaine et al. [DDH*15] proved that a curved crease with an incident *cone ruling*, i.e., multiple rulings sharing a common end-point, cannot be folded smoothly and must form a kink (a point at which the first derivative is not continuous). It is also known that the bending energy at the apex of a cone diverges [AP10]. Therefore, the cone ruling has to be excluded. In order to prevent several edges from connecting to one vertex, we use the constrained Delaunay triangulation (CDT) for generating the initial triangle mesh instead of the ear clipping method originally used in the Origami Simulator. This is because the Delaunay triangulation produces fewer flat triangles. The constraint line segments of CDT, in this case, are polylines of creases and borderlines. The reason for dividing the borderline in the previous step is to increase the number of vertices used in this triangulation so that a few triangles share one vertex.

3.3. Estimation of the ruling configuration

In order to make the edges of the triangular mesh closer to rulings, we rearrange the edges based on empirical rules for curved creases and rulings. Fuchs and Tabachnikov wrote that *if the fold is a non-closed arc, the folded paper tends to occupy such a position that the ridge lies in a plane*. This means that a planar curve is a physically stable state of a curved fold. The relationship between ruling directions (Figure 3) and fold angles of a curved fold is expressed by the following three equations, which were introduced by Tachi [Tac11] based on the description in [FT99]:

$$\begin{aligned} \kappa_{2D}(s) &= \kappa(s)\cos\alpha(s), \\ \cot\beta_L(s) &= \frac{\alpha'(s)-\tau(s)}{\kappa(s)\sin\alpha(s)}, \\ \cot\beta_R(s) &= \frac{-\alpha'(s)-\tau(s)}{\kappa(s)\sin\alpha(s)}, \end{aligned} \tag{1}$$

where s is the arc length parameter, $\kappa_{2D}(s)$ is the curvature of the crease drawn in the crease pattern, $\kappa(s)$ and $\tau(s)$ are the curvature and torsion, respectively, of the crease on the folded surface, α is the angle between the ruling and the osculating plane, and $\beta_L(s)$ and $\beta_R(s)$ are the angles between the rulings and the crease on the crease pattern.

We assume that creases are not twisted after folding, i.e.,$\tau(s) = 0$, and the folding angle is constant, i.e.,$\alpha'(s) = 0$. If these values are assigned to the above equations, then we obtain $\cot\beta_L(s) = \cot\beta_R(s) = 0$, which means that rulings are orthogonal to the tangent of the curved creases in the crease pattern. This assumption is used to rearrange the triangle mesh.

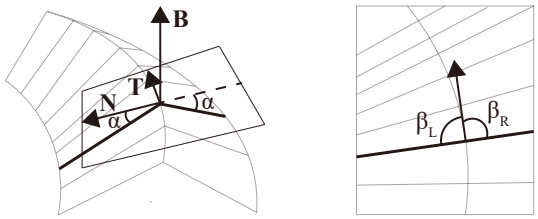


Figure 3: *A folded state (left) and an unfolded state (right) of a curved crease and rulings.*

In order to rearrange the mesh, we use the edge-swap operation based on two evaluation criteria: 1) the number of edges incident to the same vertex (assumed to be small in order to avoid cone rulings), and 2) the value by which to evaluate the orthogonality between rulings and creases on the planar state. Suppose that the edge connects the vertices V_A and V_C, which are on different crease polylines, as illustrated in Figure 4. Note that all vertices are either on a crease polyline or a borderline. We define the angle $\theta_i (i = 0,1,2,3)$ between the edge and creases. If we consider this edge as a ruling, then the values of θ_0 and θ_1 should be approximately the same, and the values of θ_2 and θ_3 should be approximately the same. In order to evaluate this property in terms of the squared error metric, we define E as follows:

$$E = (\theta_0 - \theta_1)^2 + (\theta_2 - \theta_3)^2 \tag{2}$$

For the case in which one of the vertices connected by the edge,

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.

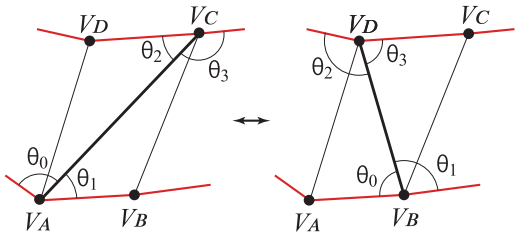


Figure 4: *Edge-swap operation.*

e.g., vertex V_C, is on the borderline, we define E as follows:

$$E = 2(\theta_0 - \theta_1)^2 \tag{3}$$

If both vertices are on the borderline, the value of E is set to infinity. In the proposed method, the following steps are repeated to select an edge to which an edge-swap operation is applied. First, the edges that satisfy all of the following conditions are selected. (1) Neither crease line nor borderline. (2) No more than three edges share the vertex V_B and V_D. (3) The value of E becomes smaller by the edge-swap operation. Then, the edge-swap operation is applied to the edge that has the smallest value of E after the edge-swap operation. This is repeated until there are no more edges available for edge swapping. In our experiments, the iterative processes were completed in all cases. Figure 5 shows a comparison of the initial state of the meshes generated by the proposed method.

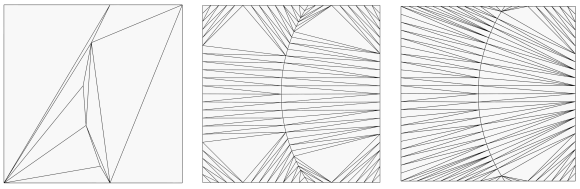


Figure 5: *Comparison of triangular meshes. (a) Mesh generated by the Origami Simulator, (b) initial mesh of the proposed method generated by the constrained Delaunay triangulation, and (c) mesh after improvement using edge-swap operations.*

4. Result

We experimented with the crease patterns shown in Figures 1 and 8. The photographs of folded paper pieces are for comparison. The pieces were in natural states placed on the table after they had been folded and released from the hand. The crease patterns were chosen to ensure that the proposed method could accommodate various types of curved folds. The mountain creases, valley creases, and borderlines are indicated in red, blue, and black. The widths of all of the crease patterns were set to 150 mm. The interval by which to divide the creases and borderlines was set to 7 mm. The tolerance for approximating curves with polylines was set to 0.07 mm. The folding angles were set to 108° in all crease patterns, except for those in Figures 8(g), 8(h), 8(i) (72°), and 8(l) (54°) assuming the natural state. The angle can be specified interactively on the Origami Simulator. The Origami Simulator merges vertices for which the distance is smaller than a threshold when a crease pattern is given. The threshold was set to 1.0 mm, which is the default value, for all crease patterns, except for Figures 8(k) and 8(l) (0.7 mm). This value was empirically decided. The simulation was run in the web browser Firefox 77.0.1 on a desktop PC with a 3.40-GHz Intel Core i7 CPU and 64 GB of RAM.

We measured the warp angle [PS07] in order to evaluate how well the resulting models were able to approximate ruled surface patches. Figure 6 illustrates how the warp angle of the edge connecting the vertices $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ is calculated. Here, $\boldsymbol{r}$ is the vector from $\boldsymbol{p}_i$ to $\boldsymbol{p}_j$, tangent vector $\boldsymbol{T}_i$ at $\boldsymbol{p}_i$ is the vector from $\boldsymbol{p}_{i-1}$ to $\boldsymbol{p}_{i+1}$, and the tangent vector $\boldsymbol{T}_j$ at $\boldsymbol{p}_j$ is the vector from $\boldsymbol{p}_{j-1}$ to $\boldsymbol{p}_{j+1}$.

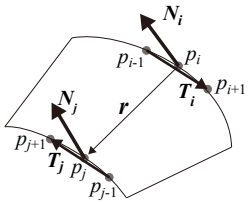


Figure 6: *Warp angle measurement.*

The normal vectors $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ at $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are calculated as $\boldsymbol{N}_i = \boldsymbol{T}_i \times \boldsymbol{r}$ and $\boldsymbol{N}_j = \boldsymbol{T}_j \times \boldsymbol{r}$, respectively. The cross product of $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ can be expressed as $|\boldsymbol{N}_i \times \boldsymbol{N}_j| = |\boldsymbol{N}_i||\boldsymbol{N}_j| \sin\theta$, where θ is the warp angle. If $\boldsymbol{r}$ is a ruling, then $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ are parallel and $\boldsymbol{N}_i \times \boldsymbol{N}_j = 0$. The average warp angles for all edges, excluding those with folding angles of less than 0.5°, that are not involved in the shape of the surface are measured.When the value is smaller, we can say that the edges are closer to the rulings. In addition, we also measured the average of the approximated strain [GDG18], which indicates how much the edges in the mesh are stretched.

Figure 1 shows the results of simulations of the crease pattern with a single curved crease. The average warp angle was 5.92° in (c) and 2.97° in (d). This indicates that the model with the proposed method better approximates the ruled surfaces. The total computation time including both triangulation and simulation was 4,623 ms. Figure 8 shows the results for various crease patterns. Patterns (a) to (i) are introduced in [Mit19]. Patterns (j) and (k) are examined in [ZIM13], and pattern (l) is an extreme case. The simulation results appear to be similar to those obtained by folding paper, as shown in the middle row. Table 1 shows the statistics for the meshes and the computation time. The numbers of triangles in the models are all less than one thousand, except (l). These are small numbers compared to those for modern standard CG models. The computation time for mesh generation was at most 611 ms. The computation time for simulations, which stop when the positions of the vertices converge, was less than 10 s, except for pattern (f). The average warp angle decreased after the edge-swap operations, except for patterns (j) and (k). The average strain for each pattern was less than 1.5%. For the case in which the ruling is not well estimated, the isometricity may not be preserved. The patterns in Figure 1 and Figures 8(b) through 8(e) are used in [RHSH19]. When comparing the results, these patterns are approximately identical.

As a limitation of the proposed method, there are cases in which appropriate shapes cannot be generated, depending on the crease patterns and parameter settings.Figure 8(m) shows the simulation results for the same pattern as pattern (i), with an interval for dividing the creases and borderlines set to 7 mm and the tolerance of approximating curves with polylines set to 0.21 mm. Figure 8(n)

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.

shows the results for an interval of 14 mm and a tolerance of 0.07 mm. Although the accuracy of the approximation for pattern (m) is lower than that for pattern (i), the appearances of the folded shapes are approximately identical. For pattern (n), the simulation of the fold did not start because the non-triangulated area bounded by the fold line and the contour line (indicated by arrows) prevented deformation.

Figure 7 shows the results for an inappropriate crease pattern having a non-foldable crease introduced in [Mit19], and a non-foldable mountain-valley assignment, i.e., all creases are assigned as mountains. The simulator was not able to generate the shapes of the creases to be folded. Since the triangulation does not make internal vertices, no buckling is made. This indicates that the proposed method is also useful for determining inappropriate crease patterns for origami design with smooth surface.

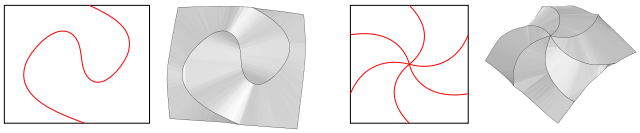


Figure 7: *Simulation results with inappropriate crease patterns having a non-foldable crease (left) or a non-foldable mountain-valley assignment (right).*

Table 1: *Result of experiments. #F indicates the number of faces. Gen and Sim indicate the calculation time for mesh generation and simulation. Strain indicates the average strain. WA w and WA w/o indicate the average warp angles with and without the edge-swap operations, respectively.*

Fig.	#F	Gen (ms)	Sim (ms)	Strain (%)	WA w/o (deg)	WA w (deg)
1(d)	148	82	4541	0.30	5.92	2.97
8(a)	140	66	6722	0.12	1.85	1.46
8(b)	364	142	4246	0.81	5.71	4.79
8(c)	404	101	4395	0.81	4.41	4.38
8(d)	310	85	4048	0.41	1.68	1.43
8(e)	310	93	3051	0.40	1.79	1.66
8(f)	405	113	10672	0.60	2.64	2.59
8(g)	340	70	8315	1.18	6.90	6.03
8(h)	634	317	9477	0.75	7.59	7.46
8(i)	838	440	7294	0.77	3.68	3.34
8(j)	400	194	5454	1.49	8.49	9.17
8(k)	598	114	6425	0.34	3.92	4.01
8(l)	1822	611	3111	0.81	4.78	4.00

5. Conclusions and future study

In the present paper, we proposed a method for approximately simulating curved folds from a crease pattern alone. Triangulation of a crease pattern based on an estimation of a ruling arrangement worked well for the simulation. One of the limitations is that, like many other existing methods, the proposed method does not address self-penetration. In addition, because of our assumption that the crease curves are not twisted in space, the outputs are limited to the shapes that are most stable. In the future, we intend to add user interactions to the proposed method. Being able to handle the shape with control points and to glue some points on the surface together increases the range of shapes that can be produced. Furthermore, interactive editing of the crease pattern will make the design process more efficient. Verification, including proof of convergence of the edge-swap operations is also a subject for future study.

Acknowledgment

The present study was supported by JST CREST Grant Number JPMJCR1911, Japan.

References

[AP10] AUDOLY B., POMEAU Y.: *Elasticity and Geometry: From Hair Curls to the Non-Linear Response of Shells*. Oxford Univ Pr on Demand, 2010. 3

[DDH*15] DEMAINE E. D., DEMAINE M. L., HUFFMAN D. A., KOSCHITZ D., TACHI T.: Characterization of curved creases and rulings: Design and analysis of lens tessellations. *Origami 6* (2015), 209–230. 3

[DDH*18] DEMAINE E. D., DEMAINE M. L., HUFFMAN D. A., KOSCHITZ D., TACHI T.: Conic crease patterns with reflecting rule lines. *Origami 7* (2018), 573–589. 3

[FT99] FUCHS D., TABACHNIKOV S.: More on paperfolding. *The American Mathematical Monthly 106*, 1 (1999), 27–35. doi:10.2307/2589583. 2, 3

[GDG18] GHASSAEI A., DEMAINE E. D., GERSHENFELD N.: Fast, interactive origami simulation using gpu computation. *Origami 7* (2018), 1151–1166. 2, 3, 4

[KFC*08] KILIAN M., FLÖRY S., CHEN Z., MITRA N. J., SHEFFER A., POTTMANN H.: Curved folding. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 1–9. doi:10.1145/1360612.1360674. 1, 2

[LYG18] LEE T.-U., YOU Z., GATTAS J. M.: Elastica surface generation of curved-crease origami. *International Journal of Solids and Structures 136-137* (2018), 13 – 27. doi:10.1016/j.ijsolstr.2017.11.029. 1

[Mit12] MITANI J.: Column-shaped origami design based on mirror reflections. *Journal for Geometry and Graphics 16* (01 2012). 1

[Mit19] MITANI J.: *Curved-Folding Origami Design*. CRC Press, 2019. 4, 5

[PS07] PÉREZ F., SUÁREZ J. A.: Quasi-developable b-spline surfaces in ship hull design. *Computer-Aided Design 39*, 10 (2007), 853–862. doi:https://doi.org/10.1016/j.cad.2007.04.004. 4

[RHSH19] RABINOVICH M., HOFFMANN T., SORKINE-HORNUNG O.: Modeling curved folding with freeform deformations. *ACM Trans. Graph. 38*, 6 (Nov. 2019). doi:10.1145/3355089.3356531. 1, 2, 4

[SVWG12] SOLOMON J., VOUGA E., WARDETZKY M., GRINSPUN E.: Flexible developable surfaces. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1567–1576. doi:10.1111/J.1467-8659.2012.03162.X. 2

[Tac11] TACHI T.: One-dof rigid foldable structures from space curves. In *Proceedings of the IABSE-IASS Symposium* (2011), pp. 20–23. 3

[WM19] WATANABE Y., MITANI J.: Visualization of folding motion of rotationally symmetric curved folding. *Computer-Aided Design and Applications 17* (09 2019), 513–524. doi:10.14733/cadaps.2020.513-524. 1, 2

[ZIM13] ZHU L., IGARASHI T., MITANI J.: Soft folding. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 167–176. doi:10.1111/cgf.12224. 2, 4

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.

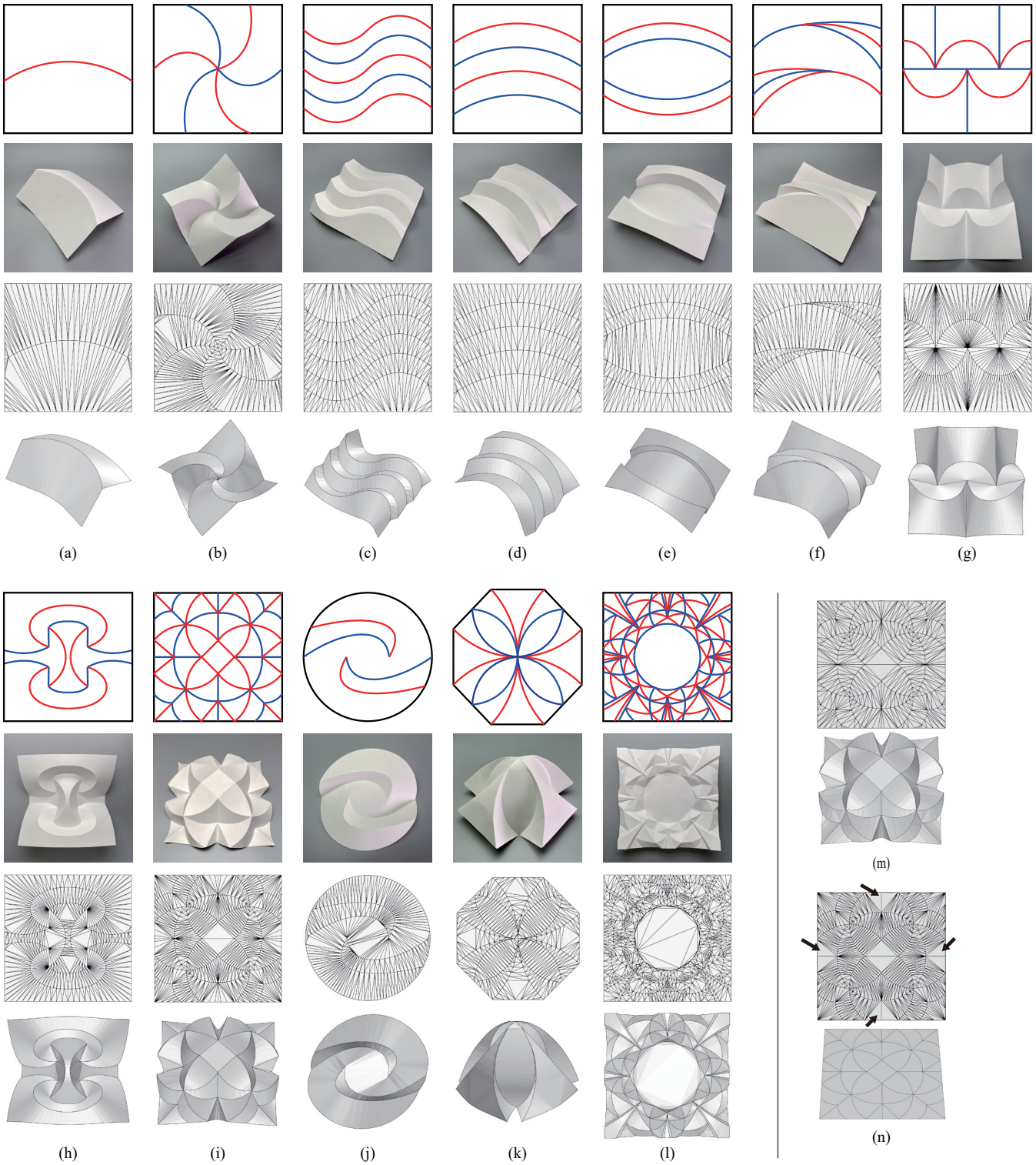


Figure 8: *Simulation results with crease patterns having multiple curved creases. From top to bottom: Input crease patterns, photographs of the folded paper pieces, the triangle mesh, and the simulation results.*

© 2020 The Author(s)
Eurographics Proceedings © 2020 The Eurographics Association.