Modeling and Visualization of Curved Folding

September ２０１９

Yuka Watanabe

Modeling and Visualization of Curved Folding

Graduate School of Systems and Information Engineering

University of Tsukuba

September　２０１９

Yuka Watanabe

Abstract

Curved folding is a field of origami that is becoming popular in recent years, attracting many people by its beautiful appearance. The creases on the flat paper are in the shapes of lines and curves. By folding the paper along the creases, a piece of paper becomes a 3D shape. While such paper arts have been created by many artists and hobbyists, the behavior of the paper in curved folding is not clear. The folding motions, or the shapes of the paper in between the flat unfolded state and the final folded state, are not easy to clarify and to visualize because the curved surface has the flexibility to change its bending directions. In other words, the ruling, the straight lines on a developable surface perpendicular to the bending direction, transits on the surface while the paper is being folded. Nonetheless, visualizing the paper shape being folded has a significance in using the curved folding technique in many applications, such as metal plate manufacturing in curved folded shape using machines. Understanding the shape of the paper while being deformed is essential for the automation of the process. It may also be useful in designing a new crease pattern or help users learn how to fold the curved folding efficiently.

In this thesis, prototypes of a graphical user interface (GUI) system are proposed, which model and visualize the curved folding while allowing the ruling transition during the folding motion. Through the GUI, the user can design the shape of the paper by changing the 2D or 3D shape of the curved crease and its folding angle. The curved surfaces are represented by quad strips adjacent to the curved crease, having the quads divided by the rulings. As the user changes the shape or the folding angle of the crease, the directions of the rulings are derived, and the paper shape is generated. In the simulation of the folding and the unfolding motions, the folding angle is changed continuously within the original angle and zero, the flat state. Then the shapes of the quads are updated at every time step of the motion as the rulings are recalculated from the new curve shape and the folding angle.

To start the research, we performed some preliminary experiments about the folding motion simulation using the 3D models while prohibiting the moving of the rulings. This experiment was carried out to verify the necessity of the model supporting the ruling transition while the bending direction of the paper changes. The result shows that the rigid folding method does not simulate such case, and the spring-mass model supports with some strain. Then, an initial attempt to estimate the paper shapes being folded is introduced. The paper with some color pattern printed was captured by RGBD camera to obtain the 3D shape of the paper. This trial had been pended due to the low performance of the camera but demonstrated the necessity of the 3D model of the curved folded paper, which leads to our research.

As a basic method of the research, the process of modeling a piece of paper with one curved crease is first introduced, including the calculation method of the discretized curved fold model, the procedure of user manipulation, and the method of evaluating the developability and the visual appearances. Then, the special cases where the curved crease contains an inflection point is introduced, with a solution to such cases. As the further development, the method to deal with multiple creases are brought up, such as the user interface to add some creases on the existing curved surfaces. The method

carries some difficulties in adjusting the shape of the added crease so that the derived rulings are not crossing on the paper, which we proposed some solutions. Another achievement is the modeling method of rotationally symmetric curved folding. It is realized by placing some simple segments of curved folded surfaces in rotational symmetry. The folding motion of such shape which requires the rulings to transit in the intermediate state was visualized successfully with small gaps between the segments. For all 3D models created, the developability of the polygon model is verified numerically and evaluated with the visual comparison with photos of real paper.

Acknowledgments

First and foremost, I would like to express my very great appreciation to my supervisor, Professor Jun Mitani, for insightful comments, suggestions, and beneficial information about curved folding and other related researches. This thesis has been done under his direction and would not have been possible without his helpful advice and encouragement.

I would like to thank Yoshihiro Kanamori, for his advice on the image processing that I was working on at the beginning of my doctoral year, and Professor Kenjiro Miura of Shizuoka University for bringing me to try our basic and primitive method to model origami-sphere, the rotational symmetric curved folding, by his comments in CAD'18. I would also like to thank Professor Hiroyuki Kudo and Associate professor Hotaka Takizawa for reviewing this thesis and giving insightful comments.

My gratitude also goes to my boss and my colleagues in Toshiba, for letting me concentrate on my research papers when I needed, even in the difficult situation of our projects.

Finally, I am grateful to my family for always believing in me and giving me time to work on this thesis.

Contents

List of Figures iii
Chapter 1. Introduction 1
1.1. The area of the research 1
1.2. Motivation 1
1.3. Principal contribution 3
1.4. Outline 7
1.5. Publications 8
Chapter 2. The geometry of curved folding 9
2.1. The geometry of curves and surfaces 9
2.1.1. Curves 9
2.1.2. Surfaces 11
2.1.3. Continuity of curves and surfaces 13
2.2. Modeling of developable surface in discrete form 13
2.3. The calculation of the curved folding 15
2.3.1. Equations 15
2.3.2. Interpretations of the equations 17
Chapter 3. Related Work 19
3.1. Modeling of developable surfaces and curved folding 19
3.1.1. Modeling of developable surface 19
3.1.2. Approximation by developable surface 21
3.1.3. The design method of curved folding 22
3.2. Simulation of a folding motion 23
3.2.1. Simulation with fixed mesh data 23
3.2.2. Dynamic simulation of developable surface and curved folding 24
3.3. Analysis of curved folding 25
3.4. Applications 26
Chapter 4. Simulation of curved folding with fixed rulings 28
4.1. Folding origami-sphere with the rigid folding 28
4.2. Simulation of curved folding on Origami simulator 33
Chapter 5. Estimating paper shape from RGBD images 36
5.1. Method 36
5.1.1. RGBD camera 36
5.1.2. Camera calibration 37
5.1.3. Detection of patterns on RGB images 38
5.1.4. Extraction of developable surface and the rulings 40
5.2. Result and discussion 41
Chapter 6. Curved folding with one crease 43
6.1. Calculation of crease and rulings 43
6.1.1. Discrete expression of the mathematical formula 43
6.1.2. The calculation process of crease and rulings 46
6.2. Designing the curved folding 48

6.2.1. The GUI system 48
6.2.2. The design process of the shape of curved folding 49
6.2.3. Simulation of a folding motion 50
6.2.4. Design of curved folding from ruling directions 51
6.3. Results and discussions 51
6.3.1. Evaluation of the developability 52
6.3.2. Comparison with real paper 54
6.3.3. Applicability of the method 55
Chapter 7. Rectification of curved crease at the inflection point 56
7.1. Rulings at the inflection point 56
7.1.1. Rectification of folding angle 57
7.1.2. Rectification of torsion 59
7.1.3. GUI for the rectification 59
7.2. Results and discussions 60
7.2.1. Evaluation of the developability 60
7.2.2. Comparison with real paper 61
7.2.3. Evaluations using 3D printer object 62
Chapter 8. Curved folding with multiple creases 63
8.1. Adding curved creases on a curved surface 63
8.1.1. User interface and the procedure of adding a crease 64
8.1.2. Calculation of additional crease 67
8.1.3. Cost functions for input restriction and optimization 68
8.1.4. Trimming the paper 70
8.2. Result and discussion 70
8.2.1. Evaluation of the developability 70
8.2.2. Comparison with real paper 71
8.2.3. Other examples 72
Chapter 9. Curved folding with rotational symmetry 73
9.1. Method 73
9.1.1. Overall process 74
9.1.2. Crease pattern generation 75
9.1.3. 3D model generation and interpolation 76
9.1.4. GUI of the prototype system 78
9.2. Result and discussion 80
9.2.1. Developability of one segment 80
9.2.2. Connectivity between the segments 81
9.2.3. Comparison with real paper 81
9.2.4. Discussion 83
Chapter 10. Conclusion and future work 84
10.1. Summary of contributions 84
10.2. Future work 85
References 88

List of Figures

Figure 1-1 Examples of curved folding from flicker [Mitani]. (a) Huffman's Tower. (b) Relief of a wave. (c) Spherical wrapping. 1
Figure 1-2 Developable surface. 2
Figure 1-3 Folding motion of curved crease and rulings as 3D polygon shape (top row) and 2D ruling configuration (bottom row). Black curve shows the curved crease, purple lines show rulings on left side of the crease, and pink lines show rulings on the right side. 3
Figure 1-4 Prototype of GUI system. 4
Figure 1-5 Example of generated curved folding with ruling crossing, failed to be in existable shape. 4
Figure 1-6 Curved folding with curved crease having inflection point. 5
Figure 1-7 Curved folding with multiple creases. The curved crease in the center, between purple and pink rulings, is the original crease. The other two curved creases on the sides are the added creases. 5
Figure 1-8 Fold motion of rotational symmetric curved folding. Top row: rendered 3D model. Middle row: wireframe model with one segment rendered with the rulings. Bottom row: photos of real paper. 6
Figure 2-1 Curve parametrized by arc length s. 9
Figure 2-2 Frenet frame on 3D curve. 9
Figure 2-3 Planes of space curve. (a) Normal plane. (b) Rectifying plane. (c) Osculating plane. 10
Figure 2-4 Elements used for definition of curvature. (a) Derivative of tangent vector. (b) Radius R of osculating circle. 10
Figure 2-5 Surface parametrized by u, v. 11
Figure 2-6 Principal curvatures $k1, k2$ as reciprocal of the radius of osculating circle $R1, R2$ and principal directions $\boldsymbol{V}1, \boldsymbol{V}2$. 12
Figure 2-7 Gauss map. (a) Surface with normal vectors. (b) Gauss map of surface. 12
Figure 2-8 Ruled surfaces. (a) Ruled surface which is not developable. (b) Developable surface. 12
Figure 2-9 Continuity of curves and surfaces. Top row: curves. Bottom row: surfaces. (a) G0 continuity. (b) G1 continuity. (c) G2 continuity. 13
Figure 2-10 Modelling of curved folding. (a) 3D polygon model. (b) Description of elements in 3D polygon model. (c) Creases and rulings mapped onto 2D space. 14
Figure 2-11 Total corner angles adjacent to vertex. 14
Figure 2-12: Definitions of vectors and angles. (a) 3D model. (b) Crease curve with rulings mapped to 2D space. 15
Figure 2-13 Curved folded surface in various folding angles. (a) $\alpha(s) = 0,\ ks = k2D(s)$. (b) $0 < \alpha(s) < \pi 2,\ ks > k2D(s)$. (c) $\alpha(s) \to \pi 2,\ ks \to \infty$. 17
Figure 2-14 Rulings in various folding angles. (a) Constant folding angle, $\alpha'(s) = 0$. (b) Increasing folding angle, $\alpha'(s) > 0$. (c) Decreasing folding angle,

$\alpha'(s) < 0$. 18
Figure 2-15 Rulings in various torsions. (a) Planer curve, $\tau s = 0$. (b) Positive torsion, $\tau s > 0$. (c) Negative torsion, $\tau s < 0$. 18
Figure 3-1 Developable surface composed of developable primitives. (a) Cone spline surface [Chen_04]. (b) Mappings of different cones and cylinders [Hwang_15]. 20
Figure 3-2 Developable surface in the form of parametric surface. (a) Cubic developable Bézier patches [Chu_02]. (b) Developable surface defined by spline curves [Tang_16]. 20
Figure 3-3 Developable surface as a net-based polygon model. (a) Discrete shell origami [Burgoon_06]. (b) DOG nets [Rabinovich_18a]. 20
Figure 3-4 Developable surface as a ruling based polygon model [Kilian_08]. 21
Figure 3-5 3D mesh data composed of developable surfaces [Stein_18]. (a) Original 3D data. (b) Developable polygon data. 22
Figure 3-6 3D mesh data composed of developable strips [Liu_07]. (a) Triangulation of model. (b) Triangulation flattening. 22
Figure 3-7 Developable surface made from boundary curves [Rose_07]. (a) Input boundary curves. (b) Two possible developable surfaces. 22
Figure 3-8 Soft folding [Zhu_13]. 23
Figure 3-9 Flat foldable vault structure [Tachi_13]. 24
Figure 3-10 String actuated folding motion [Kilian_18]. 24
Figure 3-11 Curved folding by DOG nets [Rabinovich_18b]. 24
Figure 3-12 Curved folding by David Huffman. 25
Figure 3-13 Deployable structures folded by curved folds [Lee_13]. (a) Unfolded configuration. (b) Folded configuration. 26
Figure 3-14 RoboFolds [Epps_14]. 26
Figure 3-15 Self-assembling origami propellers [Miyashita_15]. 27
Figure 3-16 D-core joint made by curved folding [Nelson_16]. 27
Figure 4-1 3D models of *origami-spheres* designed by *ori-revo*. (a) Sphere with six flaps, each quad strip having 12 rulings. (b) Sphere with four flaps, quad strip having three rulings. 29
Figure 4-2 The result of the experiment of rigid folding. 32
Figure 4-3 The input crease pattern with one crease. 33
Figure 4-4 The result of Origami simulator for crease pattern in Figure 4-3. Top row: fold percent 50%. Middle row: fold percent 75%. Bottom row: fold percent 100%. 34
Figure 4-5 The input crease pattern of *origami-sphere*. 35
Figure 4-6 The result of Origami simulator for crease pattern in Figure 4-5. (a) Fold percent: 0%. (b) Fold percent: 40%. (c) Fold percent: 60%. (d) Fold percent: 80%. 35
Figure 5-1 Data used in paper shape estimation. (a) Captured data. cu, cv is pixel position in RGB image, du, dv is pixel position in depth image, dz is the depth value on du, dv, cz is the depth value on cu, cv calculated from other parameters, and x, y, z is 3D position of a point in the point cloud. (b) Crease pattern with color pattern. cpx, cpy is the 2D position on the crease pattern.

.....37
Figure 5-2 DepthSense© 325. 37
Figure 5-3 Images used for camera calibration. (a) Depth image. (b) RGB image. 38
Figure 5-4 Detection of colored dots. (a) Input image. (b) Result of detection. (c) Red component of the image. (d) Green component. (e) Blue component. ..39
Figure 5-5 Matching of 3×3 patterns between (a) detected dots and (b) the predefined color pattern. 39
Figure 5-6 Process of detecting the whole pattern. (a) Initial result of 3×3 patterns. (b) Outliers removed. (c) Other detected dots integrated into the whole pattern. 39
Figure 5-7 Projection of detected dots on to 3d data. (a) Color pattern detected on RGB image. (b) Point cloud with color data. (c) Detected color pattern plotted on 3D data. 40
Figure 5-8 Extraction of point cloud on one developable surface. (a) predefined area on the color pattern. (b) Corresponding area in the 3D data. (c) Extracted point cloud. 40
Figure 5-9 Estimation of rulings and making polygon model. (a) Detected dots plotted on 3D space. (b) 3D point cloud plotted on the 2D space, with crease pattern and color pattern. (c) Rulings estimated in 2D space. (d) Polygon model of one surface reprojected in 3D space. (e) Polygon model reprojected to the second surface in 3D space. 41
Figure 6-1 Vectors in Frenet frame calculated as discrete derivatives of three consecutive sample points $\boldsymbol{X}i-1, \boldsymbol{X}i, \boldsymbol{X}i+1$. (a) Tangent vector $\boldsymbol{T}i$. (b) Normal vector $\boldsymbol{N}i$. 45
Figure 6-2 Cumulative vector calculation on sequence of vertices. 45
Figure 6-3 Interpolation and reconstruction of crease curve. (a) 2D curvatures on the control points. (b) 2D curvatures interpolated. (c) Reconstructed 2D curve. 46
Figure 6-4 Polygon strip on the end of crease curve. (a) End of crease curve placed outside the paper. New vertex is shown in red. (b) End of crease curve placed inside of the paper. Extended crease is shown in red dotted line. In both figures, newly added triangles (or quads in some cases) are shown in blue. ...48
Figure 6-5 Prototype of GUI system. (a) 3D polygon. (b) 2D mapped crease and rulings. (c) Graphs of parameters. (d) Control panel. 49
Figure 6-6 The iteration of user input. 50
Figure 6-7 Folding motion. Top: 3D polygon model. Bottom: 2D configuration. 50
Figure 6-8 Indices for evaluation of developability. (a) Sum of corner angles. (b) Flatness of quads. 52
Figure 6-9 2D crease curve and ruling in different resolutions of discretization. The number of vertices on the crease are (a) 20, (b) 40, (c) 60, (d) 80, (e) 100. 53
Figure 6-10 Graph of Table 6-3. 53
Figure 6-11 Graph of Table 6-4. 54
Figure 6-12 The visual comparison. Top row: photos of real paper. Middle row: 3D polygon model. Bottom row: 2D crease pattern with rulings. 54

Figure 7-1 Curved crease having a point of zero-curvature. (a) Crease curve. (b) Rulings with non-zero torsion. (c) Rulings with increasing folding angle. 57
Figure 7-2 Examples of ruled and non-developable surfaces. (a) Surface twisting along straight fold line. (b) Surface with increasing folding angle on straight crease line. 57
Figure 7-3 Elimination of rulings. Rulings are eliminated in the dotted area where the curvature of the crease is close to zero. 57
Figure 7-4 Rectification of folding angle. (a) 3D polygon model (top) and crease and rulings mapped to 2D (bottom) before the rectification. (b) After rectification. (c) Graph of folding angle before and after the rectification. In section (i) folding angle is rectified to be constant. In section (ii) it is interpolated by Bézier curve. 58
Figure 7-5 The range of section (ii) in three types. Top row shows curvature, red dotted graph shows folding angle before rectification, and blue graph shows folding angle after rectification. 58
Figure 7-6 Rectification of torsion. (a) 3D polygon model (top) and crease and rulings mapped to 2D (bottom) before the rectification. (b) After rectification. (c) Graphs of τsks (top) and graph of τs (bottom) before and after the rectification. In section (i), torsion is set to be zero. In section (ii), it is interpolated by Bézier curve. 59
Figure 7-7 GUI for changing rectification parameters. 60
Figure 7-8 2D crease curve and ruling in different resolutions of discretization. The number of vertices on the crease are (a) 20, (b) 40, (c) 60, (d) 80, (e) 100. 60
Figure 7-9 Graph of Table 7-1. 61
Figure 7-10 Graph of Table 7-2. 61
Figure 7-11 The visual comparison. Top row: photos of real paper. Middle row: 3D polygon model. Bottom row: 2D crease pattern with rulings. 62
Figure 7-12 3D printed object with thin metal sheet (top) and paper (bottom). 62
Figure 8-1 Curved folded paper with multiple curved creases. 64
Figure 8-2 GUI for additional curved crease. 65
Figure 8-3 Iteration of user input for additional crease. 65
Figure 8-4 Procedure for adding a curve. (a) Free-form curve drawn by the user. (b) B-spline curve approximating the free-form curve. (c) Additional curved crease in initial state. (d) Additional curved crease after user adjustment. 65
Figure 8-5 Simulation of folding motion. Folding angle of the primary curve is (a) 7 to 26 degrees, (b) 11 to 40 degrees, (c) 38 to 57 degrees. 67
Figure 8-6 Process of calculating additional curved crease and the paper shape. (a) 2D curved crease $\mathbf{X}2DRj\boldsymbol{i}$. (b) 3D curved crease $\mathbf{X}Rj\boldsymbol{i}$. (c) Frenet-frame of additional curved crease. (d) Folding angle of additional curved crease. (e) Ruling angle $\beta RRj\boldsymbol{i}$ of 2D curved crease. (f) Ruling vector $\boldsymbol{rR}Rj\boldsymbol{i}$ of the 3D curved crease. 68
Figure 8-7 Non-injective area caused by rulings crossing. 69
Figure 8-8 Trimming the paper. (a) Free-form curve input by the user. (b) The paper trimmed on the user input curve. 70
Figure 8-9 Visual comparison with real paper. (a), (b): Crease patterns printed.

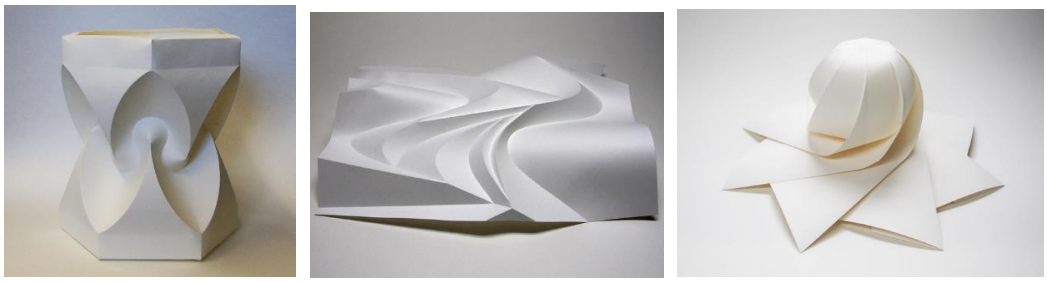
(c)-(e): 3D polygon models of crease pattern (a), in different folding angles, generated by our GUI system. (f), (g): 3D polygon models of crease pattern (b). (h)-(j): Photos of real paper folded by crease pattern (a), corresponding to the 3D model (c)-(e). (k), (l): Photos of real paper folded by crease pattern (b), corresponding to the 3D model (f), (g). 71
Figure 8-10 Other examples of 3D polygon model and the crease pattern with multiple curved creases. Left: Designed shape. Right: Folded state. 72
Figure 9-1 Example of rotationally symmetric curved folding from Flicker [Mitani]. (a) Sphere in cylinder type. (b) Sphere in disk type. 73
Figure 9-2 *Origami-sphere* in disk type with six segments. (a) 3D model. (b) Crease pattern. Both are designed by *ori-revo* [Mitani_09]. 74
Figure 9-3 One segment of origami-sphere rendered with texture and rulings. (a) 3D model. (b) 2D crease pattern. Modeled by our GUI system. 74
Figure 9-4 Process of folding motion generation. Frame 0 is the final folded state. Frame 20 is the flat state. One of the frames in the intermediate state is picked up, modified by the user, and set as a new keyframe. Other frames are interpolated from the adjacent keyframes. 75
Figure 9-5 Crease pattern of one segment. 76
Figure 9-6 Placement of boundary crease. (a) $N = 5$ (b) $N = 6$ (c) $N = 7$ 76
Figure 9-7 Shape of boundary crease. (a) $k2D = 0.0$ (b) $k2D = 0.0005$ (c) $k2D = 0.001$ 76
Figure 9-8 3D shape refinement of a segment. Top row: Before modification. Bottom row: After modification. (a), (d): 3D model. (b), (e): Crease and rulings mapped to 2D. (c), (f): Graphs of parameters. 77
Figure 9-9 Adjustment of segment pose. (a) Before adjustment. (b) After adjustment. 78
Figure 9-10 Adjustment of segment pose. (a) Lines approximating segment boundaries. (b) Lateral edges of equilateral pyramid. 78
Figure 9-11 GUI of the prototype system. 79
Figure 9-12 Flowchart of user operation. 79
Figure 9-13 Curved folding corresponding to the 3D models in Table 9-1. (a) N6, (b) N6_k1, (c) N5, (d) N8, and (e) N6_skew. 80
Figure 9-14 Connectivity of segments. Circles on the graph indicate keyframes whose shape and pose are modified by the user. Squares indicate the frames where only the pose is modified. 81
Figure 9-15 Visual comparison with real paper. (a) N6, (b) N6_k1, (c) N5, (d) N8, and (e) N6_skew. For each sample, top: rendered 3D models, middle: wireframe models with one segment rendered with rulings, bottom: photos of real paper. 83

Chapter 1.
Introduction

This chapter gives an introduction to the modeling and visualization of the curved folding and its folding motion, which is the main topic of this thesis. In Section 1.1, the curved folding, the main research area of the thesis, is first introduced. In Section 1.2, as the motivation of the research, the uncertainty of the curved folded paper shapes, with some explanation about the property of the paper, is discussed, followed by the needs and the significance of understanding them. Then, in Section 1.3, the contribution of our research is listed and explained, which is the modeling, visualizing, and enabling users to edit the curved folded paper model interactively. Finally, the outline of this thesis and the list of publications are introduced in Section 1.4 and 1.5.

1.1. The area of the research

Curved folding is a field of origami that is becoming popular in recent years, attracting many people by its beautiful appearance. The creases on the 2D crease pattern are in the shapes of lines and curves. By folding the paper along the creases, a piece of paper becomes a 3D shape, as some examples shown in Figure 1-1. The geometry of the curved folding is explained in Chapter 2. Their shapes may be used for containers, lamp shapes, etc., and some other applications are introduced in Chapter 3, related work.



(a) (b) (c)

Figure 1-1 Examples of curved folding from flicker [Mitani]. (a) Huffman's Tower. (b) Relief of a wave. (c) Spherical wrapping.

1.2. Motivation

While the paper arts of curved folding have been created by many artists and hobbyists, the behavior of the curved folded paper is not yet so clear. The folding motions, or the shapes of the paper in between the flat unfolded state and the final folded state, are not easy to clarify and to visualize because the curved surface has the flexibility to change its bending directions. The smooth surfaces, made of paper or a thin plate, composing a curved folding is known to be developable surfaces, which may be flattened on to a plane without stretching or tearing (Figure 1-2). When it is bent, the surface contains rulings, or the straight lines on the curved surface perpendicular to the surface's bending

directions, with identical normal vectors at any point on a ruling. Figure 1-3 shows a folding motion of a curved folding with one curved crease on a piece of paper generated by our method. They are shown as a 3D polygon models and its 2D configurations, showing the crease curve and the rulings mapped to the 2D space of the flattened state. In both figures, rulings are depicted explicitly. Unlike the classical paper folding with straight crease lines, the bending direction of the curved surface changes continuously as the paper is being folded, causing the directions of the rulings to transit. The ruling configuration, or the directions of the rulings mapped to the 2D space, also changes if the 3D shape of the crease curve is deformed. By curling or twisting the 3D crease curve, different appearances are created from the same 2D crease under the restriction of developability. Because of this flexibility, it is not easy to predict possible 3D shapes or the folding motion of a curved folding.

Nonetheless, visualizing the paper shape being folded has a significance in using the curved folding technique in many applications. To manufacture a metal plate in curved folded shape using machines, such as RoboFolds [Epps_14], understanding the shape of the plate while being deformed is essential for the automation of the process. It may also be useful in designing a new crease pattern by examining its foldability, that is if the shape is able to be folded without strain or self-collision. Also, the users could learn how to fold the curved folding efficiently, which is sometimes difficult for beginners. In this thesis, we will explore the method of modeling, visualizing, and designing the shape of the curved folding with the user manipulation, considering its folding motion and allowing its surface to change the bending direction while being folded.

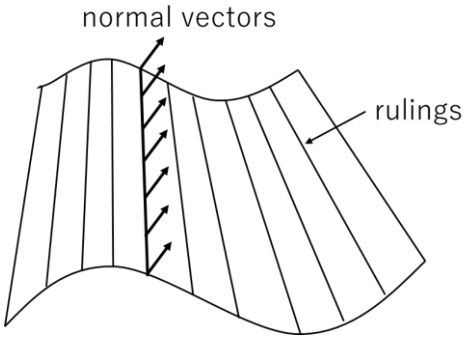


Figure 1-2 Developable surface.

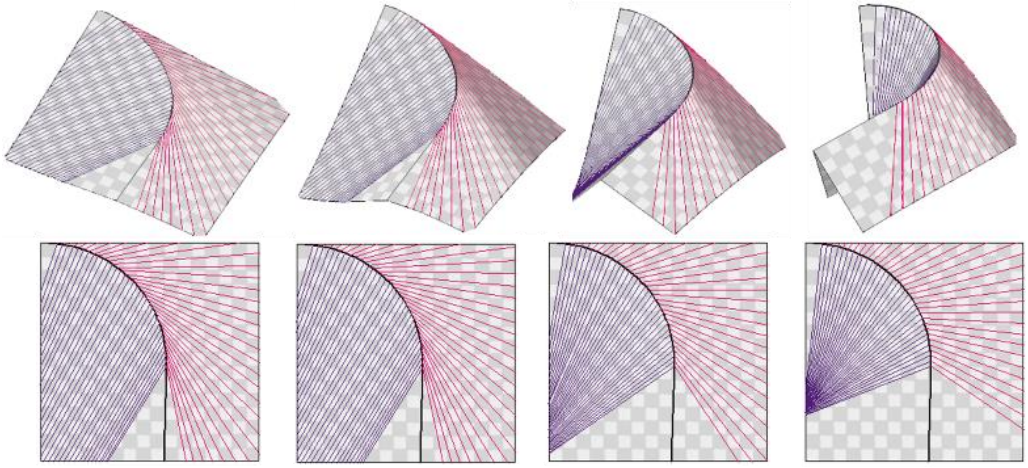


Figure 1-3 Folding motion of curved crease and rulings as 3D polygon shape (top row) and 2D ruling configuration (bottom row). Black curve shows the curved crease, purple lines show rulings on left side of the crease, and pink lines show rulings on the right side.

1.3. Principal contribution

In this thesis, prototypes of graphical user interface (GUI) system are proposed which model and visualize the curved folding, allowing the ruling transition during the folding motion. The main contribution of this thesis is the followings:

1. Interactive GUI system where the user can change the 2D crease curve, the 3D crease curve, and its folding angle while checking the resulting paper shape immediately.
2. Simulation of the folding motion by the interpolation of the folding angle between the folded state and the flat unfolded state, updating the rulings and the paper shape in every time step.
3. A method to handle a special case which the crease curve contains an inflection point where the curvature becomes zero.
4. A method to add curved creases on an existing curved surface and to adjust the curve shape interactively to make a user intended paper shape.
5. Generation of the folding motion of a rotationally symmetric curved folding, by modeling the paper as a set of simple segments aligned in rotational symmetry.

The detail of each item is described below.

As an interactive GUI system, shown in Figure 1-4, the user inputs and adjusts the parameters on some points on the curved crease to edit the shape of the curved folding. The parameters are the 2D and 3D curvatures, torsion, and the folding angle. The system supports the user by two means, by the immediate feedback to enable interactive manipulation and by the restriction of the inputs to keep the curved surface smooth. To enable the interactive manipulation, the system derives the shapes of the crease curve and the curved surfaces from the input parameters immediately and displayed on the screen, using the method introduced in Chapter 6 with the equations in Chapter 2. To keep the generated surfaces to be in smooth shape, it restricts the user input to edit the

parameters only on some points evenly placed on the crease curve. The parameters of all other points on the curve are calculated by the spline interpolation of the user input, making them change smoothly along the curve. As a result, the shape of the crease curve is ensured to be continuous, the directions of the rulings change gradually along the curve, and the curved surfaces are always smooth. Such a system supports the user to keep the parameters to be in the feasible region to make existable shapes in the real world. Otherwise, a naive input may, for example, allow the ruling directions to be fluctuating and to cross each other on the surface, as shown in Figure 1-5, which are to be improved by the interactive editing.

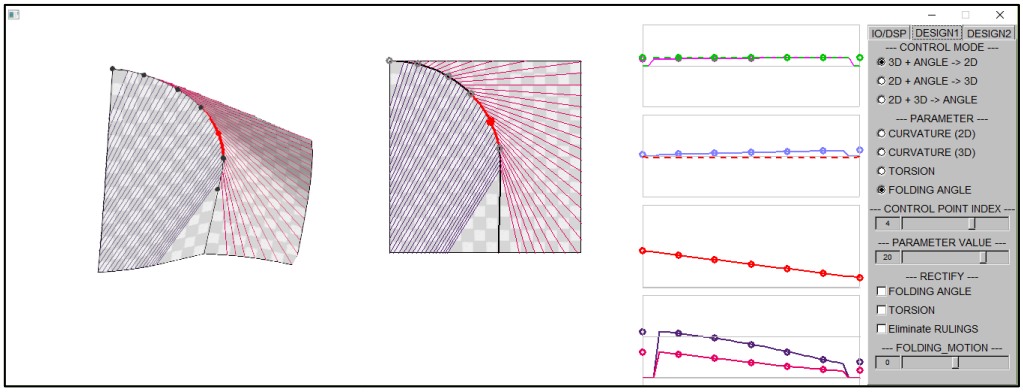


Figure 1-4 Prototype of GUI system.

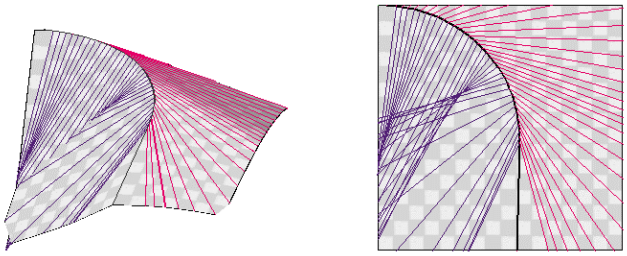


Figure 1-5 Example of generated curved folding with ruling crossing, failed to be in existable shape.

While the design of the curved folding carries some difficulties, the simulation of folding motion is realized by a simple interpolation of the folding angles and the torsions of the 3D crease curve between the user input value and zero, the flat state. As some examples shown in Figure 1-3 and Figure 1-6, the folding motion is the series of 3D paper shapes while being folded. Using the equations and the methods in Chapter 2 and 6, we could derive the 3D shapes and the ruling directions straight forward from the given parameters. The folding motion is generated by carrying out this fast, non-iterative calculation in every time step.

Our system also supports the special case where the crease curve contains an inflection point, which is a point with zero-curvature where the crease is a straight line in a very local region. We added a process to rectify some parameters to meet the

property of the paper shape in the real world. To be specific, the torsion of the 3D crease curve must be zero, and the folding angle must be constant around the zero-curvature point, to be explained in detail in Chapter 7. Instead of directly restricting the user input, we took a method to rectify the parameters afterward. With the rectification, the curved folding with the crease in shape of the letter *S*, having an inflection point in the center, is modeled successfully to be in an existable shape, as shown in Figure 1-6.

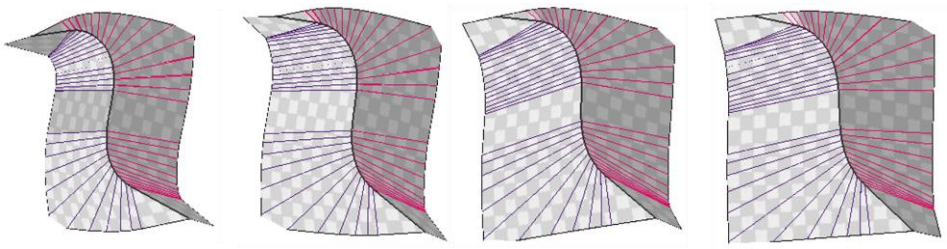


Figure 1-6 Curved folding with curved crease having inflection point.

In adding a curved crease on a curved folded surface, shown in Figure 1-7, the system provides some supportive user interface to help the user generate an intended and existable shape. In the system, the curved crease is approximated by a B-spline curve to keep the smoothness, and the user is allowed to adjust its shape interactively by moving the control points of the parametric curve in 2D space. Yet it is a quite delicate task so that the crossings of the rulings still easily occur by a naive manipulation. To resolve this problem, we provide some additional process to restrict the movement of the control point and to optimize their positions based on some cost functions. This method was useful to some extent when the paper contains only a few creases. But as the user adds more creases, adjusting the curve shape to resolve the rulings crossing would become extremely difficult.

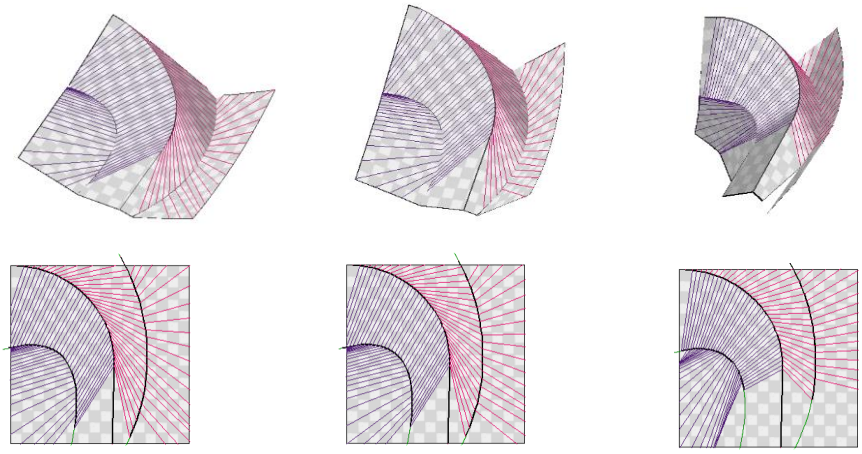


Figure 1-7 Curved folding with multiple creases. The curved crease in the center, between purple and pink rulings, is the original crease. The other two curved creases on the sides are the added creases.

Finally, we propose a method to generate the folding motion of rotational symmetric curved folding, shown in Figure 1-8, integrating the techniques introduced above. In this work, we assume that the paper shape is rotationally symmetric at all stages of the folding motion, though in the real world it may temporarily be asymmetric. The paper is modeled as a group of identical segments containing only one crease, placed in rotational symmetry, shown in the middle row of Figure 1-8 as a rendered piece with the rulings. By editing one segment, the rest of the paper is cloned in accordance with the symmetric property of the crease pattern. Consequently, the curved folding with several creases is modeled on the basis of our previous method, as each segment contains only a small number of creases. Our model supports straight creases in the curved folding by allowing a small gap on the creases located between the segments.

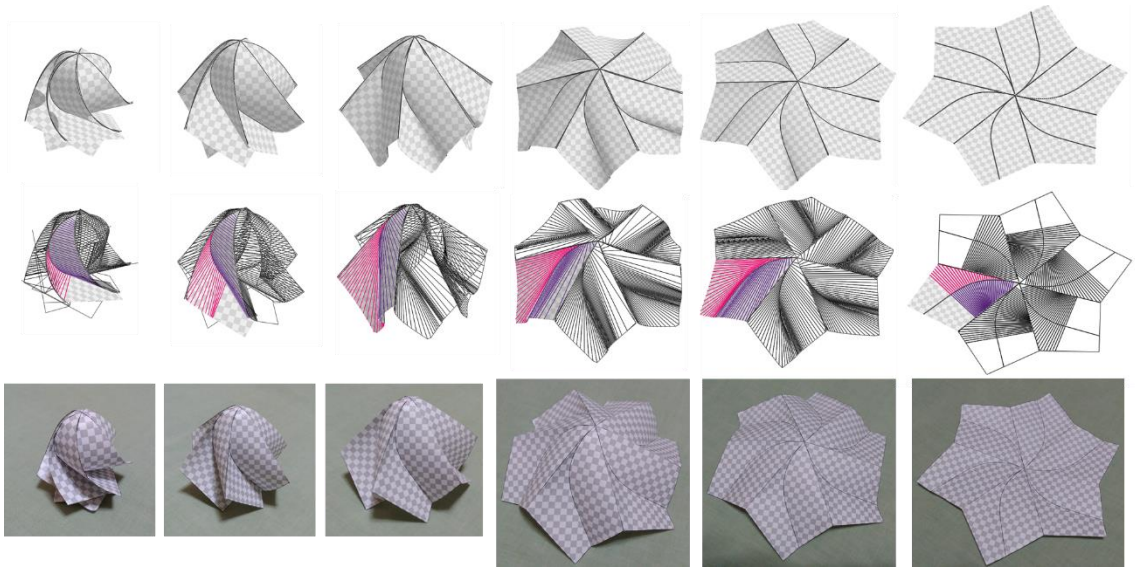


Figure 1-8 Fold motion of rotational symmetric curved folding. Top row: rendered 3D model. Middle row: wireframe model with one segment rendered with the rulings. Bottom row: photos of real paper.

1.4. Outline

This thesis consists of 10 chapters. The geometry of the curves, surfaces, and the curved folding is first explained in Chapter 2. It is based on the general geometry and the equations introduced by Fuchs and Tabachnikov [Fuchs_99], which describes the relation between the folding angles, the shape of the crease curve mapped to the 2D paper, and the shape of the crease curve in 3D space. Then the formula to calculate the rulings and the curved surface is explained. These equations are used in our work in the following chapters.

In Chapter 3, we review the related work. Starting from the modeling method of developable surfaces and the curved folding, we introduce the research on their design methods and the simulation of folding motion, followed by the dynamic simulations proposed in recent years. Then papers about the mathematical analysis on the curved folding are presented, some of which we use in our research. At last, some applications using the curved folding are given.

In Chapter 4, preliminary experiments on the simulation of the folding motion are performed and reported, using the 3D models with fixed rulings. This experiment was carried out to verify the necessity of the model supporting the ruling transition while the bending direction of the paper changes. The result shows that the rigid folding method does not simulate such case, and the spring-mass model supports with some strain.

In Chapter 5, an initial attempt to estimate the paper shapes being folded is introduced. The paper was printed with some color pattern to identify the positions on the paper. Then it was captured by RGBD camera to obtain the images and the 3D point cloud data with textures. This trial had been pended due to the low performance of the camera but demonstrated the necessity of the 3D model of the curved folded paper, which leads to the work in Chapter 6.

The basic method of modeling a piece of paper with one curved crease is introduced in Chapter 6 and also used in the following chapters. The content of this chapter includes the calculation method of the discretized curved folding model using the equations in Chapter 2, the procedure of user manipulation, and the method of evaluating model's developability and visual appearances. In Chapter 7, the special case where the curved crease contains an inflection point is introduced, with the solutions to such cases.

As the further development, the method to work with the curved folding having multiple creases are proposed. Chapter 8 introduces a user interface to add some creases on the existing curved surfaces, with some difficulties in adjusting its shape to make an existable paper shape and some solutions to it. In Chapter 9, another method is proposed to model a rotationally symmetric curved folding by placing some simple segments of curved folded surfaces in rotational symmetry. The folding motion of such shape, which requires the rulings to transit in the intermediate state, was visualized successfully with small gaps between the segments.

Finally, in Chapter 10, we summarize the contribution of our work and discuss the future work of this research.

1.5. Publications

This thesis is based on the following publications:

Journal papers (with peer review)

1. Yuka Watanabe, Jun Mitani: Interactive Modelling of Curved Folds with Multiple Creases Considering Folding Motions. Computer-Aided Design & Applications, 16(3), 2019, 452-465.
2. Yuka Watanabe, Jun Mitani: Visualization of Folding Motion of Rotationally Symmetric Curved Folding. Computer-Aided Design & Applications. (to appear)

International conference papers (with peer review)

1. Yuka Watanabe, Jun Mitani: Modelling the Folding Motions of a Curved Fold. In Origami7: Proceedings of the 7th International Meeting on Origami in Science, Oxford, England, September 5-7, 2018, pages 1135-1150.
2. Yuka Watanabe, Jun Mitani: Interactive Modelling of Curved Folds with Multiple Creases Considering Folding Motions. The 15th annual International CAD Conference (CAD'18), Paris, France, July 9-11, 2018, pages 204-208.
3. Yuka Watanabe, Jun Mitani: Visualization of Folding Motion of Rotationally Symmetric Curved Folding. The 16th annual International CAD Conference (CAD'19), Singapore, Singapore, June 24-26, 2019, pages 197-201.

Chapter 2.
The geometry of curved folding

This chapter explains the geometry used for the modeling of curved folding.

2.1. The geometry of curves and surfaces

A curved folded paper is composed of the *curves*, representing the creases, and the *surfaces*, representing the areas on the paper surrounded by either the creases or the edges of the paper. In this section, definitions and notations about the curve, the surface, and their properties are explained.

2.1.1. Curves

A curve is used to represent a crease of a curved folded paper. It may be defined in 2D space or 3D space and is noted as $\boldsymbol{X}(s)$, the position vector parametrized by the arc length s, as shown in Figure 2-1.

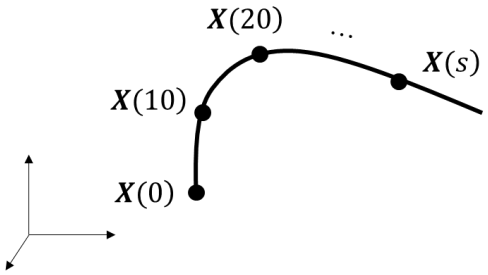


Figure 2-1 Curve parametrized by arc length s.

On the 3D curve, tangent vector $\boldsymbol{T}(s)$, normal vector $\boldsymbol{N}(s)$, and binormal vector $\boldsymbol{B}(s)$ are the orthogonal unit vectors composing the Frenet frame on the curve, as shown in Figure 2-2.

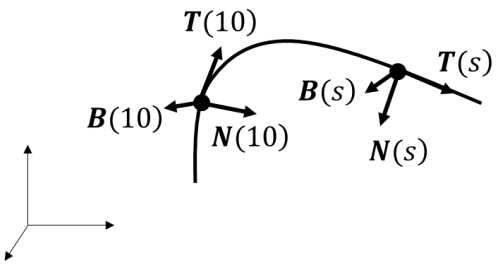
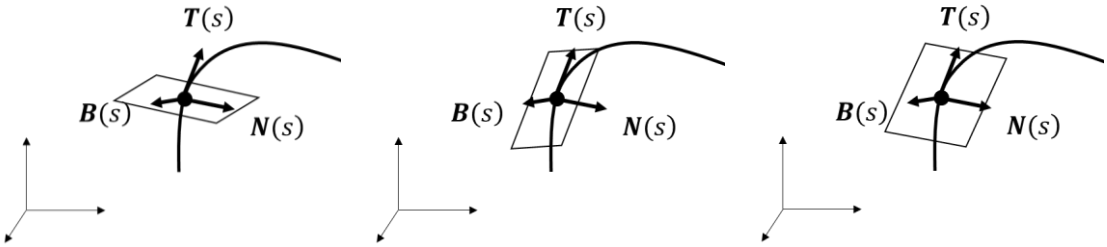


Figure 2-2 Frenet frame on 3D curve.

With the local coordinate system on the curve, the normal plane, the rectifying plane, and the osculating plane are defined as the planes containing two coordinates and perpendicular to the other coordinate, as shown in Figure 2-3.



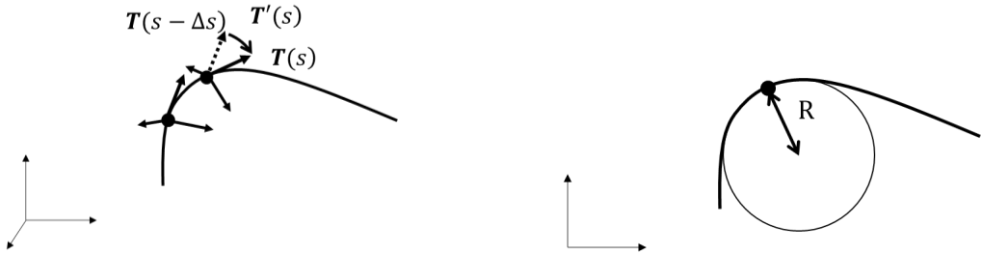
(a) (b) (c)

Figure 2-3 Planes of space curve. (a) Normal plane. (b) Rectifying plane. (c) Osculating plane.

The coordinates in the Frenet frame are calculated as

$$\begin{bmatrix} \boldsymbol{T}(s) \\ \boldsymbol{N}(s) \\ \boldsymbol{B}(s) \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}'(s) \\ \boldsymbol{X}''(s)/k(s) \\ \boldsymbol{X}'(s) \times \boldsymbol{X}''(s)/k(s) \end{bmatrix}, \qquad (2.1)$$

where $k(s)$ is the curvature of the curve. The tangent vector $\boldsymbol{T}(s)$ is calculated as a velocity of the curve respect to the arc length s instead of time t, making the velocity to be a unit vector. The normal vector $\boldsymbol{N}(s)$ is a unit vector in the direction of the acceleration of the curve, or the change in the tangent vector, meaning it is curved to this direction. The curvature $k(s)$ is calculated as the magnitude of the acceleration, or the derivative of the tangent vector. It shows how much it is curved, or the rate of the change in the direction of the tangent vector respect to the arc length, as shown in Figure 2-4(a). For a planer curve, the curvature is also defined as $k = 1/R$, the reciprocal of the radius R of the osculating circle, as shown in Figure 2-4(b). If the direction of the tangent change largely, the osculating circle becomes small, and the curvature, the reciprocal of the radius, becomes large.



(a) (b)

Figure 2-4 Elements used for definition of curvature. (a) Derivative of tangent vector. (b) Radius R of osculating circle.

For the space curve, binormal vector $\boldsymbol{B}(s)$ is defined as the outer product of the tangent vector $\boldsymbol{T}(s)$ and the normal vector $\boldsymbol{N}(s)$. The torsion $\tau(s)$ measures the rotation of the normal vector $\boldsymbol{N}(s)$ and the binormal vector $\boldsymbol{B}(s)$ around the tangent vector. By Frenet–Serret formula, the vectors $\boldsymbol{T}(s)$, $\boldsymbol{N}(s)$, $\boldsymbol{B}(s)$, curvature $k(s)$ and torsion $\tau(s)$ are integrated as,

$$\begin{bmatrix} \boldsymbol{T}'(s) \\ \boldsymbol{N}'(s) \\ \boldsymbol{B}'(s) \end{bmatrix} = \begin{bmatrix} 0 & k(s) & 0 \\ -k(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{T}(s) \\ \boldsymbol{N}(s) \\ \boldsymbol{B}(s) \end{bmatrix}. \tag{2.2}$$

For a planer curve, tangent vector $\boldsymbol{T}(s)$, the normal vector $\boldsymbol{N}(s)$, and the curvature $k(s)$ are calculated with the same equations. Binormal vector $\boldsymbol{B}(s)$ and the torsion $\tau(s)$ are not defined in the 2D space.

Mathematically, the definition of curve includes a line as its special case having a straight shape, but usually, we note them separately as the curve being the curved shape and the line in the straight shape.

2.1.2. Surfaces

The surfaces represent the areas on the paper surrounded by either the creases or the edges of the paper. It may be expressed as a 2D manifold in 3D space, parametrized by (u, v), as shown in Figure 2-5.

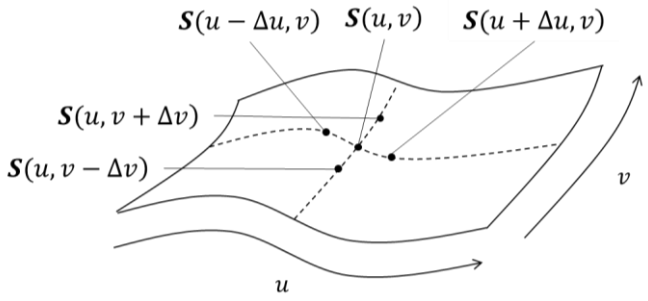
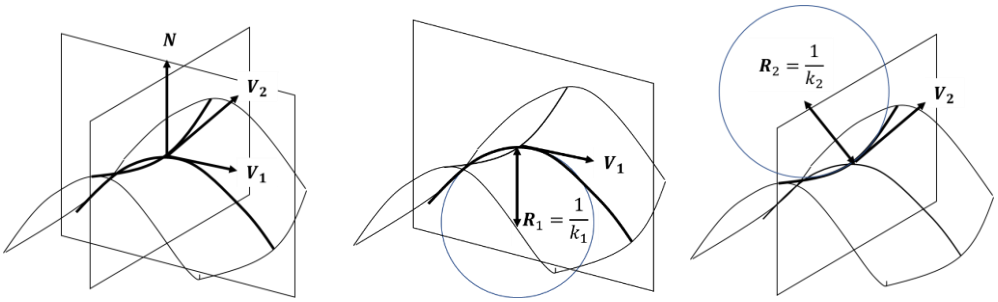


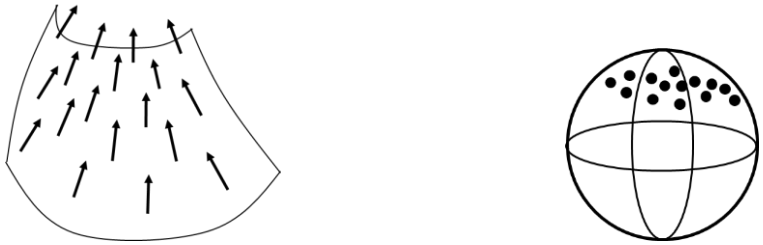
Figure 2-5 Surface parametrized by (u, v).

The shape of the surface may be characterized by the two *principal curvatures* k_1, k_2, which are the maximum and the minimum curvatures of a point on the surface. Their directions, referred to as the *principal directions*, are orthogonal to each other, as shown in Figure 2-6. The surface may be categorized by the Gaussian curvature K, the product of two principal curvatures, and mean curvature H, the average of the two principal curvatures. Another description of the surface characteristic is the Gauss map, or Gaussian image, shown in Figure 2-7. It is the normal vectors on the surface plotted on the sphere, showing the distribution of the surface orientation. The shape of the surface is expressed by the normal distribution, such as in flat shape, bending in one direction, convex shape in all direction, etc.



(a) (b) (c)

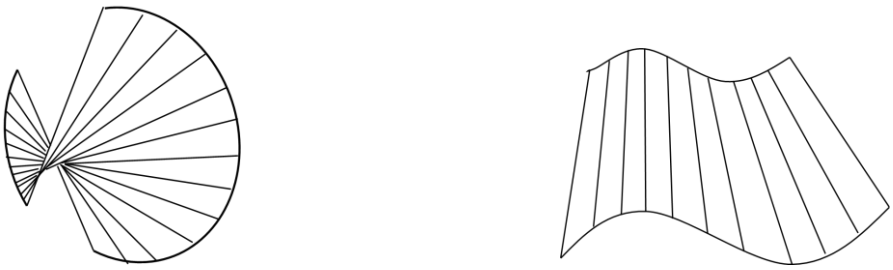
Figure 2-6 Principal curvatures k_1, k_2 as reciprocal of the radius of osculating circle R_1, R_2 and principal directions $\boldsymbol{V}_1$, $\boldsymbol{V}_2$.



(a) (b)

Figure 2-7 Gauss map. (a) Surface with normal vectors. (b) Gauss map of surface.

The paper or a thin plate composing a curved folding is known to be a *developable surface*. It is a subset of *ruled surface*, which is a smooth surface made as a trajectory of a line, and every point on the surface is on a straight line, called a *ruling*, that lies on the surface (Figure 2-8(a)). A developable surface also contains rulings, with every point on the surface having zero Gaussian curvature (Figure 2-8(b)). In other words, the surface may be flattened on to a plane without stretching or tearing. On a developable surface, the rulings lie on the surface in the direction perpendicular to the surface bending. It is the principal direction with the minimum curvature, which equals to zero in case of zero Gaussian curvature. The rulings never intersect on the surface except on the edge of the rulings, such as an apex of a cone. On the ruling, the normal vectors are identical on any point of a ruling. So, the Gauss map of a developable surface would be a thin curve since the normal vectors locally distribute in one direction.



(a) (b)

Figure 2-8 Ruled surfaces. (a) Ruled surface which is not developable. (b) Developable surface.

2.1.3. Continuity of curves and surfaces

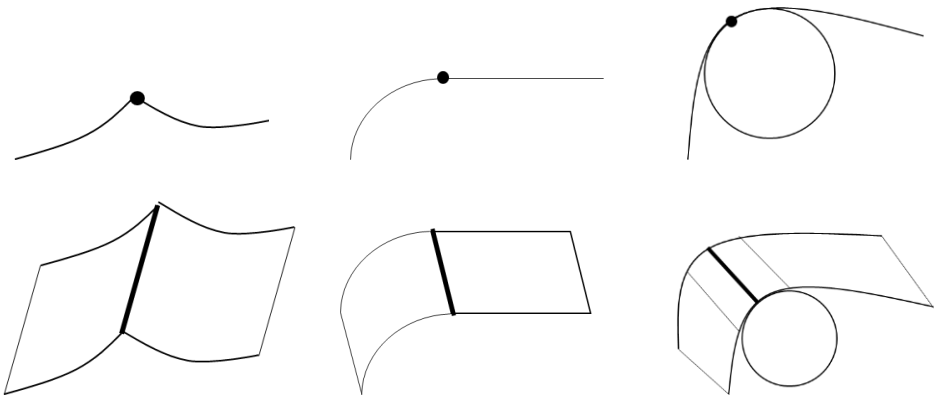
The smoothness of curves and surfaces are described by the *geometric continuity*, as listed below, with the description about the differentiability with respect to arc length, and shown in Figure 2-9.

G0: The curves/surfaces touch at the joint point / common edge, where it may not be differentiable.

G1: The curves/surfaces also share a common tangent direction at the joint point / common edge. It is differentiable on the point/edge, but not second-order differentiable.

G2: The curves/surfaces also share a common center of curvature at the joint point / common edge. It is second-order differentiable on the point/edge.

In describing the continuity of the parametric function, there is another measurement denoted as the *parametric continuity*, or C0, C1, C2 continuity, whose differentiability is discussed with respect to the parameter, such as time and other variables not always proportional to the arc length. It describes the dynamic behavior of a point on a trajectory rather than the curve shape itself. In our work, we focus on the geometric continuity as we are interested in the shape of the curve or surface. Since our methods use curvatures and torsion to calculate the rulings, as later explained in Section 2.3, the crease curve must be G2 continuous everywhere.



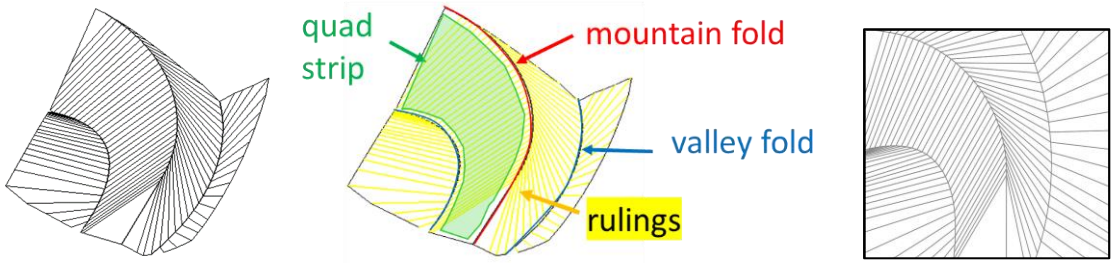
(a) (b) (c)

Figure 2-9 Continuity of curves and surfaces. Top row: curves. Bottom row: surfaces. (a) G0 continuity. (b) G1 continuity. (c) G2 continuity.

2.2. Modeling of developable surface in discrete form

There are many types of models which represent a sheet of paper, and some of them are introduced in Chapter 3, related works. In our research, we use the ruling-based polygon model. Figure 2-10(a), (b) shows a general 3D polygon model representing the curved folded paper by discretizing the continuous surfaces. The curved surfaces are modeled as the quad strips adjacent to the curved crease with the rulings being the shared edges between the quads. A curved crease, which is a continuous curve on a real

paper, is approximated by a sequence of edges shared by two quad strips, connecting the vertices on them. Thus, the rulings are the edges starting at the vertices on the curve and lie on the curved surface in the directions perpendicular to the directions of the paper bending. As this polygon model is based on the rulings, its geometry must be updated when the paper is bent in different directions, which naturally occurs in folding a piece of paper along a curved crease. Figure 2-10(c) shows the creases and the rulings mapped onto the flattened 2D paper, which is used to illustrate the crease pattern and the 2D configuration of the rulings.



(a) (b) (c)

Figure 2-10 Modelling of curved folding. (a) 3D polygon model. (b) Description of elements in 3D polygon model. (c) Creases and rulings mapped onto 2D space.

On the discrete polygon model, Gaussian curvature on a vertex is calculated as,

$$K = 2\pi - \sum \alpha_i. \qquad (2.3)$$

It is a deficiency of total corner angles adjacent to the vertex, shown in Figure 2-11, to 2π. If $K = 0$, the local area around the vertex is developable, as it can be flattened on a plane. If $K < 0$, or the total corner angles are larger than 2π, two principal curvatures have different signs and the local shape is categorized as a saddle shape. Conversely, if $K > 0$ and the total corner angle is smaller than 2π, two principal curvatures have identical signs and the local shape would be a cap shape.

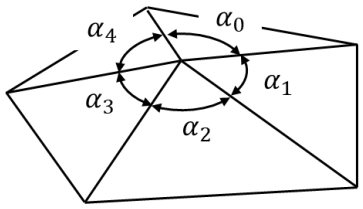
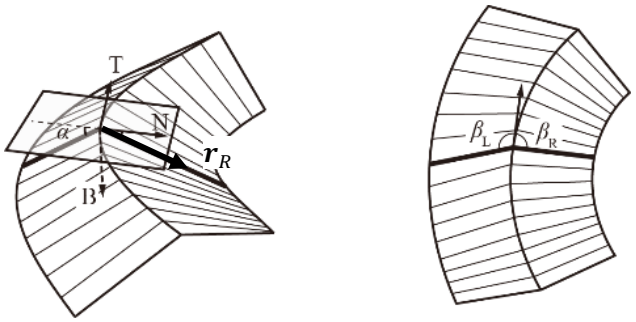


Figure 2-11 Total corner angles adjacent to vertex.

2.3. The calculation of the curved folding

In this section, the geometry and the calculation method of a curved crease and the adjacent curved surfaces are explained. It follows the equations introduced by Fuchs and Tabachnikov [Fuchs_99], which are also organized and interpreted by Tachi in his work of building rigid foldable curved folding [Tachi_11b]. As an outline, the equations explain the relations between (i) the shape of crease curve in 3D space, (ii) the shape of the crease curve mapped onto the 2D space, and (iii) the folding angle. Then the directions of the rulings mapped on the 2D space are calculated. Finally, the rulings in 3D space are derived using the rulings in 2D space, the direction of the curved crease in 3D space, and the folding angle which defines the orientation of the curved surface adjacent to the curved crease. The quad strips composing the discretized 3D polygon model of the curved folded surface is obtained as the quads having two rulings as their edges. The relationship and the mathematical framework are also described thoroughly in [Demine_14].

The definitions of the vectors on the curve and the angles which define the rulings are shown in Figure 2-12, with the tangent vector $\boldsymbol{T}(s)$, normal vector $\boldsymbol{N}(s)$, and binormal vector $\boldsymbol{B}(s)$ on the 3D crease curve $\boldsymbol{X}(s)$. The folding angle $\alpha(s)$ is the angle between the surface and the osculating plane projected to the normal plane of the curve, being always the same for the left and right side. The rulings angles $\beta_L(s)$ and $\beta_R(s)$ are the angles between the tangent vector and the rulings on the left and right side in the 2D mapped space.



(a) (b)

Figure 2-12: Definitions of vectors and angles. (a) 3D model. (b) Crease curve with rulings mapped to 2D space.

2.3.1. Equations

Given a space curve $\boldsymbol{X}(s)$ parameterized by the arc length s, its tangent vector $\boldsymbol{T}(s)$, normal vector $\boldsymbol{N}(s)$, and binormal vector $\boldsymbol{B}(s)$ are given by the equation (2.1). For the intuitive understandings, we use the coordinate system in which the binormal vector $\boldsymbol{B}(s)$ always orient to the backside of the paper. Then, looking in the direction of the tangent vector $\boldsymbol{T}(s)$, the normal vector $\boldsymbol{N}(s)$ always orients to the right side of the crease curve and the curvature $k(s)$ takes a positive or negative value depending on whether the curve is clockwise or counterclockwise looking from the top. The ideas of

such orientation fixed vectors and the signed curvature are also described in [Demine_14] as the *top-side Frenet frame*. By the Frenet–Serret formula, shown as equation (2.2), and the fact that the vectors in the Frenet frame are normal vectors, the curvature and the torsion are derived as,

$$k(s) = |\boldsymbol{T}'(s)|, \tag{2.4}$$

$$\tau(s) = |\boldsymbol{N}'(s) + k(s)\boldsymbol{T}(s)| = |-\boldsymbol{B}'(s)|. \tag{2.5}$$

Similarly, the curvature in 2D is derived as

$$k_{2D}(s) = |\boldsymbol{T}_{2D}'(s)|, \tag{2.6}$$

where $\boldsymbol{T}_{2D}(s)$ is the tangent vector of the crease curve mapped to the 2D space.

By [Fuchs_99], the relation between the folding angle, the 2D curve, and the 3D curve is

$$k_{2D}(s) = k(s)\cos\alpha(s). \tag{2.7}$$

The geometric relations between the curve shape defined by the curvature and the torsion, the folding angle, and the rulings mapped to the 2D space are

$$\cot\beta_L(s) = \frac{-\alpha(s)' + \tau(s)}{k(s)\ \sin\alpha(s)'}, \tag{2.8}$$

$$\cot\beta_R(s) = \frac{\alpha(s)' + \tau(s)}{k(s)\sin\alpha(s)}. \tag{2.9}$$

Note that the signs of some parameters in equations (2.8) and (2.9) are different from those of Tachi and Fuchs due to the difference in the vector orientations. At last, the rulings in 3D space are calculated as follows,

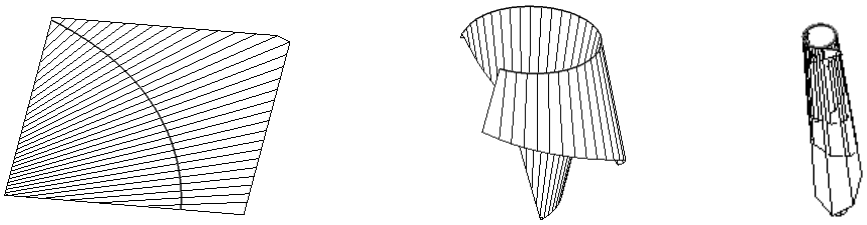
$$\boldsymbol{r}_L = \cos\beta_L\,\boldsymbol{T} - \sin\beta_L\cos\alpha\boldsymbol{N} + \sin\beta_L\sin\alpha\,\boldsymbol{B}\,, \tag{2.10}$$

$$\boldsymbol{r}_R = \cos\beta_R\,\boldsymbol{T} + \sin\beta_R\cos\alpha\boldsymbol{N} + \sin\beta_R\sin\alpha\,\boldsymbol{B}\,. \tag{2.11}$$

By these equations, the ruling directions on the 2D paper are derived from the parameters of the curve and the folding angle. Then, the rulings in 3D space are calculated from the rulings in 2D, folding angle, and the Frenet frame. However, as each ruling is calculated separately from the parameters on the local point, there may be some conflicts between the rulings, such as the rulings crossing on the 2D paper. This cannot occur on the real paper but may happen on the calculation if the inputs are not controlled properly. The method to resolve the rulings crossing and to restrict the user input to avoid such cases is one of the important issues of our research. There is also a problem of self-collision. As the equations do not take care of the collision avoidance, some faces may happen to pass through another face by the calculation. In our system, the user needs to check the shape visually and control the input to avoid the collisions.

2.3.2. Interpretations of the equations

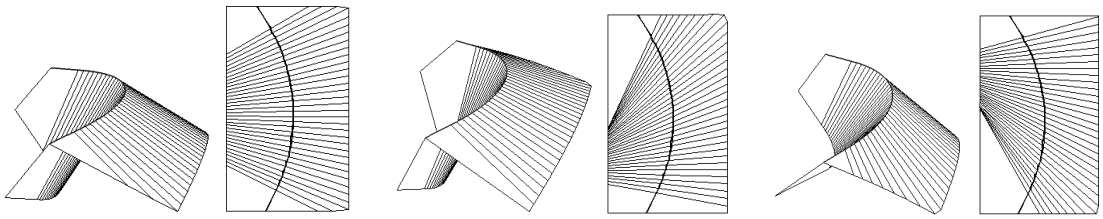
Equation (2.7) shows the relation between the curvature $k(s)$, the curvature in 2D space $k_{2D}(s)$, and the folding angle $\alpha(s)$. As the folding angle $\alpha(s)$ increases from 0 to $\pi/2$, $\cos\alpha(s)$ decreases from 1 to 0, and the curvature of the 3D curve $k(s)$ becomes larger relative to the curvature in 2D space $k_{2D}(s)$. Figure 2-13 shows some 3D models of the curved folded surfaces generated from the same 2D crease and folded in different folding angles. In the flat state, or $\alpha(s) = 0$, the curvature in 3D and 2D coincides (Figure 2-13 (a)). As the folding angle increases, the folded paper curls and the 3D curvature increases (Figure 2-13 (b)). By folding the crease as much as possible to nearly $\pi/2$, the paper has to be rolled tightly, making the curvature of the 3D curve to approach infinity (Figure 2-13 (c)). Note that the curved crease cannot be folded completely to $\alpha(s) = \pi/2$.



(a) (b) (c)

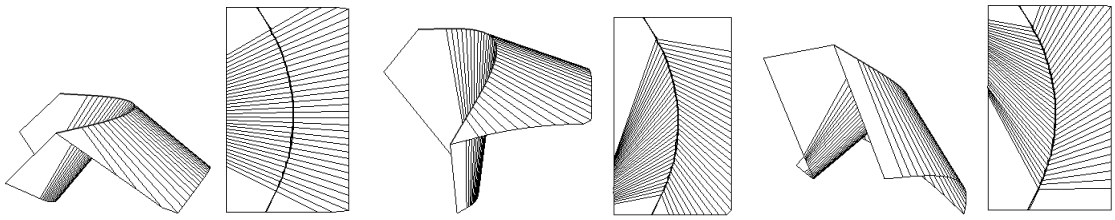
Figure 2-13 Curved folded surface in various folding angles. (a) $\alpha(s) = 0,\ k(s) = k_{2D}(s)$. (b) $0 < \alpha(s) < \pi/2,\ k(s) > k_{2D}(s)$. (c) $\alpha(s) \to \pi/2,\ k(s) \to \infty$.

By equations (2.8) and (2.9), we can derive the ruling angles $\beta_L(s)$, $\beta_R(s)$ from the curvature $k(s)$, the torsion $\tau(s)$, and the folding angle $\alpha(s)$. Figure 2-14 shows the rulings in different folding angles. As the derivative of the folding angle $\alpha(s)'$ is in the numerators of the equations (2.8) and (2.9) with different signs, the increase or the decrease of the folding angle along the crease curve affects the rulings on the left and right sides in the opposite directions. Figure 2-15 shows the rulings in different torsions. As opposed to the derivative of the folding angles $\alpha(s)'$, the torsion $\tau(s)$ is in the numerators of the two equations with the same sign. So, it affects the rulings on the both sides of the crease in the same direction. By controlling the folding angles and the torsion, the rulings in 2D mapped space may be generated in various configurations. However, since the equations (2.8) and (2.9) has the curvature $k(s)$ and $\sin\alpha(s)$ in the denominator, the rulings are not calculated if the curvature $k(s)$ is zero or the folding angle $\alpha(s)$ is zero. Zero curvature corresponds to the crease curve having an inflection point or being a straight line. Zero folding angle is the paper in a flat unfolded state.



(a) (b) (c)

Figure 2-14 Rulings in various folding angles. (a) Constant folding angle, $\alpha'(s) = 0$. (b) Increasing folding angle, $\alpha'(s) > 0$. (c) Decreasing folding angle, $\alpha'(s) < 0$.



(d) (e) (f)

Figure 2-15 Rulings in various torsions. (a) Planer curve, $\tau(s) = 0$. (b) Positive torsion, $\tau(s) > 0$. (c) Negative torsion, $\tau(s) < 0$.

From the equation (2.7), we could roughly say that given two of the three elements, 3D curve, 2D curve, and folding angle, the other element is derived from the two elements. In other words, there are these three cases,

Case A: The 2D curve is derived from the 3D curve and the folding angle.

Case B: The 3D curve is derived from the 2D curve, the folding angle, and the torsion.

Case C: The folding angle is derived from the 2D and 3D curves.

It is summarized in the equation since

(i) The 3D space curve is defined by the curvature $k(s)$ and the torsion $\tau(s)$, and

(ii) The 2D curve is defined by the 2D curvature $k_{2D}(s)$.

In case B, the torsion $\tau(s)$ is also needed as it is not given by the equation (2.7). Then, by equation (2.8) and (2.9), the ruling directions in 2D space, $\beta_L(s)$ and $\beta_R(s)$, are derived and, by equation (2.10) and (2.11), the 3D ruling directions are obtained using vectors $\boldsymbol{T}(s)$, $\boldsymbol{N}(s)$, and $\boldsymbol{B}(s)$ derived from the 3D space curve.

Chapter 3.
Related Work

Curved folding is a well-studied research area in math, engineering, arts, etc. It is emphasized in computer-aided design and manufacturing because the property of developable surfaces, made by bending and folding of a thin plate without stretching, is preferable in manufacturing. In computer graphics, modeling method of a developable surface is studied to design and to create an animation of paper and other thin materials. In the following subsections, research about the developable surface and the curved folding are introduced.

3.1. Modeling of developable surfaces and curved folding

3.1.1. Modeling of developable surface

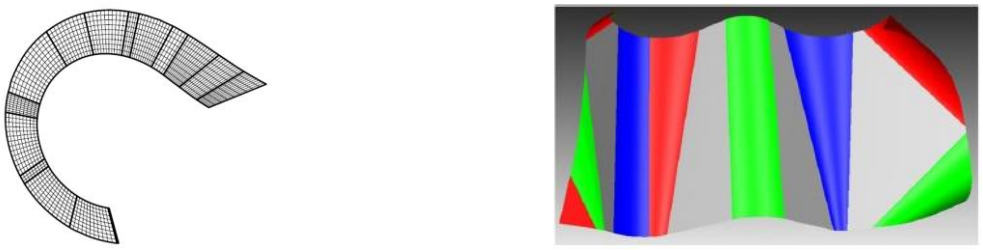
The developable surface may be modeled in the computers as a continuous surface or a discrete polygon model. One of the modeling methods is continuously aligning pieces of developable primitives such as cones, cylinders, or planes (Figure 3-1). A cone spline surface is composed of pieces of right circular cones in different heights and base radii [Leopoldseder_03], [Chen_04]. Hwang et al. introduced an interactive modeling method of developable surfaces by wrapping a planar figure around cones and cylinders [Hwang_15].

Another modeling method of a continuous developable surface is constrained parametric surfaces (Figure 3-2). Chu et al. modeled surfaces as quadratic or cubic developable Bézier patches, defined by two Bézier boundary curves constrained by the surface developability [Chu_02]. Similarly, the interactive method proposed by Tang et al. models developable surface as a pair of spline curves, and optimizes by the constraint solver to meet the condition of developability, though in this case the final surface is expressed in the form of ruling based polygon model [Tang_16].

As a discrete model, developable surfaces are modeled as equally divided mesh or nets, or a polygon model based on the rulings (Figure 3-3). Burgoon et al. proposed a discrete shells model, where the paper is initially modeled as equally divided triangular meshes, and as it is being folded, new edges and vertices are added [Burgoon_06]. Rabinovich et al. modeled the developable surface by DOG (Discrete orthogonal geodesic) Nets [Rabinovich_18a, b]. To ensure that the surface modeled in the discrete method is developable, one could check that the Gauss curvature of every quad is small according to the discrete surface theory proposed by [Hoffmann_16].

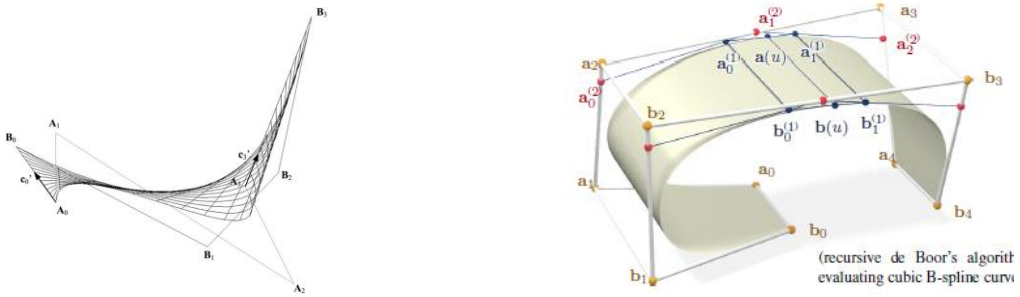
For ruling based polygon model, Bo et al. proposed an interactive modeling method of developable surface or its bending motions through a geodesic curve controlled by the user [Bo_07]. The surface is modeled as an envelope of the rulings generated by the curve. Instead of giving the curve shape directly, the method of Solomon et al. allows the user to prescribe the 3D positions of points and lines defined in 2D space [Solomon_12]. Then the developable surface is refined by relaxing the rough polygon

model to the curved configuration that minimizes the nonlinear elastic thin plate energy. In the approximation of developable surfaces from given data, a discrete model is useful. Kilian et al. approximate nearly developable 3D shape data with curved creases by patches of discrete developable surfaces expressed as PQ strips with the rulings being the edges [Kilian_08] (Figure 3-4). Perriallat et al. also modeled the developable surface approximated from sparse 3D points obtained from images by discrete polygons divided by the estimated rulings [Perriallat_13]. In our research, we also use the ruling based polygon model, as described in Section 2.2.



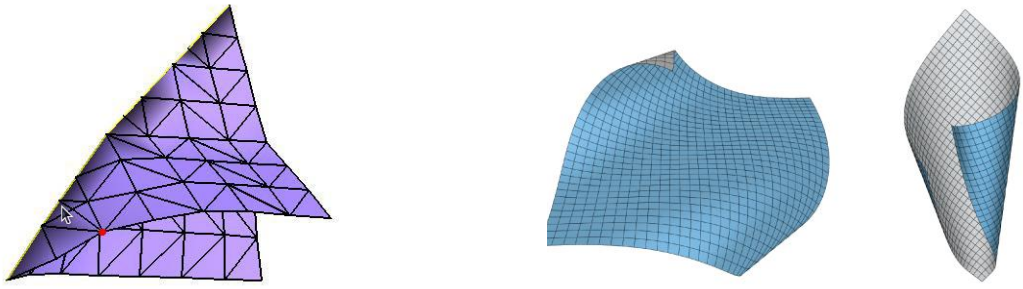
(a) (b)

Figure 3-1 Developable surface composed of developable primitives. (a) Cone spline surface [Chen_04]. (b) Mappings of different cones and cylinders [Hwang_15].



(a) (b)

Figure 3-2 Developable surface in the form of parametric surface. (a) Cubic developable Bézier patches [Chu_02]. (b) Developable surface defined by spline curves [Tang_16].



(a) (b)

Figure 3-3 Developable surface as a net-based polygon model. (a) Discrete shell origami [Burgoon_06]. (b) DOG nets [Rabinovich_18a].

Figure 3-4 Developable surface as a ruling based polygon model [Kilian_08].

3.1.2. Approximation by developable surface

There is a lot of research on approximating 3D data by developable surfaces because many products are manufactured from flat thin materials and are composed of developable shapes. The input 3D data may be point clouds, 3D points obtained by stereo images, 3D mesh, etc. In approximation of point cloud with one sheet or strip of developable surface, the Gaussian image, or the surface normal vectors plotted on a sphere, is often used in the initial stage to estimate the type of surface and the bending directions [Chen_04], [Peternell_04a,b], followed by the fitting of the recognized types of surface, such as cones, cylinders, or other developable surface. Such method assumes the point cloud is dense enough so that the surface normal of a point could be calculated from the neighboring points. To take as input the sparse 3D points obtained from images, the method of Perriallat et al. first defines the boundary of the surface and selects possible rulings from many candidate lines connecting pairs of sample points on the boundary, according to the 3D point distribution and the constraint that the rulings does not cross each other [Perriallat_13]. The surface is modeled as a discrete developable surface composed of planar polygons adjacent via the rulings. In processing more complicated 3D shape, such as developable surface with creases, Kilian et al. reconstruct a 3D polygon mesh of a developable surface from an input 3D data, such as point clouds or 3D mesh, by initially fitting planar polygons on the 3D shape and then optimizing their shapes and orientations for its developability and the connectivity [Kilian_08].

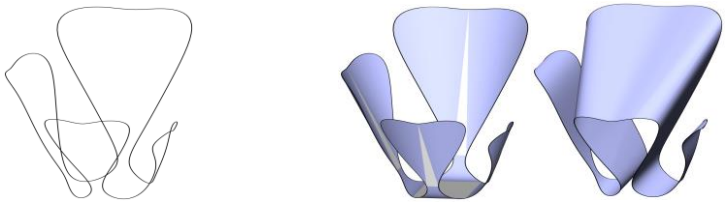
Other than above research which takes nearly developable 3D data as an input, there are many methods to model object surfaces as pieces of developable surfaces. Liu et al. propose a method to construct conical mesh, or planer quad mesh useful in architecture, to express curved surfaces by optimization [Liu_06]. Stein et al. takes 3D mesh data as an input and re-meshes it to be composed of some developable surfaces [Stein_18] (Figure 3-5). There are also many papers on making developable strips from 3D mesh data, for applications such as paper crafts [Mitani_04,06] and other manufacturing products [Liu_07], [Gonzalez-Quintial_15] (Figure 3-6). There is also another approach for making a developable surface for product design, taking boundary curves as input [Rose_07], [Wang_08] (Figure 3-7).

(a) (b)

Figure 3-5 3D mesh data composed of developable surfaces [Stein_18]. (a) Original 3D data. (b) Developable polygon data.

(a) (b)

Figure 3-6 3D mesh data composed of developable strips [Liu_07]. (a) Triangulation of model. (b) Triangulation flattening.



(a) (b)

Figure 3-7 Developable surface made from boundary curves [Rose_07]. (a) Input boundary curves. (b) Two possible developable surfaces.

3.1.3. The design method of curved folding

Though there is a lot of research on developable surfaces, there are not as many papers on how to design a new curved folding. Sternberg introduced an idea of folding a piece of paper into the shape of *voltex* by the curved creases and discussed the tessellation of curved folding [Sternberg_09]. Mitani proposed a system to design curved folding with rotational sweep [Mitani_09], followed by the method of folding a curved surface on a planer curve and reflecting [Mitani_11]. To design more generalized shapes, Tang et al. proposed an interactive method where the user controls the 3D positions of some vertices, and the shapes are optimized by the constraint solver to meet the condition of developability [Tang_16].

Apart from curved folding strictly defined on completely developable surface, Zhu et al. proposed *soft folding*, where the user inputs the shape of the curved crease in 2D, its fold magnitude, and the sharpness [Zhu_13] (Figure 3-8). A thin plate is folded or bent according to the input and forms 3D shapes in nearly developable surfaces, such as products made of thick cloth. Another work of designing the folds along a curve is the

flat foldable structure proposed by Tachi [Tachi_11a,13]. The shape of the structure is designed according to the shape of the 3D folding curve. However, the structure is not a curved folding because its folds are composed of two sheets joined together along a curve in the flat folded state and could not be flattened by unfolding.

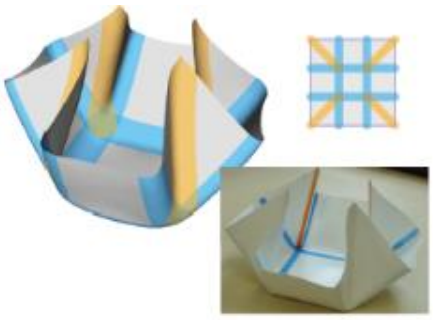


Figure 3-8 Soft folding [Zhu_13].

3.2. Simulation of a folding motion

In the simulation of folding motion, the shape of the paper is modeled in every time step during the motion, keeping the continuity between adjacent time steps. For such use, the developable surfaces are modeled as a discrete polygon model, with fixed or dynamic mesh configuration, as introduced in the following subsections.

3.2.1. Simulation with fixed mesh data

In the field of paper folding simulation, Tachi developed software to simulate the folding motion of the paper using the rigid folding method [Tachi_09]. 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 a design method for flat-foldable vault structures composed of flat-foldable tubes assembled by welding two sheets [Tachi_11a,13] (Figure 3-9). Although the structure is folded along the curve, the folding angles are restricted to be constant throughout the curve, which ensures the rulings to be fixed so that the structure could be modeled with the rigid folding method. It does not support any deformation where the ruling directions on the surface change during the folding motion. Ghassaei et al. implemented a fast and interactive web application for origami simulation, which works on GPU [Ghassaei_18]. Their model is based on the constraints on edge length, folding angle, and the angle between the edges, 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 but does not consider the developability nor the ruling transition in the intermediate state.

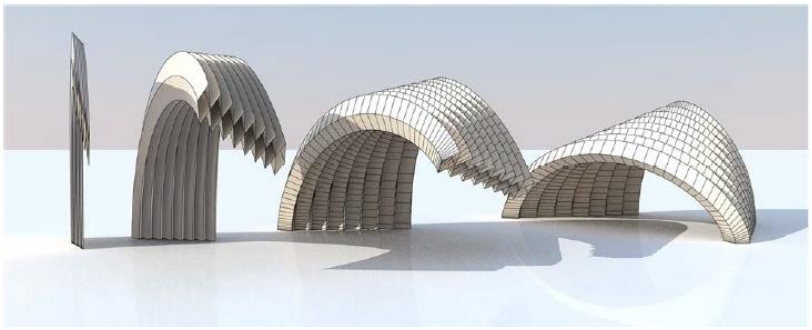


Figure 3-9 Flat foldable vault structure [Tachi_13].

3.2.2. Dynamic simulation of developable surface and curved folding

In recent research, dynamic simulation methods of the developable surface have been proposed. Narain et al. proposed a method to simulate a sheet of thin materials by adaptive mesh refinement, which could model irregular deformations such as crumpled paper [Narain_13]. As to the folding motion of curved folding, Kilian et al. also take an approach of re-meshing the triangular mesh according to the local surface curvatures in modeling the 3D shape of a paper in string actuated folding motion [Kilian_18] (Figure 3-10). Their methods are successful in representing various shapes, but resulting mesh data may contain thousands of faces, not necessarily suitable for observation of the rulings. Rabinovich et al. introduced discrete orthogonal geodesic nets, which approximates a smooth developable surface by optimizing the corner angles of the quad mesh to be orthogonal [Rabinovich_18b] (Figure 3-11). In their methods, a curved crease is modeled as the intersecting curve of the two DOF nets. Their methods are successful in representing the shape of curved folded paper, but different from our research goal of providing a design system that the user can control the paper shape with the visualization of the rulings.

Figure 3-10 String actuated folding motion [Kilian_18].

Figure 3-11 Curved folding by DOG nets [Rabinovich_18b].

3.3. Analysis of curved folding

Analysis of curved folding and the behavior of the rulings have been studied since before the computer graphics technologies have developed to be able to render or simulate 3D shapes interactively.

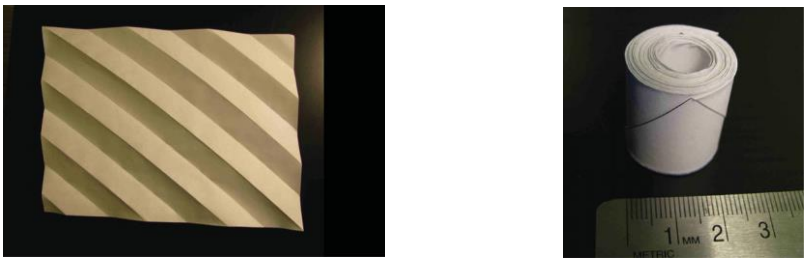
David Huffman had studied about curved folding from both perspectives of math and art. He had analyzed the local mathematical behavior of folding developable surface along curved creases [Huffman_76]. Fuchs and Tabachnikov introduced the mathematical relation between 2D and 3D crease curve, folding angle, and the directions of the rulings on the curved folding [Fuchs_99]. Their equations are interpreted and explained with examples of resulting crease curve shapes with different parameters to be used for the design of rigid folding of curved crease [Tachi_11b]. On the other hand, Huffman's artworks on curved folding have been analyzed by Demine et al. with the cooperation of his family after his death [Demaine_10a,14,18] (Figure 3-12). In their work, the rulings, which are not depicted in the original crease pattern, were estimated and each curve on the crease pattern was categorized as circle, ellipse, or parabolas. Their research on paper folding ranges widely, including the designing curved surface with straight pleated creases [Demaine_10b]. They proved that a straight crease could not be bent unless the paper is unfolded or folded completely, which also characterizes the difference between a curved crease and a straight crease. They have summarized the art, applications, and mathematics on curved folding in [Demaine_11]. Dias et al. developed recursion equations to describe a series of concentric curved folds, which are prescribed by a single fold [Dias_12].

Figure 3-12 Curved folding by David Huffman.

Duncan and Duncan studied the geometry of curved folding and proved their theory in two special cases, the 3D crease being a planar shape and constant folding angle [Duncan_82]. They had foreseen that such geometric rules would encourage the manufacturing using curved folding of sheet metal. The shape of such curved fold was also studied in the area of mechanical science as the behavior of *developable cone*, or a thin disk supported on the rim of a bowl and its center pushed down by a thin object [Farmer_05]. Later, their works were studied and verified by Seffen using Gauss's spherical image construction [Seffen_18]. Roeschel examined the case where two curved surfaces linked by a curved fold is both cylinders using the spherical image [Roeschel_16].

3.4. Applications

Some of the design implementation using the curved folding is summarized in Demaine et al., such as designs of a lamp, bench, column covers, and car design [Demaine_11]. Other applications described in a paper are listed below. Frey et al. proposed a method to model and to design developable surface for sheet metal stamping to form a shape without stretching the material [Frey_04]. As an application for deployable structures on small spacecraft, a technique is presented to wrap a flat sheet around a hub to minimize its size, by folding the sheet along the crease in a slightly curved shape, considering the thickness of the sheet [Lee_13] (Figure 3-13). Epps et al. found a company *Robofolds*, which manufactures curved folded metal sheets by robot hand [Epps_14] (Figure 3-14). Miyashita et al. propose self-assembling origami propellers, where the curved creases are folded by heat to form the 3D shape of propellers [Miyashita_15] (Figure 3-15). Nelson et al. describe a deployable compliant rolling-contact element joint (D-CORE joint) that employs curved-folding origami techniques to enable the transition from a flat to deployed state [Nelson_16] (Figure 3-16).



(a) (b)

Figure 3-13 Deployable structures folded by curved folds [Lee_13]. (a) Unfolded configuration. (b) Folded configuration.

Figure 3-14 RoboFolds [Epps_14].

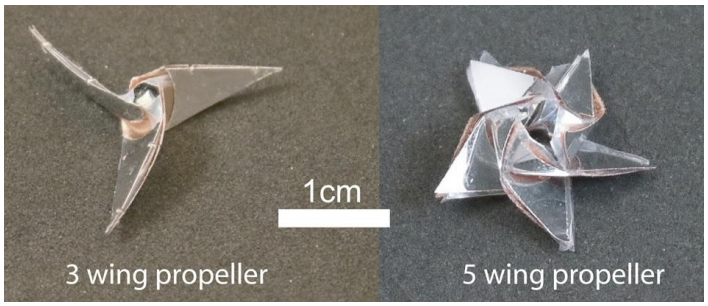


Figure 3-15 Self-assembling origami propellers [Miyashita_15].

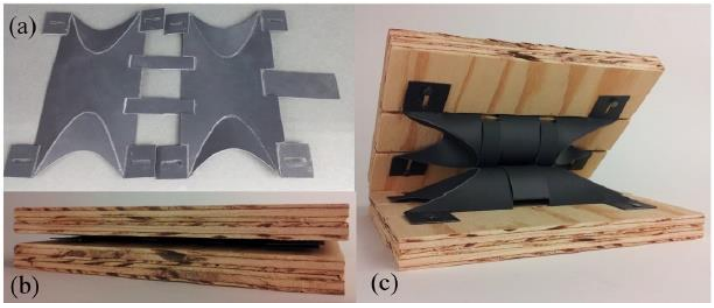


Figure 3-16 D-core joint made by curved folding [Nelson_16].

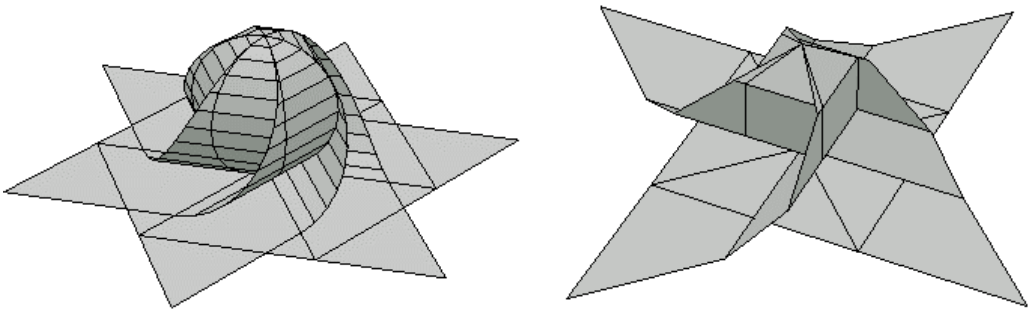
Chapter 4.
Simulation of curved folding with fixed rulings

As noted in Section 3.2, the paper shapes are generally modeled as a discrete polygon model in simulating the folding motion. Following the previous work, we will work with the discrete polygon model on this and the following chapters. This chapter explains the preliminary and verification experiments on curved folding simulation using a fixed ruling model. The fixed ruling model is a polygon model where the ruling configuration does not change during the folding motion. The rulings are kept in the original directions on the 2D paper while the folding angle is updated. The experiment was carried out to verify the necessity of the model supporting the ruling transition while the bending direction of the paper changes. As introduced in the related work, there is software to simulate the folding motion of a paper modeled as fixed mesh data. In the experiment, some example crease patterns are tested with the rigid folding method [Tachi_09] and the Origami simulator [Ghassaei_18]. The result shows that there are some cases that a paper model with curved creases cannot be folded by the rigid folding method, being locked because the folding motion, or the movement of the faces according to the folding angle, cannot be carried out without disjointing the shared edges between the other adjacent faces. We also verified one of our curved fold model with the Origami simulator and observed that the paper model contains some strain while being folded. On both software, the self-collision is not restricted, and there were some cases that a plane is passing through another plane.

4.1. Folding origami-sphere with the rigid folding

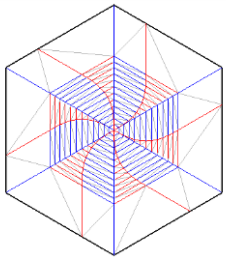
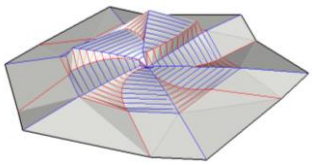
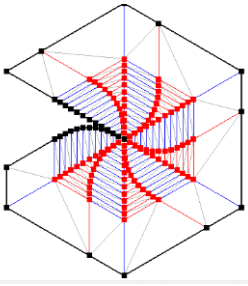
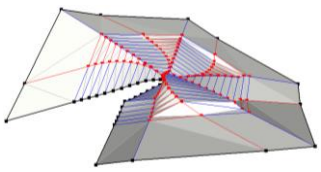
Rigid origami is a modeling method of a paper folding with flat rigid sheets joined by hinges. A well-known software targeting this type of model is the Rigid origami simulator, developed by Tachi [Tachi_09]. However, to carry out the preliminary experiment, we used the software implemented by Mitani. Given a set of creases and the faces, the software gradually increases or decreases the folding angles of all creases simultaneously, moving the vertices to the position consistent to the folding angles and by restricting the change in the edge length. The sample crease patterns are some variations of the *origami-sphere* as they are some good examples of curved crease containing both curved and straight creases. Some segments of the *origami-sphere* cut out from the original crease pattern are also used to eliminate the constraints on the lengths of some edges. The original crease patterns are designed using ori-revo [Mitani_09] and exported as 3D polygon models, as shown in Figure 4-1. The input crease patterns and the results of the simulation are shown in Figure 4-2. In most cases, the final shape of the polygon model converged to nearly flat shape, far from being a sphere shape as in Figure 4-1. This is considered to be caused by the confliction between the vertex positions derived from the folding angle and the length of the edges.

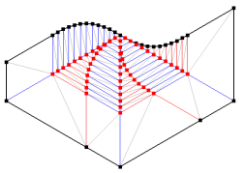
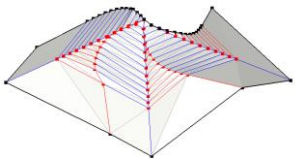
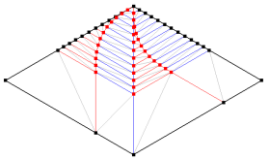
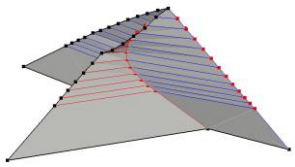
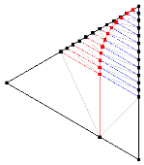
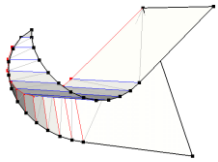
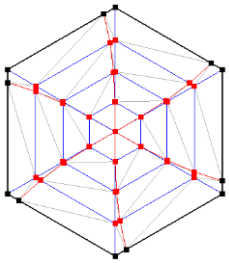
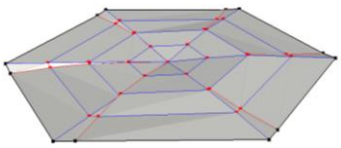
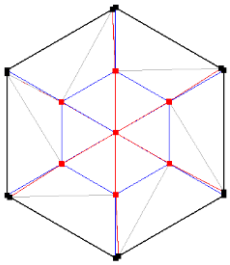
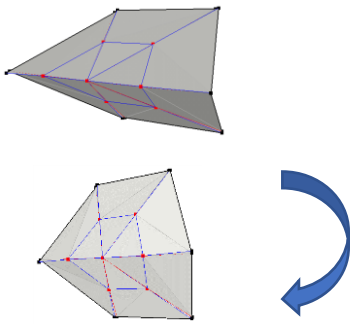
By optimizing the folding angle to meet all the constraints, the folding angle is not able to increase and the folding motion stops. If a part of the crease pattern is cut out in the radial direction, the straight creases seem to be folded more easily than the curved creases. The curved crease is folded only if there are no other creases. At first, this result seemed a bit strange because the *origami-sphere* is actually foldable using real paper. Then observing the real paper being folded, we noticed that the bending directions of the curved surfaces are changing gradually during the folding motion and that we need to consider about the transition of the ruling directions.

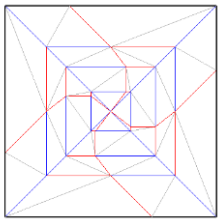
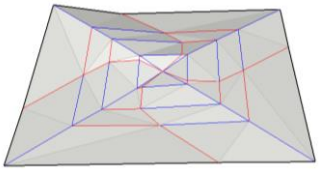
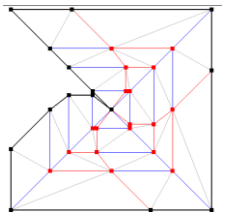
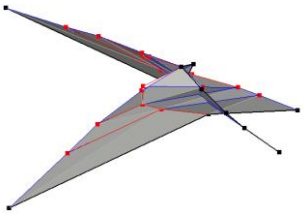
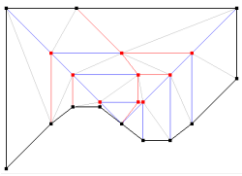
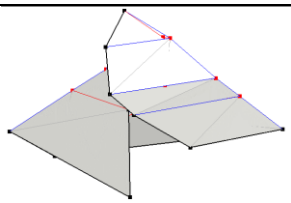
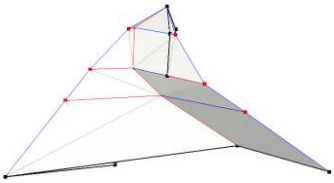
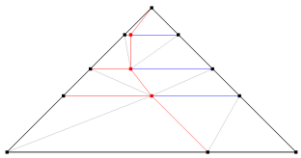
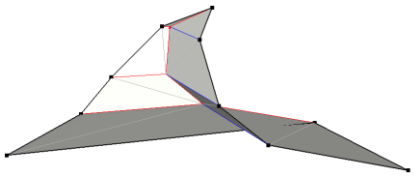
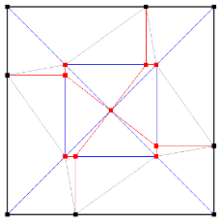
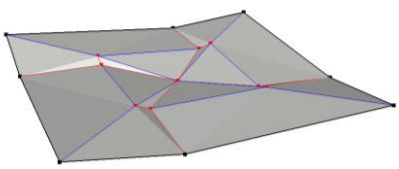


(a) (b)

Figure 4-1 3D models of *origami-spheres* designed by *ori-revo*. (a) Sphere with six flaps, each quad strip having 12 rulings. (b) Sphere with four flaps, quad strip having three rulings.

No.	Input crease pattern	Result
1	 Figure 4-1(a), whole crease pattern	 Not folded.
2	 Figure 4-1(a), one surface cut off.	 Not folded.

3	 Figure 4-1(a), 1/2 cut out by the curved creases.	 Not folded.
4	 Figure 4-1(a), 1/3 cut out by the straight creases.	 The strait crease is folded. The curved creases are not folded.
5	 Figure 4-1(a), 1/6 cut out by the straight creases.	 The curved crease is folded.
6	 Figure 4-1(a), center area with three rulings.	 Not folded.
7	 Figure 4-1(a), the center area with one ruling.	 Two straight creases are folded and then unfolded.

8	 Figure 4-1(b), whole crease pattern.	 Not folded.
9	 Figure 4-1(b), one surface cut off.	 The straight creases are folded with self-collision. The curved creases are not folded.
10	 Figure 4-1(b), 1/2 cut out by the curved creases.	  The straight creases are folded. The curved crease is folded incompletely
11	 Figure 4-1(b), 1/4 cut out by the straight creases.	 The curved crease is folded.
12	 Figure 4-1(b), the center area with one ruling	 Not folded.

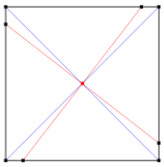
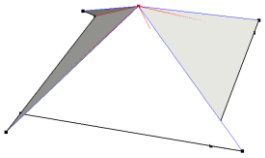
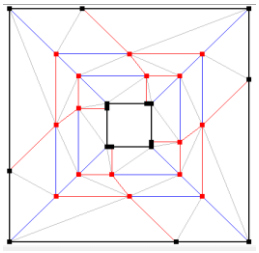
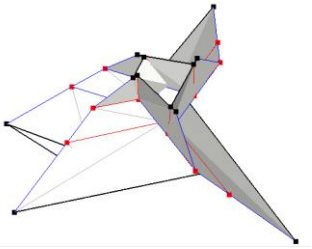
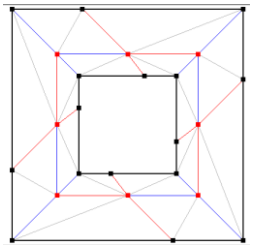
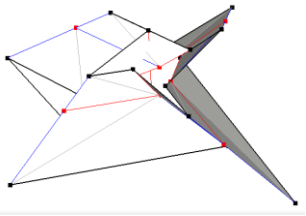
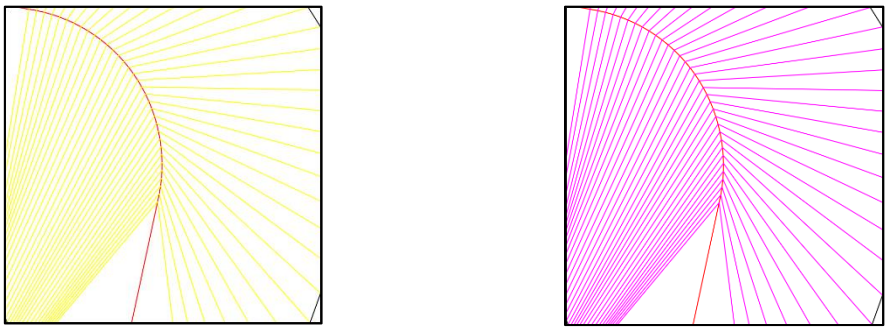
13	 Figure 4-1(b), the center area with no ruling	 Folded.
14	 Figure 4-1(b), the center area cut off, and the outer area with two rulings remains.	 Folded but not in the shape of a sphere.
15	 Figure 4-1(b), the center area cut off, and the outer area with one ruling remains.	 Folded but not in the shape of a sphere.

Figure 4-2 The result of the experiment of rigid folding.

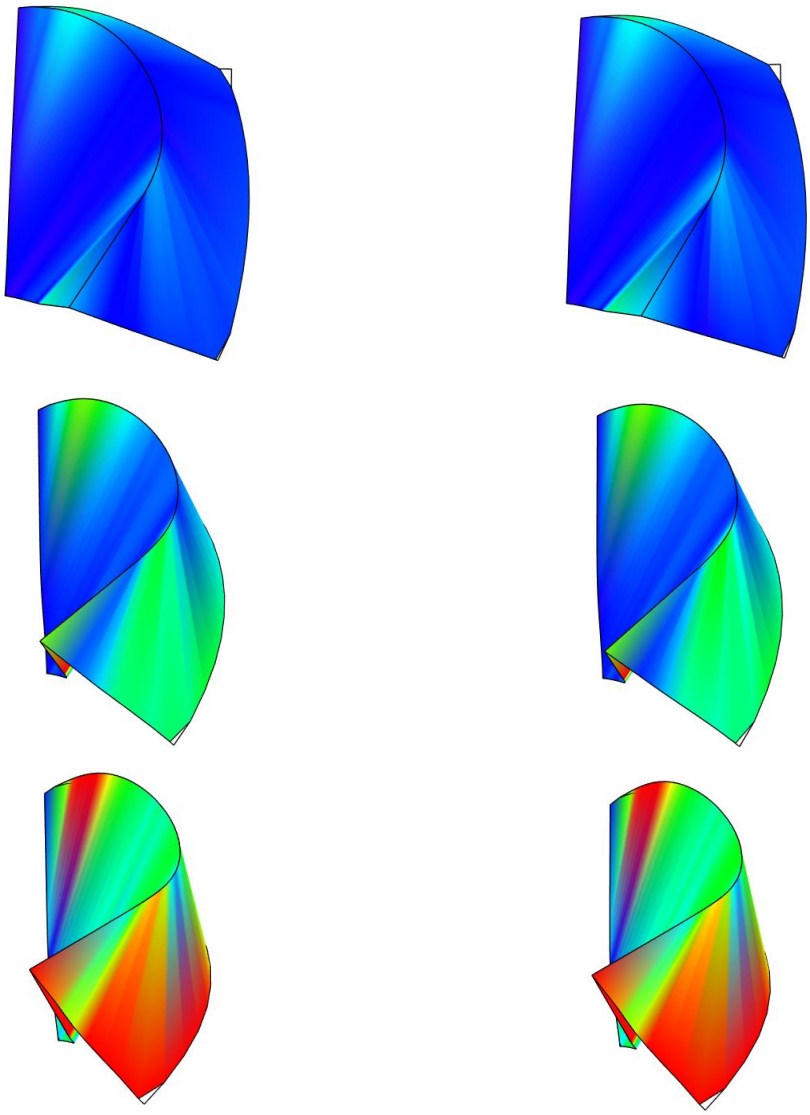
4.2. Simulation of curved folding on Origami simulator

The origami simulator is a web application developed by Amanda Ghassaei. The crease pattern is imported from SVG, a 2D vector format, and the types of the creases is defined by the color of the lines, red is the mountain fold, blue the valley fold, black the boundary of the paper, yellow the facet crease which tries to remain flat, and magenta the edge where no force of folding is added. The software provides the interface to control the degree of folding as *fold percent*. It also visualizes the *strain* on the vertices by the color. The strain is a measurement of how much the material is being stretched or compressed. On this software, blue indicates zero strain and red indicates *max strain*, specified on the GUI. The paper shape is calculated by the optimization of the spring-mass model, allowing some stretching or shearing of the polygonal faces. Figure 4-3 shows the input crease patterns with one curved crease on the paper, and Figure 4-4 shows the results of the simulation using these crease patterns. Figure 4-5 shows the input crease patterns of *origami-sphere*, and Figure 4-6 is its results. Because of the flexibility of the spring-mass model used in the system, apparently, the paper seems to be folded successfully guided by the *fold percent* input by the user. However, by the color indicating the strain, with the configuration of *max strain* being 1%, the strain becomes very high if the generated paper shape does not match the directions of the rulings. Specifically, for *origami-sphere*, the strain becomes very high in Figure 4-6(c) where the sphere is just about to be folded completely. By these observations, it is considered that the polygon models with fixed rulings are not satisfactory for modeling the folding motion of curved folding.



(a) (b)

Figure 4-3 The input crease pattern with one crease.



(a) (b)

Figure 4-4 The result of Origami simulator for crease pattern in Figure 4-3. Top row: fold percent 50%. Middle row: fold percent 75%. Bottom row: fold percent 100%.

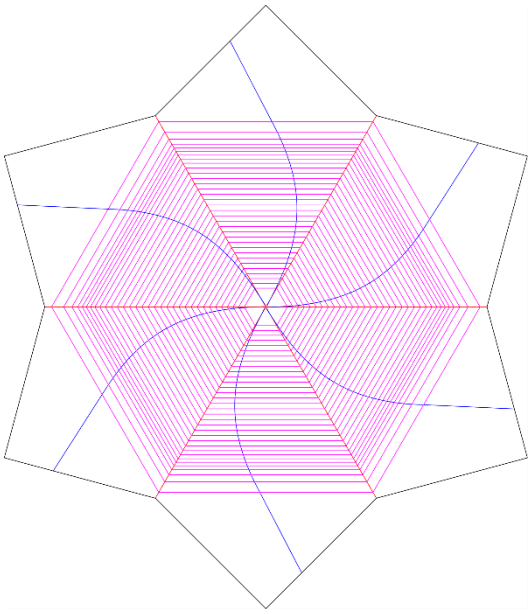


Figure 4-5 The input crease pattern of *origami-sphere*.

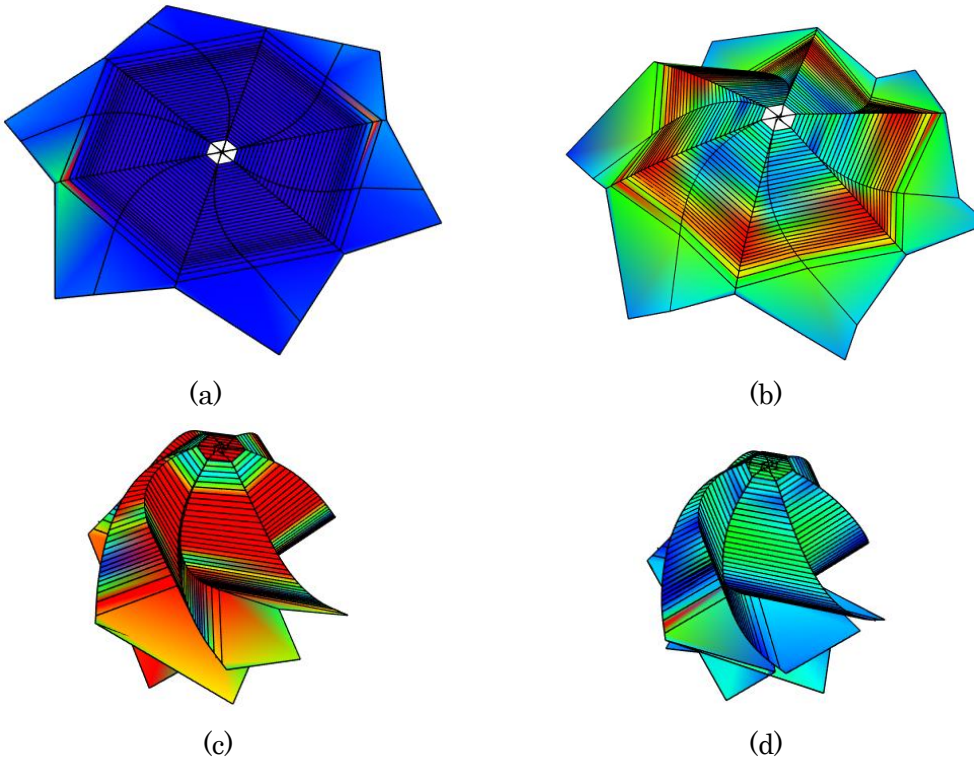


Figure 4-6 The result of Origami simulator for crease pattern in Figure 4-5. (a) Fold percent: 0%. (b) Fold percent: 40%. (c) Fold percent: 60%. (d) Fold percent: 80%.

Chapter 5.
Estimating paper shape from RGBD images

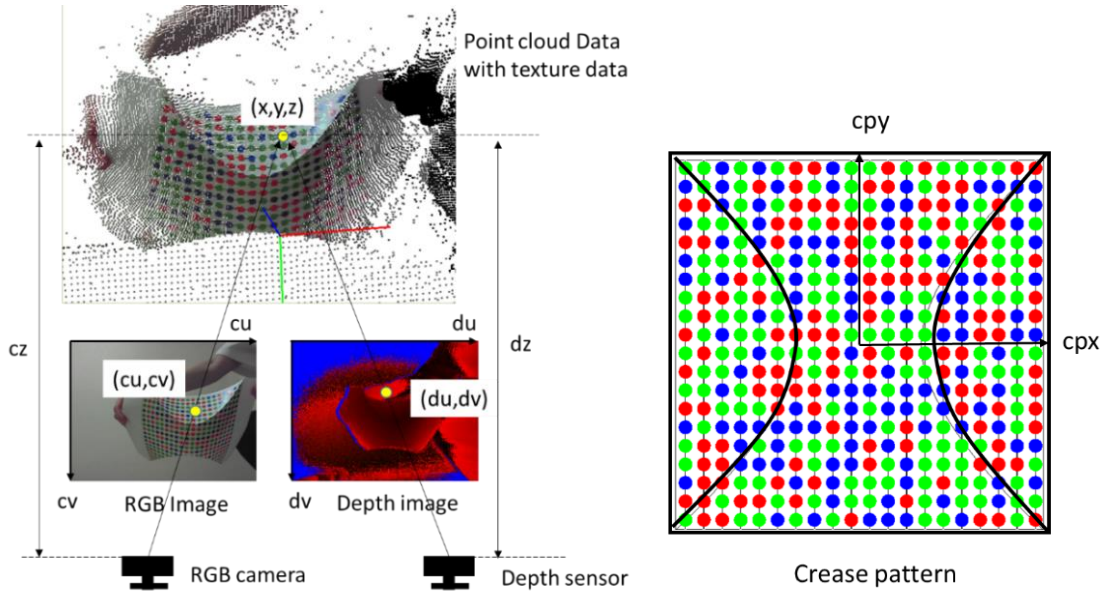
To understand the shape of the paper while being folded, we first made an attempt to reconstruct its shape from a sequence of RGBD images capturing the deformation of the curved folded paper. The procedure is described in Section 5.1. But this trial has been pended as we noticed the specification and the performance of the 3D camera did not meet our requirement to measure the 3D shape of the curved folded paper, as referred in Section 5.2. It may be another meaningful research topic in the future to utilize the partial low-resolution 3D data representing the real paper shape to control the shape of the 3D model generated by our method, described in the following chapters.

5.1. Method

Figure 5-1 shows an example of input data and the predefined configuration used in this work. As shown in Figure 5-1(a), the curved folded paper is captured by a pair of an RGB camera and a depth sensor. From the RGB image and the depth image captured by the input device, 3D point cloud data with texture is generated. Our goal is to estimate the rulings and the 3D shape of the curved folded paper being folded, using the captured 3D data and the color pattern printed on the sheet of paper as shown in Figure 5-1(b). The purpose of using the color pattern is to identify the 3D position of each point of the paper so that the shape and the movement of the paper could be analyzed.

5.1.1. RGBD camera

The input device used for this trial was DepthSense© 325, released from SoftKinetic (Figure 5-2). It is a short distance TOF sensor, capturing the RGB images and the distance map using infrared laser light, at the frame rate of 25-30 fps. The main application of this product is to capture a human gesture rather than the measurement of the 3D shape itself. Nevertheless, it is a low-cost, easily accessible camera which could capture the 3D shape data in a sequence of frames. The specification is described in Table 5-1. The images were captured using the sample program working on SoftKinetic DepthSense SDK.



(a) (b)

Figure 5-1 Data used in paper shape estimation. (a) Captured data. (cu, cv) is pixel position in RGB image, (du, dv) is pixel position in depth image, dz is the depth value on (du, dv), cz is the depth value on (cu, cv) calculated from other parameters, and (x, y, z) is 3D position of a point in the point cloud. (b) Crease pattern with color pattern. (cpx, cpy) is the 2D position on the crease pattern.

Figure 5-2 DepthSense© 325.

Table 5-1 Specification of DepthSense© 325

Depth tracking range	15cm to 1 meter
Depth sensor resolution	QVGA (320×240)
Depth sensor field of view	74° H – 57.9° V
Color sensor resolution	HD (720p)

5.1.2. Camera calibration

From the RGB image and the depth image captured by the input device, the 3D point cloud data is reconstructed and textured using the internal and external camera parameters. Since the default settings of the camera parameters are not accurate for

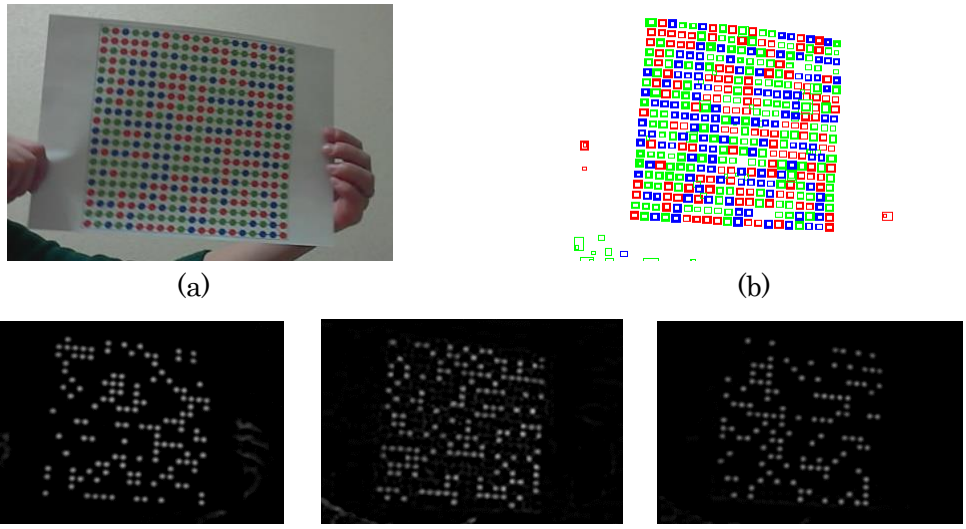
such a low-cost input device, we performed the camera calibration by capturing the checkerboard pattern and re-calculating the camera parameters using OpenCV libraries. Some of the images used for the calibration are shown in Figure 5-3. Because the checkerboard pattern detection on OpenCV library does not support such blurred corners or the black tiles with gaps, the corners were plotted manually and input to the calculation process.

(a) (b)

Figure 5-3 Images used for camera calibration. (a) Depth image. (b) RGB image.

5.1.3. Detection of patterns on RGB images

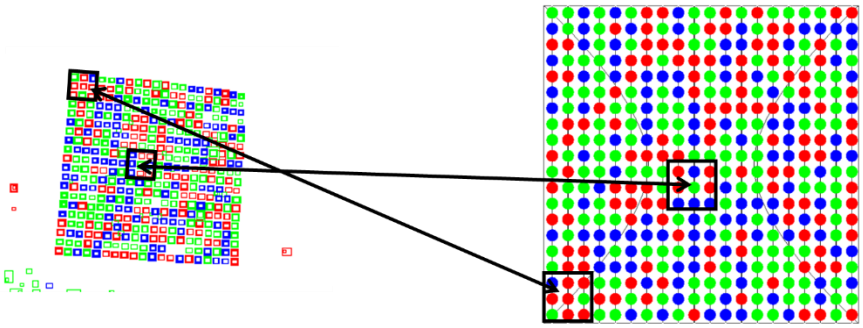
To identify the points on the paper in the RGB images, we used the pseudo-random color pattern proposed by [Scholz_05] (Figure 5-1(b)). The pattern contains 20×20 dots with three colors, red, green, and blue, assigned randomly and verified that each 3×3 pattern is unique in the whole pattern. The predefined color pattern is printed on a piece of paper and captured by the RGBD camera. For each RGB image captured, the dots on the paper are detected from each color component of the image separately and then put together with the color feature (Figure 5-4). The detection of the dots from one component of the image is simply performed by binarizing the gray image with multiple thresholds and detecting blobs in a certain range of size. Then by searching the neighboring dots, 3×3 patterns are detected and matched with the predefined color pattern, as illustrated in Figure 5-5. Then the whole pattern is detected from the image starting from the local 3×3 pattern. As shown in Figure 5-6, the initial detection result of 3×3 patterns is likely to include some false positives, which turn out to be outliers. After removing such outliers, other dots are integrated into the whole pattern by searching the detected blobs in the expected locations. Finally, most of the dots in the pattern is detected from the input RGB image, being able to identify the location of each point in the image.



(a) (b)

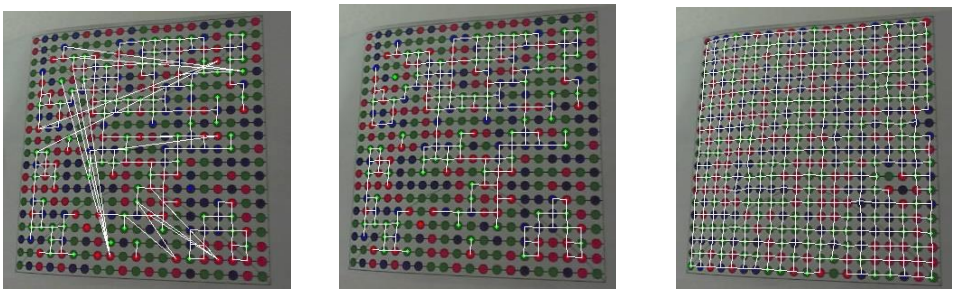
(c) (d) (e)

Figure 5-4 Detection of colored dots. (a) Input image. (b) Result of detection. (c) Red component of the image. (d) Green component. (e) Blue component.



(a) (b)

Figure 5-5 Matching of 3×3 patterns between (a) detected dots and (b) the predefined color pattern.

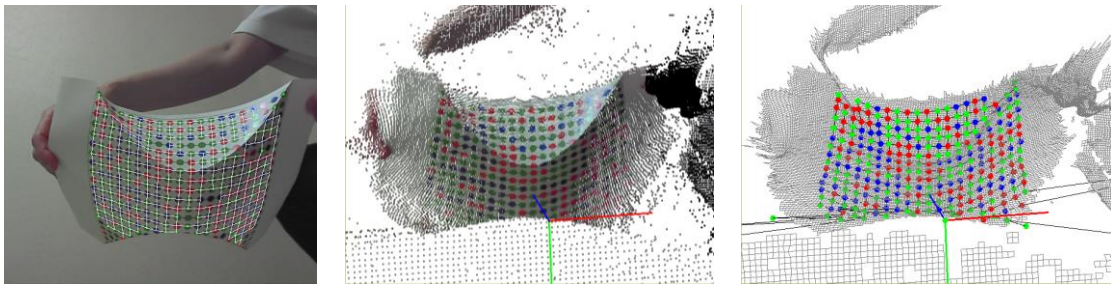


(a) (b) (c)

Figure 5-6 Process of detecting the whole pattern. (a) Initial result of 3×3 patterns. (b) Outliers removed. (c) Other detected dots integrated into the whole pattern.

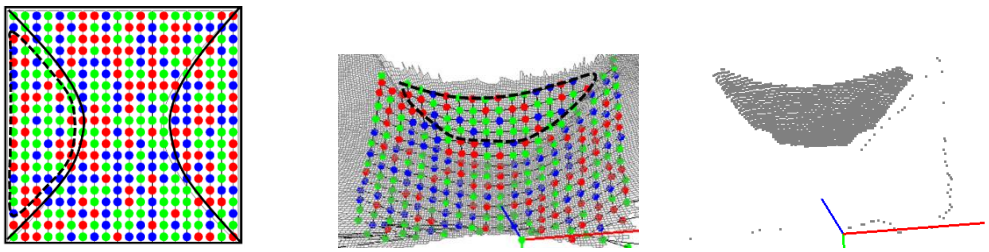
5.1.4. Extraction of developable surface and the rulings

Since the color pattern is detected in the RGB image, it is then projected on to the 3D space. The purpose of this process is to extract the 3D point cloud corresponding to a developable surface, using the dots in 2D and 3D spaces, and to model the 3D data as a polygon model. The projection onto the 3D space is carried out by collecting the 3D point cloud corresponding to the pixels of the detected dots. Figure 5-7 shows an example of a color pattern detected on the curved folded paper and the pattern plotted on the 3D data. Then the point cloud in one developable surface of the curved fold is extracted, as shown in Figure 5-8. This process is carried out by extracting the point cloud which lies in the triangles or quads composed of the dots located in the target developable surface. The rulings are estimated using the method of [Perriallat_13], as shown in Figure 5-9. Sample points are placed on the creases and the paper edges, which compose the boundary of the area. Initially, all pairs of the sample points are the possible rulings. Then the pairs are scored according to the distribution of the point cloud between the two points. As the point cloud is distributed on the lines connecting the two sample points, a higher score is given to that pair. Then the lines connecting the pairs of sample points with scores higher than the threshold are adopted as rulings. If there is a ruling crossing other rulings, the ones with the lower score are eliminated. The faces between the rulings and the boundary of the area are then placed in the 3D space, as in Figure 5-9 (d) and (e). The poses of the faces are calculated by the least square method to approximate the 3D point cloud located on the face.



(a) (b) (c)

Figure 5-7 Projection of detected dots on to 3d data. (a) Color pattern detected on RGB image. (b) Point cloud with color data. (c) Detected color pattern plotted on 3D data.



(a) (b) (c)

Figure 5-8 Extraction of point cloud on one developable surface. (a) predefined area on the color pattern. (b) Corresponding area in the 3D data. (c) Extracted point cloud.

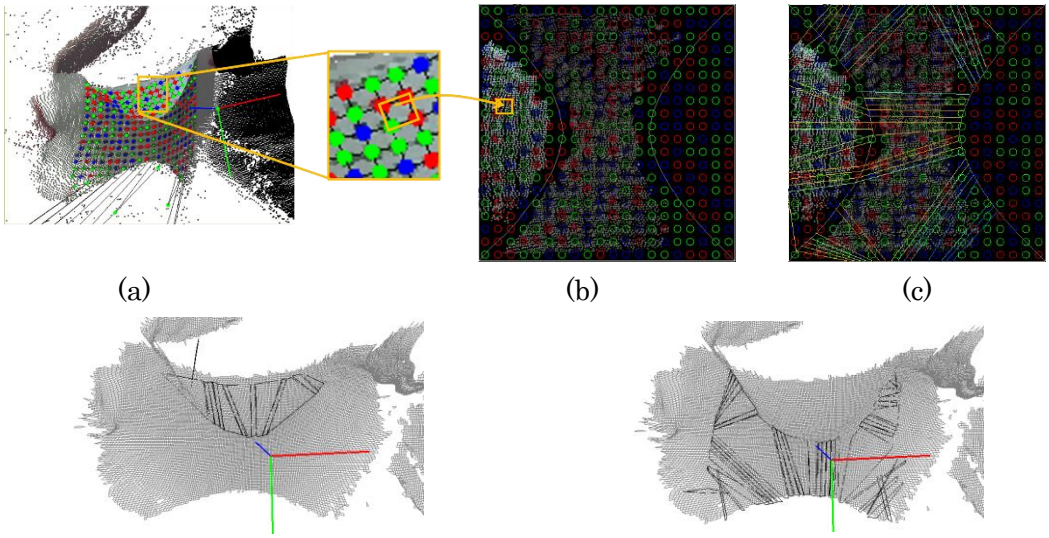


Figure 5-9 Estimation of rulings and making polygon model. (a) Detected dots plotted on 3D space. (b) 3D point cloud plotted on the 2D space, with crease pattern and color pattern. (c) Rulings estimated in 2D space. (d) Polygon model of one surface reprojected in 3D space. (e) Polygon model reprojected to the second surface in 3D space.

5.2. Result and discussion

Although the 3D point cloud on a developable surface is extracted and some rulings are estimated on the surface, we could not go any further to generate the 3D model of the curved folded developable surface solely from the captured point cloud data. One cause for this is that the accuracy of the depth image and the resolutions of the depth and color images are not satisfactory for the 3D measurement. As the depth image is captured by a TOF sensor, the depth value always includes some Gaussian noise, requiring some size of area to cancel the noise by averaging the pixels. Since such data is not suitable in detecting the edges and the sharp creases, we took an approach to first estimate the developable surface surrounded by the creases and paper edges. But this approach also requires the input 3D data to be accurate as the rulings may change according to the small displacement of the surface. As to the image resolution, it is good for detecting 20×20 color pattern on the paper facing the RGB camera. But for curved folding with many creases, the pattern may not have sufficient size, or enough number of dots to identify a point on a paper. In the previous subsection, the example curved folding contains only two curved creases. But as the number of crease increases, the area of a developable surface becomes smaller, and so is the number of dots composing the color pattern. The 20×20 color pattern does not have enough resolution if a developable surface does not contain 3×3 patterns in most of its area.

There is also a problem of capturing a 3D object by a camera designed for a short distance. Usually, such cameras have a relatively wide angle of view to capture a large area in short distance. But this is a tradeoff between the range of the 3D surface directions that the camera is able to capture. In other words, a camera with a wide

angle of view can only capture the surface facing close to the camera. So, in case the 3D object in a rounded shape, only a small part of its surface is captured. With more creases, the occlusion by the other surfaces would be another obstacle.

With above reasons, we concluded that it is impossible to measure and analyze the 3D shape only from the captured data and that some pre-modeling of the developable surface is necessary to estimate the 3D shape from such partial and low-resolution data. The modeling method proposed and described in the following chapters is motivated by this trial.

Chapter 6.
Curved folding with one crease

As a base of our work, this chapter introduces the modeling and visualization of a curved fold with one curved crease. The content includes the geometry and the calculation method of the paper shape in Section 6.1, the procedure of designing the curved fold and the generation of the folding motion in Section 6.2, and the evaluation of a simple curved fold as an example in Section 6.3. In this chapter, we assume that the crease curve includes no inflection point.

6.1. Calculation of crease and rulings

In this section, the basic calculation method of the curved crease and the rulings, which defines the 3D shape of the curved folded paper, is explained. It is based on the equations from Chapter 2. In Subsection 6.1.1, the discrete version of the curve, the vectors, and the parameters are defined, and the calculation method used in our system is explained. In Subsection 6.1.2, the actual procedure of the calculation is introduced.

6.1.1. Discrete expression of the mathematical formula

To apply the equations introduced in Chapter 2 to the calculation of the curved crease in a polygon model, we express the curve in the discrete form $\{\boldsymbol{X}_i\}$, a sequence of sample points aligned along the curve $\boldsymbol{X}(s)$. Similarly, the vectors and parameters on the point $\boldsymbol{X}_i$ are denoted as $\boldsymbol{T}_i$, $\boldsymbol{N}_i$, $\boldsymbol{B}_i$, k_i, τ_i, α_i, and k_{2D_i}, instead of arc-length parametric functions $\boldsymbol{T}(s)$, $\boldsymbol{N}(s)$, $\boldsymbol{B}(s)$, $k(s)$, $\tau(s)$, $\alpha(s)$, and $k_{2D}(s)$. $d\boldsymbol{X}_i$, the derivative of $\boldsymbol{X}_i$, is calculated as follows, using three consecutive sample points $\{\boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \boldsymbol{X}_{i+1}\}$

$$d\boldsymbol{X}_i = \frac{1}{2}\left(\frac{d\boldsymbol{X}_{i,i-1}}{|d\boldsymbol{X}_{i,i-1}|} + \frac{d\boldsymbol{X}_{i+1,i}}{|d\boldsymbol{X}_{i+1,i}|}\right), \tag{6.1}$$

where $d\boldsymbol{X}_{i+1,i} = \boldsymbol{X}_{i+1} - \boldsymbol{X}_i$, is a vector connecting two sample points. Similarly, derivatives of the vectors $d\boldsymbol{T}_i$, $d\boldsymbol{N}_i$, and $d\boldsymbol{B}_i$ are calculated as

$$\begin{cases} d\boldsymbol{T}_i = \frac{1}{2}\left(\frac{\boldsymbol{T}_i - \boldsymbol{T}_{i-1}}{|d\boldsymbol{X}_{i,i-1}|} + \frac{\boldsymbol{T}_{i+1} - \boldsymbol{T}_i}{|d\boldsymbol{X}_{i+1,i}|}\right) \\ d\boldsymbol{N}_i = \frac{1}{2}\left(\frac{\boldsymbol{N}_i - \boldsymbol{N}_{i-1}}{|d\boldsymbol{X}_{i,i-1}|} + \frac{\boldsymbol{N}_{i+1} - \boldsymbol{N}_i}{|d\boldsymbol{X}_{i+1,i}|}\right), \\ d\boldsymbol{B}_i = \frac{1}{2}\left(\frac{\boldsymbol{B}_i - \boldsymbol{B}_{i-1}}{|d\boldsymbol{X}_{i,i-1}|} + \frac{\boldsymbol{B}_{i+1} - \boldsymbol{B}_i}{|d\boldsymbol{X}_{i+1,i}|}\right) \end{cases} \tag{6.2}$$

which are the averages of two differences between the consecutive points, normalized by the distance. Figure 6-1 shows an example of $\boldsymbol{T}_i$ and $\boldsymbol{N}_i$ calculated as the discrete derivatives $d\boldsymbol{X}_i$ and $d\boldsymbol{T}_i$. 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 3D curve $\{\boldsymbol{X}_i\}$, using the following equations. According to equation (2.1) and the fact that $\boldsymbol{T}_i$, $\boldsymbol{N}_i$, $\boldsymbol{B}_i$ are unit vectors,

$$\begin{cases} \boldsymbol{T}_i = \frac{d\boldsymbol{X}_i}{|d\boldsymbol{X}_i|} \\ \boldsymbol{N}_i = \frac{d\boldsymbol{T}_i}{|d\boldsymbol{T}_i|} \\ \boldsymbol{B}_i = \boldsymbol{T}_i \times \boldsymbol{N}_i \end{cases}. \tag{6.3}$$

The discrete version of equation (2.4) and (2.5) are

$$k_i = |d\boldsymbol{T}_i|, \tag{6.4}$$

$$\tau_i = |d\boldsymbol{N}_i + k_i\boldsymbol{T}_i| = |-d\boldsymbol{B}_i|. \tag{6.5}$$

Because the vectors and the parameters in the discrete form are calculated using three consecutive sample points $\{\boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \boldsymbol{X}_{i+1}\}$, they are not calculated on the edge of the crease curve. Having **N** vertices $\{\boldsymbol{X}_0, \dots, \boldsymbol{X}_{N-1}\}$ on the crease curve, the tangent vector $\boldsymbol{T}_i$ is calculated on $\{\boldsymbol{X}_1, \dots, \boldsymbol{X}_{N-2}\}$, the normal vector $\boldsymbol{N}_i$, the binormal vector $\boldsymbol{B}_i$ and the curvature k_i are calculated on $\{\boldsymbol{X}_2, \dots, \boldsymbol{X}_{N-3}\}$, and the torsion τ_i is calculated on $\{\boldsymbol{X}_3, \dots, \boldsymbol{X}_{N-4}\}$. In generating the 3D polygon model, the three vertices on each edge are not used, making $\boldsymbol{X}_3$ and $\boldsymbol{X}_{N-4}$ to be used as the vertices on the ends.

Conversely to the calculation described above, the 3D curve $\{\boldsymbol{X}_i\}$ can be obtained from a sequence of curvatures $\{k_i\}$ and torsions $\{\tau_i\}$. To carry it out, the system calculates the vectors $\{\boldsymbol{T}_i\}$, $\{\boldsymbol{N}_i\}$, $\{\boldsymbol{B}_i\}$ sequentially from $\{k_i\}$ and $\{\tau_i\}$, and then $\{\boldsymbol{X}_i\}$ from $\{\boldsymbol{T}_i\}$. As a first step, the vectors on one end of the curve are set as unit vectors orthogonal to each other, for example,

$$[\boldsymbol{T}_0 \quad \boldsymbol{N}_0 \quad \boldsymbol{B}_0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6.6}$$

Then, using curvature, torsion, and the vectors on the first vertex, the vectors on the second vertex are calculated as

$$\begin{cases} \boldsymbol{T}_1 = \boldsymbol{T}_0 + d\boldsymbol{T}_0 = \boldsymbol{T}_0 + k_0 dx \boldsymbol{N}_0 \\ \boldsymbol{N}_1 = \boldsymbol{N}_0 + d\boldsymbol{N}_0 = \boldsymbol{N}_0 - k_0 dx \boldsymbol{T}_0 + \tau_0 dx \boldsymbol{B}_0, \\ \boldsymbol{B}_1 = \boldsymbol{T}_1 \times \boldsymbol{N}_1 \end{cases} \tag{6.7}$$

expanding $d\boldsymbol{T}_0$, $d\boldsymbol{N}_0$, $d\boldsymbol{B}_0$ by substituting $\boldsymbol{T}'(s)$, $\boldsymbol{N}'(s)$, $\boldsymbol{B}'(s)$, $k(s)$, $\tau(s)$ in equation (2.2) with $d\boldsymbol{T}_i$, $d\boldsymbol{N}_i$, $d\boldsymbol{B}_i$, k_i, τ_i. Here, dx is the distance between two consecutive sample points, which we set constant. The vectors on the following points are calculated as

$$\begin{cases} \boldsymbol{T}_i = \boldsymbol{T}_{i-2} + 2d\boldsymbol{T}_{i-1} = \boldsymbol{T}_{i-2} + 2k_{i-1} dx \boldsymbol{N}_{i-1} \\ \boldsymbol{N}_i = \boldsymbol{N}_{i-2} + 2d\boldsymbol{N}_{i-1} = \boldsymbol{N}_{i-2} - 2k_{i-1} dx \boldsymbol{T}_{i-1} + 2\tau_{i-1} dx \boldsymbol{B}_{i-1}, \\ \boldsymbol{B}_i = \boldsymbol{T}_i \times \boldsymbol{N}_i \end{cases} \tag{6.8}$$

considering that the derivatives $d\boldsymbol{T}_i$, $d\boldsymbol{N}_i$, $d\boldsymbol{B}_i$ are calculated using three points as in equation (6.1). Figure 6-2 shows $\boldsymbol{T}_i$ derived from $\boldsymbol{T}_{i-2}$ and $d\boldsymbol{T}_{i-1} = k_{i-1} dx \boldsymbol{N}_{i-1}$, using equation (6.8). At last, sample points on the curve are calculated as follows.

$$\boldsymbol{X}_i = \boldsymbol{X}_{i-1} + dx\frac{\boldsymbol{T}_i+\boldsymbol{T}_{i-1}}{|\boldsymbol{T}_i+\boldsymbol{T}_{i-1}|}. \tag{6.9}$$

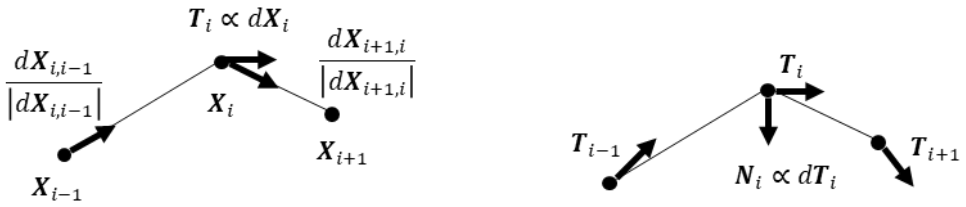
Similarly, sample points of the 2D curve $\{\boldsymbol{X}_{\boldsymbol{2D}_i}\}$ are calculated from the sequence of 2D curvatures $\{k_{2D_i}\}$ as follows.

$$[\boldsymbol{T}_{\boldsymbol{2D}_0} \quad \boldsymbol{N}_{\boldsymbol{2D}_0}] = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \tag{6.10}$$

$$\begin{cases}\boldsymbol{T}_{\boldsymbol{2D}_1} = \boldsymbol{T}_{\boldsymbol{2D}_0} + k_{2D_0}dx\boldsymbol{N}_{\boldsymbol{2D}_0}\\ \boldsymbol{N}_{\boldsymbol{2D}_1} = \begin{bmatrix}-\boldsymbol{T}_{\boldsymbol{2D}_1}y\\ \boldsymbol{T}_{\boldsymbol{2D}_1}x\end{bmatrix}\end{cases}, \tag{6.11}$$

$$\begin{cases}\boldsymbol{T}_{\boldsymbol{2D}_i} = \boldsymbol{T}_{\boldsymbol{2D}_{i-2}} + 2k_{2D_{i-1}}dx\boldsymbol{N}_{\boldsymbol{2D}_{i-1}}\\ \boldsymbol{N}_{\boldsymbol{2D}_i} = \begin{bmatrix}-\boldsymbol{T}_{\boldsymbol{2D}_i}y\\ \boldsymbol{T}_{\boldsymbol{2D}_i}x\end{bmatrix}\end{cases}, \tag{6.12}$$

$$\boldsymbol{X}_{\boldsymbol{2D}_i} = \boldsymbol{X}_{\boldsymbol{2D}_{i-1}} + dx\frac{\boldsymbol{T}_{\boldsymbol{2D}_i}+\boldsymbol{T}_{\boldsymbol{2D}_{i-1}}}{|\boldsymbol{T}_{\boldsymbol{2D}_i}+\boldsymbol{T}_{\boldsymbol{2D}_{i-1}}|} \tag{6.13}$$



(a) (b)

Figure 6-1 Vectors in Frenet frame calculated as discrete derivatives of three consecutive sample points $\{\boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \boldsymbol{X}_{i+1}\}$. (a) Tangent vector $\boldsymbol{T}_i$. (b) Normal vector $\boldsymbol{N}_i$.

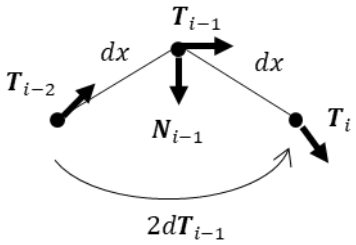


Figure 6-2 Cumulative vector calculation on sequence of vertices.

6.1.2. The calculation process of crease and rulings

In our system, the parameters on the sample points, $\{k_i\}$, $\{\tau_i\}$, $\{\alpha_i\}$, and $\{k_{2D_i}\}$, are calculated by the spline interpolation of the parameters on the control points placed on the curved crease in even intervals, as shown in Figure 6-3. The spline curves are in degree-three, making the parameters to change smoothly along the curve and assuring the rulings and the 3D shape to be also changing smoothly. With the parameters on the sample points, the shapes of the 2D and 3D curves are reconstructed, as in Figure 6-3(c). The shapes of the curves are guaranteed of the G2 continuity, as the curvatures and the torsion are smooth functions with respect to arc length. Then the ruling directions are derived by the equations in the previous subsections. The steps of this process are listed below, and the parameters and elements processed in each step are summarized in Table 6-1 for cases A, B, and C described in Section 2.2.

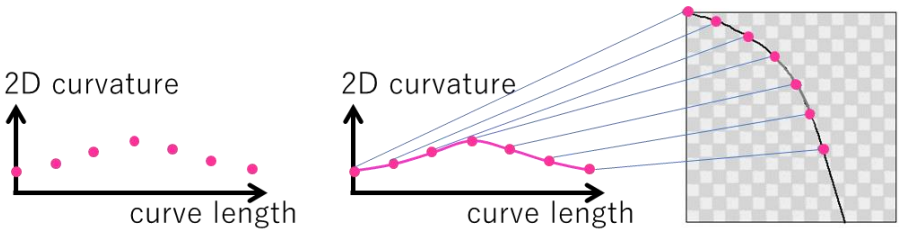


Figure 6-3 Interpolation and reconstruction of crease curve. (a) 2D curvatures on the control points. (b) 2D curvatures interpolated. (c) Reconstructed 2D curve.

Step 1. Parameters on all vertices are derived from the parameters of the control points by spline interpolation. The rectification of torsion and folding angle, which will be introduced in Chapter 7, is applied at this stage if needed.

Step 2. The 3D curve $\{\boldsymbol{X}_i\}$ and/or 2D curve $\{\boldsymbol{X}_{2D_i}\}$, are reconstructed from the given parameters using equations (6.6) to (6.9) and equations (6.10) to (6.13).

Step 3. The parameters and vectors are recalculated from the 2D and 3D curves to avoid the accumulation of numerical errors, using equations (6.13) to (6.15) or their 2D version.

Step 4. Using equation (2.7), the parameters to be derived, depending on which case A-C is used, are updated having the other two parameters as inputs.

Step 5. In cases A and B, 3D curve $\{\boldsymbol{X}_i\}$ or 2D curve $\{\boldsymbol{X}_{2D_i}\}$, are reconstructed using the derived parameters.

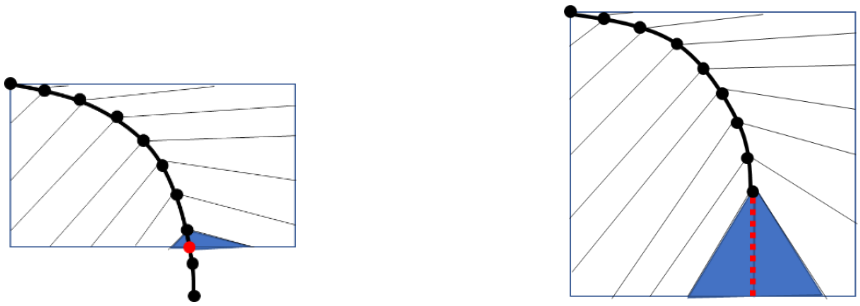
Step 6. The parameters of the reconstructed curve are recalculated.

Step 7. The rulings are calculated using equations (2.8) to (2.11).

Table 6-1 Parameters and elements processed in each step.

		Case A	Case B	Case C
	Input elements	3D curve Folding angle	2D curve Folding angle	2D curve 3D curve
	Derived element	2D curve	3D curve	Folding angle
Step 1	Interpolated parameters	$\{k_i\},\ \{\tau_i\},\ \{\alpha_i\}$	$\{k_{2D_i}\}, \{\alpha_i\}$	$\{k_i\},\ \{\tau_i\},\ \{k_{2D_i}\}$
	Rectified parameters	$\{\tau_i\},\ \{\alpha_i\}$	$\{\alpha_i\}$	$\{\tau_i\}$
Step 2	Reconstructed curve	$\{\boldsymbol{X}_i\}$	$\{\boldsymbol{X}_{2D_i}\}$	$\{\boldsymbol{X}_i\}, \{\boldsymbol{X}_{2D_i}\}$
Step 3	Recalculated parameters and vectors	$\{k_i\},\ \{\tau_i\}$ $\{\boldsymbol{T}_i\},\ \{\boldsymbol{N}_i\},\ \{\boldsymbol{B}_i\}$	$\{k_{2D_i}\},$ $\{\boldsymbol{T}_{2D_i}\},\ \{\boldsymbol{N}_{2D_i}\}$	$\{k_i\},\ \{\tau_i\},\ \{k_{2D_i}\}$ $\{\boldsymbol{T}_i\},\ \{\boldsymbol{N}_i\},\ \{\boldsymbol{B}_i\},$ $\{\boldsymbol{T}_{2D_i}\},\ \{\boldsymbol{N}_{2D_i}\}$
Step 4	Derived parameters	$\{k_{2D_i}\}$	$\{k_i\}$	$\{\alpha_i\}$
Step 5	Reconstructed curve	$\{\boldsymbol{X}_{2D_i}\}$	$\{\boldsymbol{X}_i\}$	-
Step 6	Recalculated parameters and vectors	$\{k_{2D_i}\},$ $\{\boldsymbol{T}_{2D_i}\},\ \{\boldsymbol{N}_{2D_i}\}$	$\{k_i\},\ \{\tau_i\}$ $\{\boldsymbol{T}_i\},\ \{\boldsymbol{N}_i\},\ \{\boldsymbol{B}_i\}$	-
Step 7	Rulings calculated	$\{\beta_{L_i}\}, \{\beta_{R_i}\},$ $\{\boldsymbol{r}_{L_i}\},\ \{\boldsymbol{r}_{R_i}\}$	$\{\beta_{L_i}\}, \{\beta_{R_i}\},$ $\{\boldsymbol{r}_{L_i}\},\ \{\boldsymbol{r}_{R_i}\}$	$\{\beta_{L_i}\}, \{\beta_{R_i}\},$ $\{\boldsymbol{r}_{L_i}\},\ \{\boldsymbol{r}_{R_i}\}$

After all curves and rulings are derived, the developable surface is constructed as two quad strips adjacent to the curved crease. The quads in the strips share the rulings as their two edges, and the crease curve and the boundary of the paper being the other two. As shown in Figure 6-4(a), if the end of the crease curve lies outside of the paper, making the curve to cross the paper edge, the part of the curve outside the paper is cut off. This is implemented by placing a new vertex on the intersection of the curve and the paper edge and placing a new quad or a triangle on its sides. If the end of the curve is located in the paper, as shown in Figure 6-4(b), the curve end is extended to the edge of the paper as a straight line, and flat polygons with no rulings are placed on its sides.



(a) (b)

Figure 6-4 Polygon strip on the end of crease curve. (a) End of crease curve placed outside the paper. New vertex is shown in red. (b) End of crease curve placed inside of the paper. Extended crease is shown in red dotted line. In both figures, newly added triangles (or quads in some cases) are shown in blue.

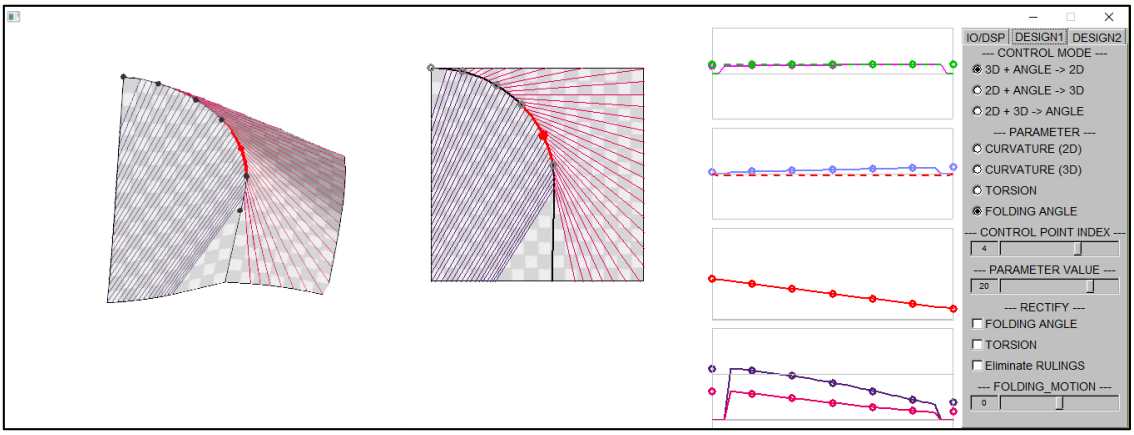
6.2. Designing the curved folding

In this section, the procedure of designing the curved folded paper is explained in terms of user manipulation.

6.2.1. The GUI system

As shown in Figure 6-5, the prototype of the GUI system has four panes, (a) the 3D polygon model of curved folded paper, (b) Crease and rulings mapped on 2D space, (c) the graphs of the parameters along the curve, and (d) the control panel through which the user can manipulate the shape of the paper. The visualization of the 3D model is realized by *OpenGL*. The panes and the widgets are implemented using *fltk 1.3.0* library and support mouse input on the 3D and 2D panes, key input, and the inputs through the widgets on the control panel. Through the mouse input, the user can change the view angle and the pose and the position of the object in 3D space. In the 2D pane, the user can change the curve position, curve orientation, or paper size. The manipulation by the mouse input is summarized in Table 6-2. Through the control panel and the key input, the user sets and adjusts the parameters on the control points on the crease curve. The radio buttons under the CONTROL MODE corresponds to the cases A, B, and C described in Subsection 6.1.2. By choosing the mode, the radio buttons under the PARAMETER are activated or deactivated based on whether it is the input parameter or the derived parameter in that mode. The user can choose one of the activated radio buttons to specify which parameter to be modified, and then choose the control point whose parameter is going to be changed by the scroll bar CONTROL POINT INDEX. The control point chosen is highlighted in the 3D and the 2D panes, with the same color as the parameter graphs, green for 3D curvature, purple for 2D curvature, blue for torsion, and red for folding angle. The value of the parameter is now changed and adjusted through a scroll bar PARAMETER VALUE or key inputs, while checking the shapes of the curved crease, rulings, and the paper shape in 2D and 3D and the shape of the parameter graphs. The control panel also supports other operations such as showing or hiding of the visual information, folding and unfolding of the crease

(Subsection 6.2.3), activation and deactivation of the rectification process (Chapter 7), and other operations described in the following chapters.



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

Figure 6-5 Prototype of GUI system. (a) 3D polygon. (b) 2D mapped crease and rulings. (c) Graphs of parameters. (d) Control panel.

Table 6-2 The operation by mouse input.

Pane	Operation	Effect
3D	Shift + Left mouse button	Viewpoint rotation
	Ctrl + Left mouse button	Viewpoint translation
	Alt + Left mouse button	Viewpoint scaling
	Shift + Right mouse button	Object rotation
	Ctrl + Right mouse button	Object translation
2D	Left mouse button + drag	Crease rotation
	Right mouse button + drag	Crease translation
	Left mouse button + drag on paper edge	Paper size modification

6.2.2. The design process of the shape of curved folding

The procedure of designing the curved folding is shown in Figure 6-6. The parameters adjusted by the user are the curvature $\{k_i\}$ and the torsion $\{\tau_i\}$ of the 3D curve, the curvature of the 2D curve $\{k_{2D_i}\}$, and the folding angle $\{\alpha_i\}$. The mode, the parameter, and the control point are specified by the input through the control panel, as described in Subsection 6.2.1. As the user changes the parameter value on a control point, the ruling directions and the shape of the surface are updated instantly, so that the user can adjust the parameters according to the feedback. After checking the 2D and 3D shapes, the user can go back to any step of choosing the mode, the parameter, or the control point. It may be necessary to adjust different parameters on different control points alternately to create the desired shape that is developable, has no rulings crossing, and no self-

collisions. After obtaining the desired shape, the user can check the folding motion, explained in Subsection 6.2.3, and go back to the previous steps if needed.

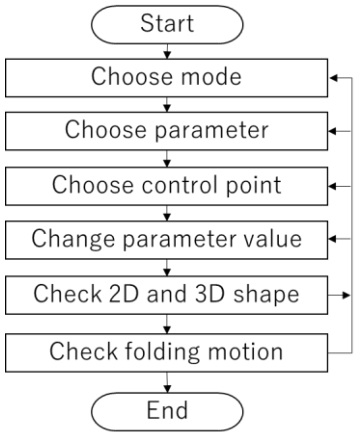


Figure 6-6 The iteration of user input.

6.2.3. Simulation of a folding motion

The folding and unfolding motion of the curved folded paper is simulated, as shown in Figure 6-7, by changing the overall folding angle $\{\alpha_i\}$ and the torsion $\{\tau_i\}$. To unfold the paper shape, the system interpolates the parameters linearly from the designed value to zero, the flat state, while 2D curvature $\{k_{2D_i}\}$ is fixed. The rest of the parameters are re-calculated, and polygon shapes are updated in the same method as in Section 6.1. To fold the paper, the system linearly changes the folding angle $\{\alpha_i\}$ from the designed value to nearly $\pi/2$, with the 2D curvature being fixed, and other parameters and polygon shapes are updated. Note that the folding angle of a curved crease cannot reach $\pi/2$, the completely folded state. In both folding and unfolding, the crossing of the rulings may occur at some point of the motion, as a result of the ruling transition.

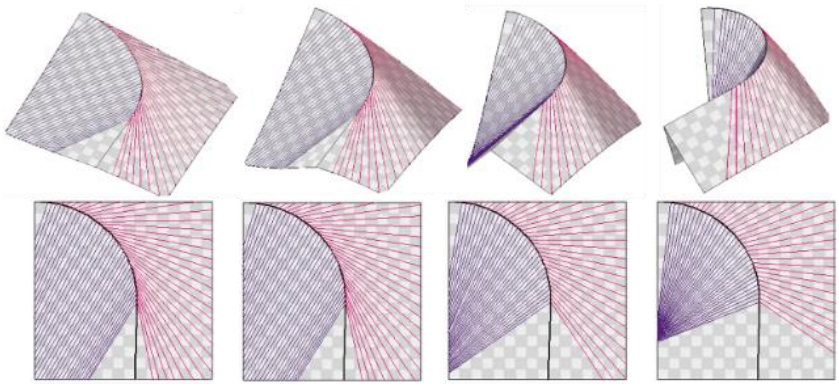


Figure 6-7 Folding motion. Top: 3D polygon model. Bottom: 2D configuration.

6.2.4. Design of curved folding from ruling directions

In the methods described in the previous subsections, the parameters of the crease curve, or the 2D and 3D curvature, the torsion, and the folding angle are the inputs, and the ruling angles were the derived parameters. Conversely, using the same equations, the parameters of the crease curve may be derived from the ruling angles. By this method, the user can specify the directions of the rulings to design the paper shape.

By integrating equations (6.7) and (6.8) and deforming the formula with $k(s)\sin\alpha(s) = k_{2D}(s)\tan\alpha(s)$, following equations are derived,

$$\alpha(s)'/\tan\alpha(s) = \frac{1}{2}\,k_{2D}(s)(-\cot\beta_L(s) + \cot\beta_R(s)), \tag{6.14}$$

$$\tau(s)/\tan\alpha(s) = \frac{1}{2}\,k_{2D}(s)(\cot\beta_L(s) - \cot\beta_R(s)). \tag{6.15}$$

The equations take the 2D curvature and the ruling angles as input and derive the differentiation of the folding angle and the torsion. Then the 3D curvature is also derived by equation (2.7). This means that the user can change the ruling directions on the fixed 2D crease curve to control the 3D paper shape defined by the 3D crease curve and the folding angle. The procedure is listed below.

Step 1. The user inputs the ruling angles on the control points through the GUI. The ruling angles on all vertices $\{\beta_{L_i}\}$, $\{\beta_{R_i}\}$ are derived by spline interpolation.

Step 2. $d\alpha_i/\tan\alpha_i$, $\tau_i/\tan\alpha_i$, which are the discrete versions of $\alpha(s)'/\tan\alpha(s)$, $\tau(s)/\tan\alpha(s)$, are calculated for all vertices using equations (6.14), (6.15).

Step 3. The folding angle α_m on the vertex $\boldsymbol{X}_m$ is set. $\boldsymbol{X}_m$ may be any vertex on the crease curve. In our implementation, it is set as the vertex in the middle point of the crease curve, or $m = |\boldsymbol{X}_i|$.

Step 4. For all other vertices, the folding angle α_i and the torsion τ_i is calculated sequentially, from $\boldsymbol{X}_m$ to the two ends of the crease curve, using the folding angle on the previous vertex α_{i-1} and the pre-calculated $d\alpha_i/\tan\alpha_i$ and $\tau_i/\tan\alpha_i$.

$$\alpha_i = \alpha_{i-1} + d\alpha_{i-1}dx, \tag{6.16}$$

$$\tau_i = \{\tau_i/\tan\alpha_i\}\tan\alpha_i, \tag{6.17}$$

$$d\alpha_i = \{d\alpha_i/\tan\alpha_i\}\tan\alpha_i. \tag{6.18}$$

6.3. Results and discussions

As an evaluation of the method and the system, the 3D shape of the curved folding designed by the system was evaluated by its developability and by visual comparison with the real paper. The following subsections describe the method of evaluation, which is also used in the following chapters. The applicability or the pros and cons of this basic method is also discussed.

6.3.1. Evaluation of the developability

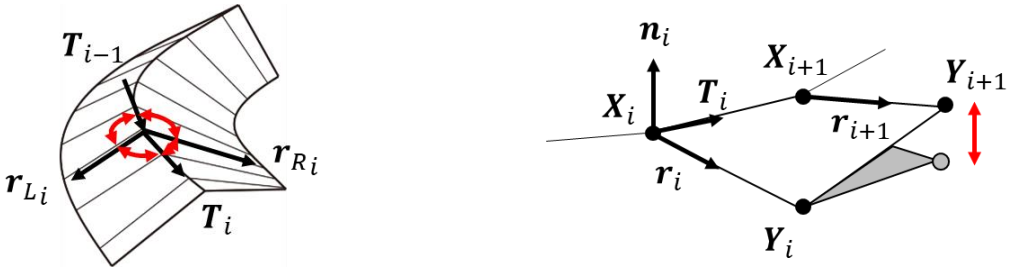
Developability of the 3D polygon model, representing the curved folded paper, is evaluated by (a) the sum of the corner angles adjacent to a vertex and (b) the flatness of the quads. If the polygon model is completely developable, then (a) should be 2π for all vertices. So, we evaluate the error at a vertex $\boldsymbol{X}_i$ as

$$Error_i = 2\pi - \{\angle(\boldsymbol{T}_i, \boldsymbol{r}_{L_i}) + \angle(\boldsymbol{T}_i, \boldsymbol{r}_{R_i}) + \angle(-\boldsymbol{T}_{i-1}, \boldsymbol{r}_{L_i}) + \angle(-\boldsymbol{T}_{i-1}, \boldsymbol{r}_{R_i})\}, \tag{6.19}$$

with $\angle(\boldsymbol{v}_i, \boldsymbol{v}_j)$ indicating the angle between vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. The right term represents the sum of corner angles around the vertex $\boldsymbol{X}_i$, as shown in Figure 6-8(a). We set the tolerance of the angle error to be 0.05 degrees, or 8.7×10^{-4} radians, which corresponds to 1.6mm of displacement in the direction of the normal vector for the paper size with a radius of 100mm. The flatness of a quad is calculated as the distance between one vertex of the quad and the plane passing through the other three vertices, as shown in Figure 6-8(b). This value should be zero for perfectly flat quads and is formulated as

$$\begin{cases} Dist_i = \boldsymbol{n}_i \cdot (\boldsymbol{Y}_{i+1} - \boldsymbol{X}_i) \\ \boldsymbol{n}_i = \boldsymbol{T}_i \times \boldsymbol{r}_i / |\boldsymbol{T}_i \times \boldsymbol{r}_i| \\ \boldsymbol{Y}_i = \boldsymbol{X}_i + l_{r_i} \boldsymbol{r}_i \end{cases}, \tag{6.20}$$

where $\boldsymbol{Y}_i$ is the 3D position of the vertex on the end of the ruling expanding from $\boldsymbol{X}_i$ and on the paper boundary, with l_{r_i} indicating the length of the ruling. $\boldsymbol{n}_i$ is the normal vector of the plane passing through the vertices $\boldsymbol{X}_i$, $\boldsymbol{X}_{i+1}$, and $\boldsymbol{Y}_i$. We set the tolerance of the displacement to be 0.1mm for the paper size of 200mm × 200mm, used in Chapters 6, 7, and 8.

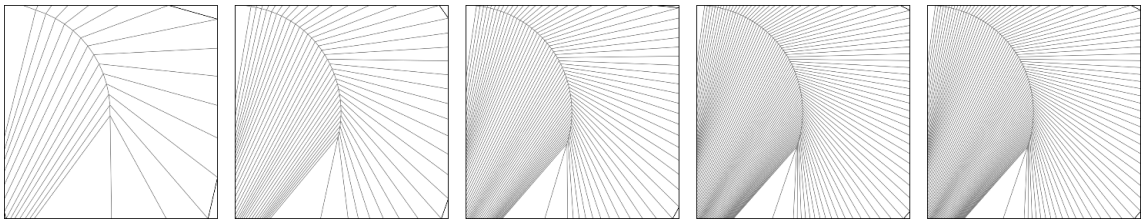


(a) (b)

Figure 6-8 Indices for evaluation of developability. (a) Sum of corner angles. (b) Flatness of quads.

Figure 6-9 shows the 2D crease curves and the rulings in different resolutions of discretization, which are 20, 40, 60, 80, and 100 vertices on the crease. For (a), all vertices on the curve except for the ones on the edge of the paper is used for the evaluation. For (b), all quads placed between two rulings are used. Table 6-3 and Figure 6-10 shows the average and the maximum of (a). Table 6-4 and Figure 6-11 shows the average and the maximum of (b). As the result for (a), the error was below

the tolerance for all vertices in resolution 60 and higher, and for the average in resolution 40. For (b), the displacement was below the tolerance for resolution 40 and higher. So, we could assume that the model should be in the resolution of at least 40 vertices on a crease curve to well represent a developable surface. Because the error becomes larger for the polygon model in low resolution, the error, which should be zero for a perfectly developable surface, should be caused by the discretization of the shape. To discuss about the Gaussian curvature of the surface, by the result of (a), the Gaussian curvature is nearly zero on all the vertices on the crease curve. With the result of (b) that all quads adjacent to the curve are flat, the Gaussian curvature is also nearly zero on all points of the quads.



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

Figure 6-9 2D crease curve and ruling in different resolutions of discretization. The number of vertices on the crease are (a) 20, (b) 40, (c) 60, (d) 80, (e) 100.

Table 6-3 Evaluation of developability by sum of corner angles adjacent to a vertex.

resolution	20	40	60	80	100
average [radian]	7.9×10^{-4}	1.6×10^{-4}	5.5×10^{-5}	2.5×10^{-5}	1.3×10^{-5}
maximum [radian]	4.8×10^{-3}	1.4×10^{-3}	4.2×10^{-4}	1.8×10^{-4}	8.8×10^{-5}

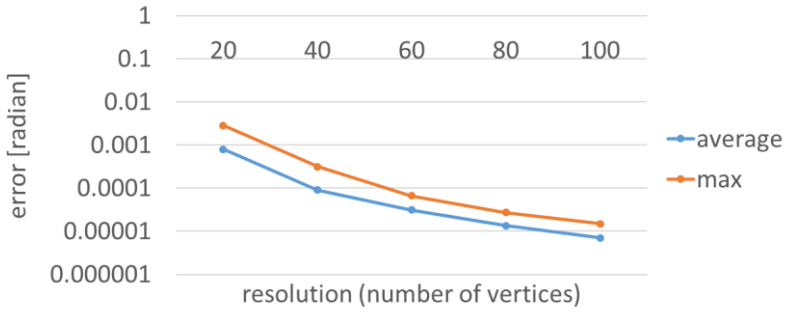


Figure 6-10 Graph of Table 6-3.

Table 6-4 Evaluation of developability by flatness of quads.

resolution	20	40	60	80	100
average [mm]	0.157034	0.020015	0.005966	0.002573	0.001314
maximum [mm]	0.655575	0.081198	0.025226	0.011502	0.006272

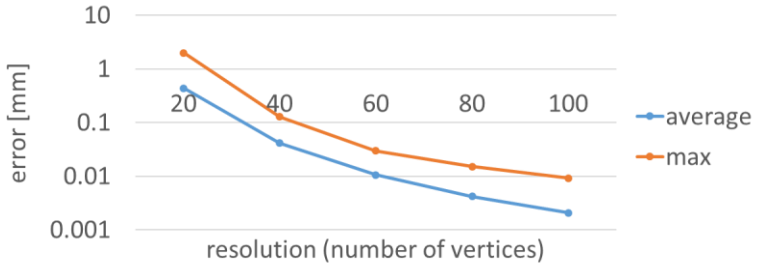


Figure 6-11 Graph of Table 6-4.

6.3.2. Comparison with real paper

The results of the visual comparison are shown in Figure 6-12. The papers in the photos were printed as the 2D crease pattern data generated by our system. It is folded and aligned manually to be in similar shape with the 3D model. Comparing the photos and the 3D polygon model, the model seems to be well representing the real paper.

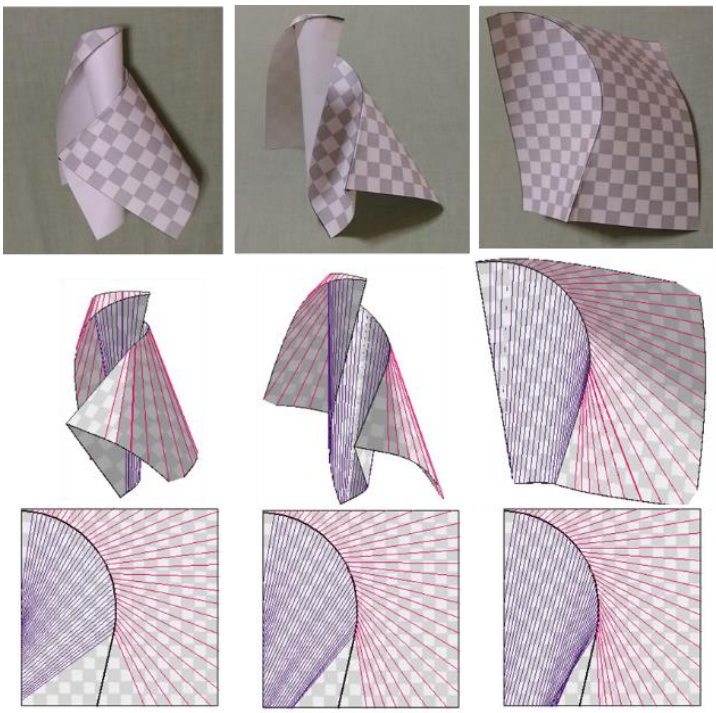


Figure 6-12 The visual comparison. Top row: photos of real paper. Middle row: 3D polygon model. Bottom row: 2D crease pattern with rulings.

6.3.3. Applicability of the method

One of the advantages of this method to generate a 3D polygon model of the curved folding is that the developability of the shape is guaranteed by the equations (2.7) - (2.9), though having some deformation caused by the discretization. The transition of the rulings is also supported by the equations, by recalculating the geometry directly from the input without an iteration process. This ensures the fast calculation of the shapes so the user can edit the shape interactively, with the result shown instantly. Another advantage is that the G2 continuity of the crease curve is guaranteed by the smooth interpolation of the input parameters, as explained in Subsection 6.1.2 and Figure 6-3, because the curvature, the second-order derivative of the curve, is continuous along the crease curve. Consequently, the Frenet-Serret frame is defined on any point of the reconstructed curve except for the two ends, enabling the calculation of the rulings using the equations in Chapter 2. The continuity of the parameters also makes the transition of the ruling directions along the curve to be smooth, and the generated polygon shape well represents a smooth curved surface.

On the other hand, our method has some disadvantages such as the usability, or the method of editing, and the lack of constraints to keep the model in an existable shape. As for the usability, the method of adjusting the parameters on the curved crease is generally not so intuitive. Even with the instant feedback, most users need to go through some trial and error to obtain the desired shape. The direct manipulation of the paper surface, which our method does not support currently, would be much more robust and easy input method for the users. By the lack of constraints, the parameters may be changed by the user to generate a 3D shape which does not exist in the real world, such as the rulings crossing or the self-collision of the faces. The method ensures the developability only under the condition that there are no rulings crossing. To avoid such cases, the visual check of the user is necessary. These issues are left as the future work of this theses.

With the method introduced in this chapter, the user is able to design a curved folded paper with one curved crease, by some user manipulation and the visual check. Simulation of folding motion is carried out using the same algorithm. The method meets our goal of visualizing the ruling transition and the shape deformation of the paper during the folding motion, but the target is limited to the curved folding with one curved crease. After explaining how to cope with the inflection point of the crease curve in the next chapter, some methods to work with multiple creases are introduced in Chapter 8 and 9.

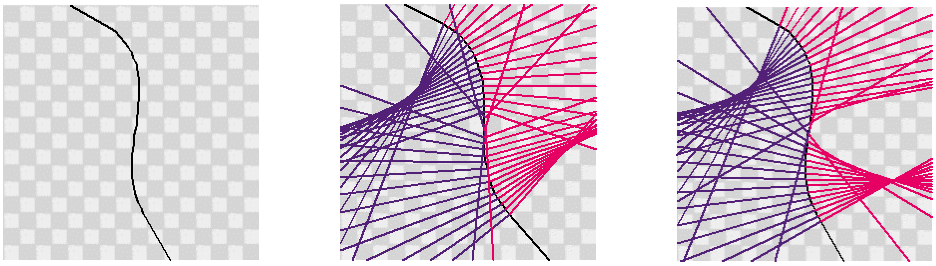
Chapter 7.
Rectification of curved crease at the inflection point

In this chapter, we discuss the method to handle the special case where the crease curve has an inflection point. This is an exceptional case of equations (2.8), (2.9), where the curvature of the crease curve becomes zero. In Section 7.1, the problems and the measures to avoid them is introduced. In Section 7.2, the generated shapes are evaluated in the same method as Chapter 6

7.1. Rulings at the inflection point

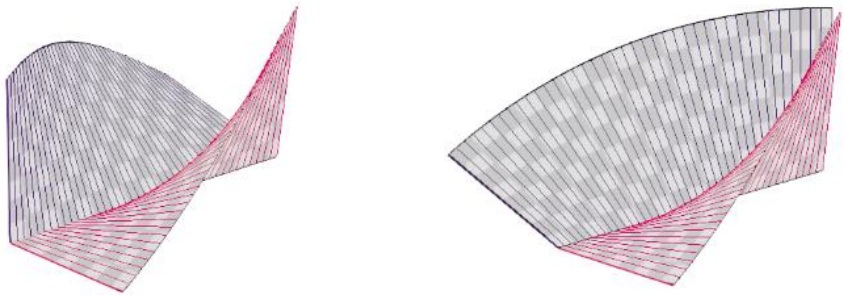
The mathematical framework explained in Chapter 2 assumes the 2D and 3D crease curve having non-zero curvature, as the curvature $k(s)$ is in the denominator of equations (2.8) and (2.9). In other words, the crease curve is not supposed to include an inflection point or to be a straight line in order to calculate the ruling directions. However, in practice, there is curved crease including an inflection point, and the users may want to design such a curve. One example of such curves is shown in Figure 7-1(a), where a curve changes from clockwise to counterclockwise as we follow the curve in the direction of the tangent vector. Since the curvature is positive in the clockwise curve, and negative in the counterclockwise curve, as described in Subsection 5.1.1, in between the two sections, the curvature is zero. At this point, as in equations (2.8) and (2.9), any non-zero value of torsion τ or difference of folding angle α' causes $\cot\beta_L$ and $\cot\beta_R$ to have infinitely large values, and β_L and β_R to be either 0 or π, which means that the ruling is parallel to the tangent vector, as shown in Figure 7-1(b),(c). On the other hand, observing paper in the real world, paper folded along straight crease line is never twisted, which means the torsion is zero, and the folding angle is constant throughout the crease. This is because a sheet of paper is a developable surface which can be flattened onto a plane without distortion. Non-zero torsion on the zero-curvature point means that the surface twists along a straight fold line, which requires the surface to stretch at the point far from the crease line (Figure 7-2(a)). A straight crease line with increasing or decreasing folding angle also has the same property (Figure 7-2(b)). These surfaces are ruled surfaces but are not developable.

In our system, however, the user input is not restricted to meet this property. Because the parameters of the curved crease are controlled at the control points, a parameter on one point of the curve is not easy to control directly by imposing constraints on the user input. Instead, to keep this restriction, we propose a rectification process for the torsion and the folding angle. The rectification methods are explained in the following subsections. Along with the rectification, the system also eliminates the rulings around the zero-curvature point, as shown in Figure 7-3, as it is not necessary to define rulings on a planar area, and they cannot be calculated using equations (2.8), (2.9).



(a) (b) (c)

Figure 7-1 Curved crease having a point of zero-curvature. (a) Crease curve. (b) Rulings with non-zero torsion. (c) Rulings with increasing folding angle.



(a) (b)

Figure 7-2 Examples of ruled and non-developable surfaces. (a) Surface twisting along straight fold line. (b) Surface with increasing folding angle on straight crease line.

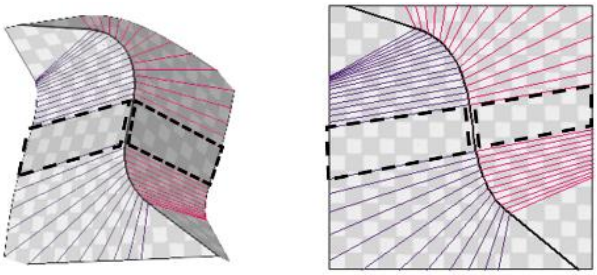


Figure 7-3 Elimination of rulings. Rulings are eliminated in the dotted area where the curvature of the crease is close to zero.

7.1.1. Rectification of folding angle

The rectification of the folding angle is summarized in Figure 7-4. It shows the 3D polygon model, the crease and the rulings mapped to 2D space, and the graphs of the folding angle along the curve, before and after the rectification. The 3D polygon models are constructed by the method described in Chapter 6. They are composed of quads placed between the 3D rulings whose directions are derived by equations (2.8) and (2.9) and their lengths are set to end at the paper edge. In case of the rulings crossing, the polygon model is not developable, with one point of a 2D space corresponding to more than one point on the 3D polygon mesh. As shown in the graph, the rectification is applied to the section (i) and to the section (ii) differently. In section (i), where the

curvature is smaller than a threshold, the folding angle is rectified to be constant. In section (ii), which is adjacent to section (i), the graph is interpolated by Bézier spline curve to smoothly meet the original point of the graph on both ends. Section (ii) range from the end of section (i) to one of the followings:

(a) a local maximum or minimum of the original folding angle,
(b) a midpoint of two inflection points, if there are more than one inflection point,
(c) the end of the curve.

Some examples of the above endpoints of section (ii) are illustrated in Figure 7-5, with section (ii) noted by the types of its range (a) – (c). This process may be regarded as removing the change of the folding angle along the curved crease from section (i) and shifts them to section (ii). In both sections, while the folding angle is modified as described, curvature remains unchanged. If the folding angle is constant, the curvature may either be zero or non-zero.

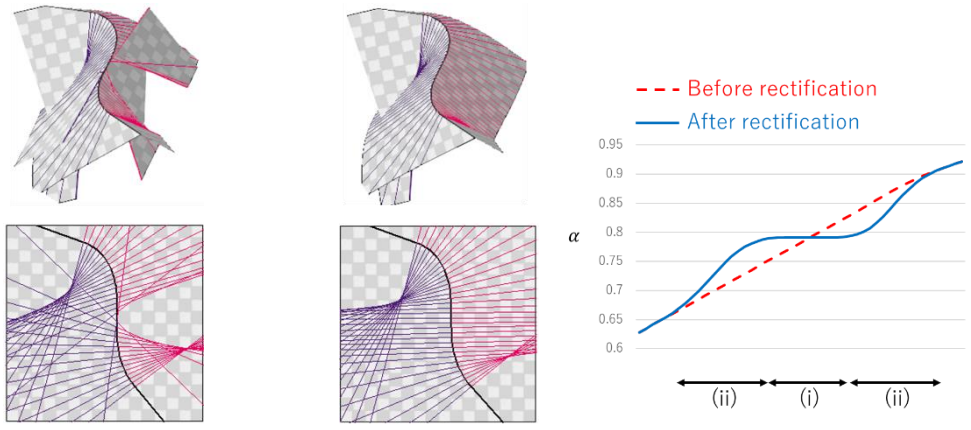


Figure 7-4 Rectification of folding angle. (a) 3D polygon model (top) and crease and rulings mapped to 2D (bottom) before the rectification. (b) After rectification. (c) Graph of folding angle before and after the rectification. In section (i) folding angle is rectified to be constant. In section (ii) it is interpolated by Bézier curve.

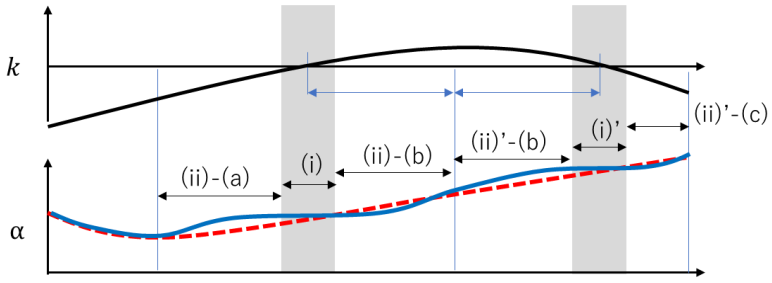
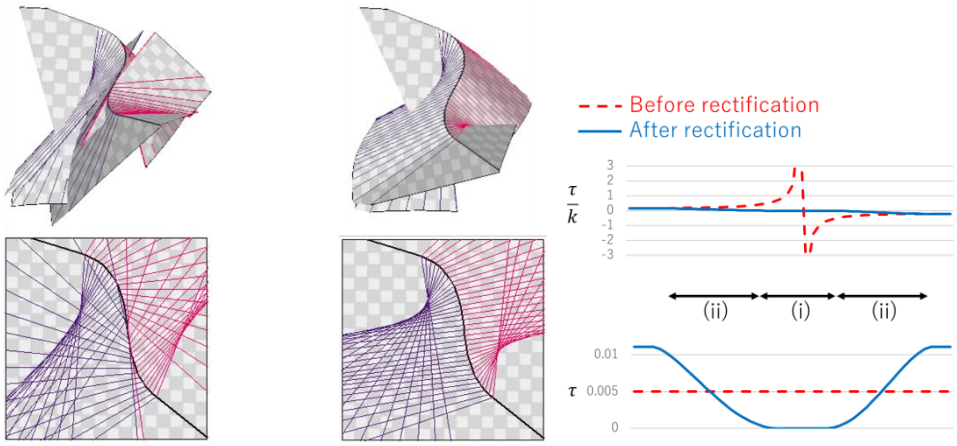


Figure 7-5 The range of section (ii) in three types. Top row shows curvature, red dotted graph shows folding angle before rectification, and blue graph shows folding angle after rectification.

7.1.2. Rectification of torsion

Figure 7-6 shows the rectification of torsion. The two graphs in Figure 7-6(c) are $\tau(s)/k(s)$ and $\tau(s)$. As given in equations (2.8) and (2.9), $\tau(s)/k(s)$ is proportional to $\cot\beta_L(s), \cot\beta_R(s)$ given constant folding angle α. This means that if this graph is smooth, then the ruling direction changes gradually along the curve, which generates smooth curved surfaces on its sides. In the rectification process, τ is set to zero in section (i), around zero-curvature point, which also sets $\tau(s)/k(s)$ to be zero. Then, in section (ii), $\tau(s)/k(s)$ is interpolated by Bézier spline curve to keep the graph smooth. The ranges of section (i) and (ii) are set in the same method as in folding angle rectification but using $\tau(s)/k(s)$ instead of the folding angle to obtain the end of section (ii). As in the rectification of the folding angle, curvature remains unchanged in both sections. If the torsion is zero, the curvature may be zero but not necessarily. $\tau(s)$ is then normalized so that the total amount of torsion on the curve is the same before and after the rectification, expecting the curve end to be close to the original point before the rectification. These processes change the shape of the 3D curve and the adjacent curved surface to make the whole surface developable, like a sheet of paper in the real world.



(a) (b) (c)

Figure 7-6 Rectification of torsion. (a) 3D polygon model (top) and crease and rulings mapped to 2D (bottom) before the rectification. (b) After rectification. (c) Graphs of $\tau(s)/k(s)$ (top) and graph of $\tau(s)$ (bottom) before and after the rectification. In section (i), torsion is set to be zero. In section (ii), it is interpolated by Bézier curve.

7.1.3. GUI for the rectification

Figure 7-7 shows the prototype GUI system with the widgets for the rectification on the control panel. The top part contains the checkbox to choose whether to apply the rectification process, for the folding angle, the torsion, and the elimination of the rulings. The rest of the control panel carries the input widgets to change the range of sections (i) and (ii), and to control the shape of the Bézier curve to interpolate the parameter graph. The interface is for trials and should be improved for better use of the application.

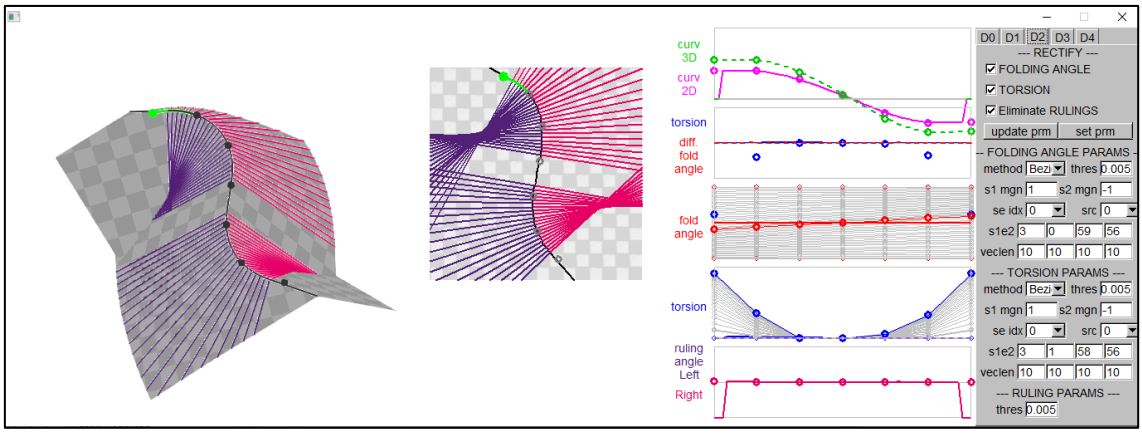


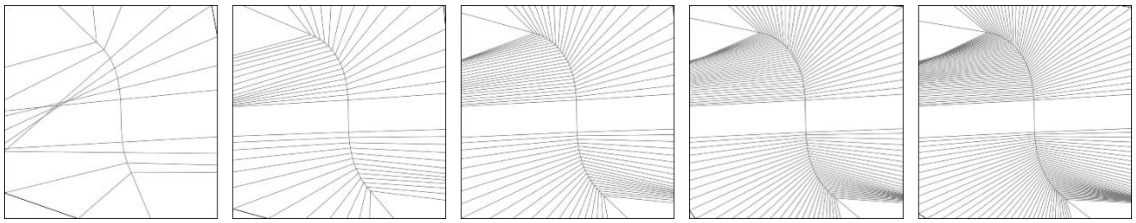
Figure 7-7 GUI for changing rectification parameters.

7.2. Results and discussions

As an evaluation of the method, a 3D polygon shape of a curved folded paper, containing zero-curvature point and rectified by the proposed method, is tested by the same criteria as in Chapter 6. The original parameters of this curved crease include non-zero torsion and increasing folding angles.

7.2.1. Evaluation of the developability

The 3D polygon models made by the crease curve in Figure 7-8 are evaluated by (a) the sum of the corner angles adjacent to a vertex and (b) the flatness of the quads. The results are shown in Table 7-1, Table 7-2, Figure 7-9, and Figure 7-10. For (a), the errors are below the tolerance of 8.7×10^{-4} radians for resolution 40 and higher. For (b), the displacement is below the tolerance of 0.1mm for the average in resolution 40 and for all data for 60 and higher. We could conclude that the generated 3D polygon surface is approximately developable for resolution of around 40 and higher. In the case of lowest resolution, crossing of the rulings was not resolved.



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

Figure 7-8 2D crease curve and ruling in different resolutions of discretization. The number of vertices on the crease are (a) 20, (b) 40, (c) 60, (d) 80, (e) 100.

Table 7-1 Evaluation of developability by sum of corner angles adjacent to a vertex.

resolution	20	40	60	80	100
average [radian]	8.0×10^{-4}	9.1×10^{-5}	3.1×10^{-5}	1.4×10^{-5}	7.1×10^{-6}
maximum [radian]	2.8×10^{-3}	3.1×10^{-4}	6.7×10^{-5}	2.7×10^{-5}	1.5×10^{-5}

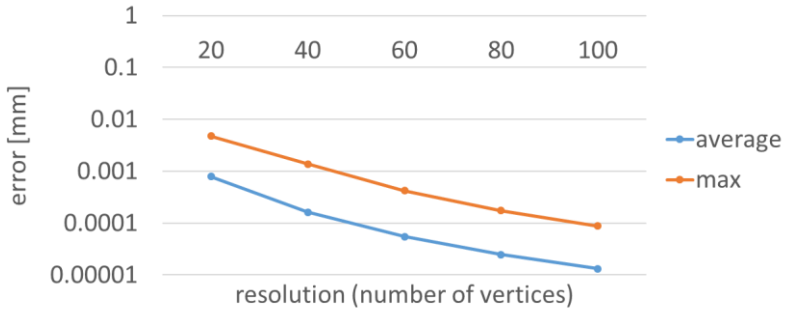


Figure 7-9 Graph of Table 7-1.

Table 7-2 Evaluation of developability by flatness of quads.

resolution	20	40	60	80	100
average [mm]	0.436465	0.041724	0.010618	0.004203	0.00208
maximum [mm]	1.979861	0.128637	0.029707	0.015156	0.009272

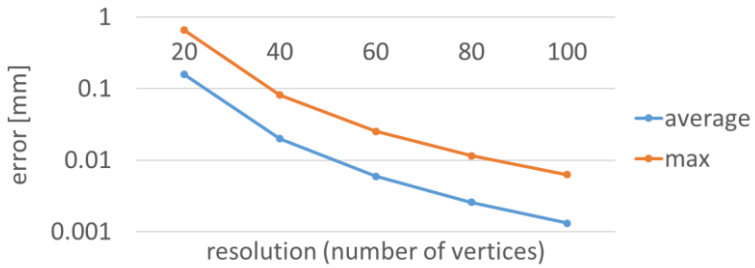


Figure 7-10 Graph of Table 7-2.

7.2.2. Comparison with real paper

The results of the visual comparison are shown in Figure 7-11. The papers in the photos were printed as the 2D crease pattern data generated by our system. It is folded and aligned manually to be in similar shape with the 3D model.

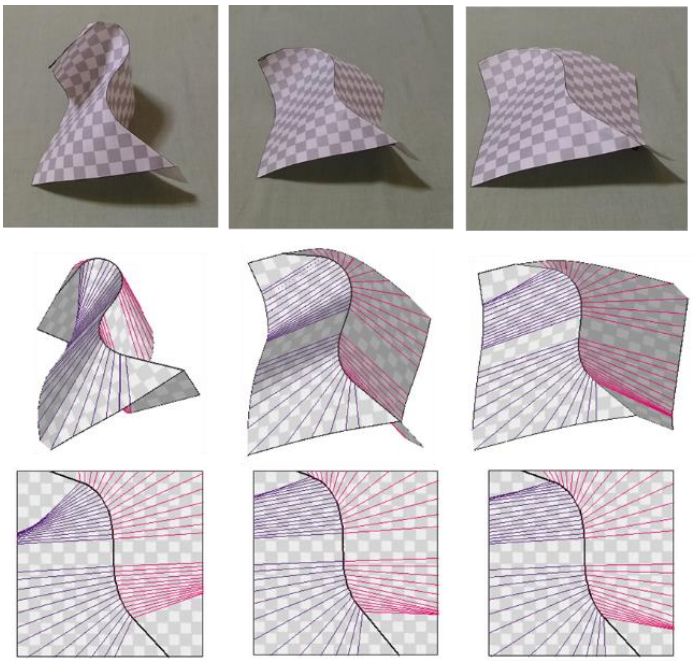


Figure 7-11 The visual comparison. Top row: photos of real paper. Middle row: 3D polygon model. Bottom row: 2D crease pattern with rulings.

7.2.3. Evaluations using 3D printer object

In addition to the above evaluation, the 3D polygon shape is reproduced as a solid object made by a 3D printer, as shown in Figure 7-12. By placing a sheet of metal between the two objects and a sheet of paper on the object with nearly no gaps, we confirmed that the accuracy of the developability is enough for reproducing a physical model with a metal sheet.

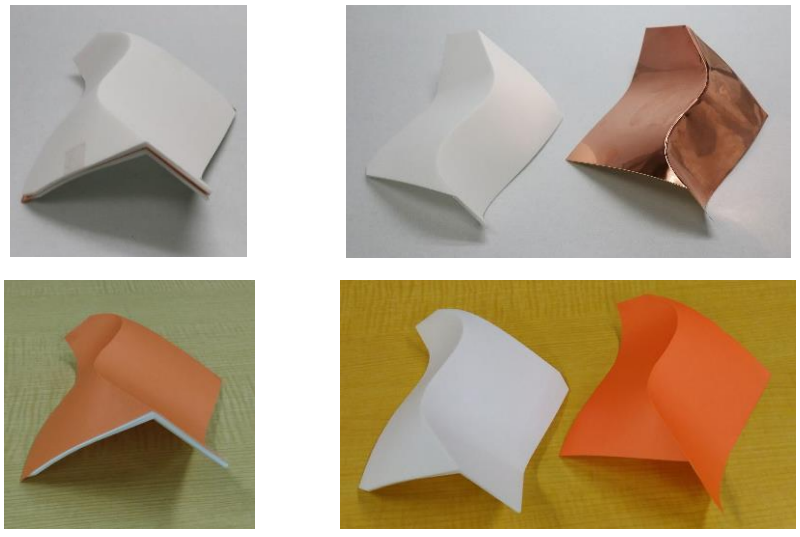


Figure 7-12 3D printed object with thin metal sheet (top) and paper (bottom).

Chapter 8.
Curved folding with multiple creases

In the previous chapters, we introduced a method to design a curved folding with only one curved crease on a piece of paper. In this chapter, we will introduce a method to add some other curved creases on the curved surfaces. We assume all curved creases end at the edge of the paper, with no branching, crossing or diminishing creases. As the first crease is designed by the method in the previous chapters, the 3D shapes of the adjacent developable surfaces are determined. Then the second curved crease is added on one of the two adjacent surfaces. This generates another developable surface by dividing the original surfaces along the new curved crease. The ruling directions and the 3D surface shape are then calculated according to the shapes of the original surface and the new curved crease. The method and the process are explained in detail in Section 8.1. Because the shapes of the surfaces are determined according to an adjacent curved crease, creases must be added one at a time, on a fixed curved surface. To have the rulings and the paper shape to be in the existable state, which means the rulings are on the surface not crossing to each other, the shapes of the newly added creases must be controlled properly. This is performed by the interactive user manipulation, with the prompt feedback of the 2D and 3D shapes generated by the input. Yet the task of adjusting the shape of the crease curve is often difficult for the added creases. This is because the user cannot edit the parameters directly like the first crease, as the curved surface on one side of the crease is already fixed. A small unintended shift of a control point affects the parameters on the curve, causing a large shift in ruling directions calculated from them. To resolve this problem and support the user input, we added some trial process of input restriction and optimization. By restricting the movement of the control point and to optimizing their positions based on some cost functions, the process performed well on some occasions but still has some room for improvement. In Section 8.2, we evaluated the generated 3D polygon data with multiple curved creases in the same method as in Chapter 6. Through the process of creating some other variations, we found it is difficult to add more than one or two creases on one side of the original curved surface.

8.1. Adding curved creases on a curved surface

The curved folded paper designed by this system contains one primary curved crease and some additional curved creases, as shown in Figure 8-1. The primary curved crease is the first crease to be placed on the paper and is designed by the method described in Chapter 6. Starting with one curved crease on a piece of paper, the user adds an additional crease on either side of the primary crease, adjust its shape, and simulate the folding motion through the GUI. In this section, the GUI and the procedure of the user manipulation are first described in Subsection 8.1.1. Then the calculation method to derive the 3D paper shape from the additional crease is explained in Subsection 8.1.2. Because the task of adjusting the shape of the crease curve is difficult in some cases,

some functions to support the user input is proposed and explained in Subsection 8.1.3, followed by another solution of trimming the paper in Subsection 8.1.4.

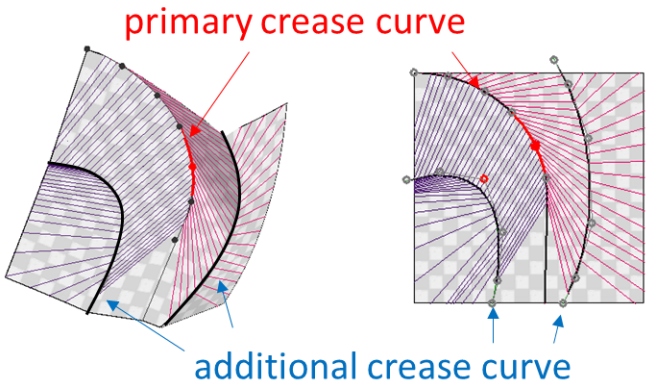


Figure 8-1 Curved folded paper with multiple curved creases.

8.1.1. User interface and the procedure of adding a crease

The procedure of adding a new crease is performed by the GUI, shown in Figure 8-2 with a new card tab on the control panel for adding curved creases, and follows the flowchart of user manipulation, shown in Figure 8-3.

The additional crease curve is initially input by the user as a free-form curve drawn on the 2D space, as shown in Figure 8-4(a). This is done in the curve drawing mode, activated by checking the checkbox FOLD on the control panel. After the initial shape is specified by the user, the curve is added as a new crease by the button ADD CREASE. As an internal process, the newly added crease is approximated by the B-spline curve so that the shape of the curve becomes smooth (Figure 8-4(b)). Then the shape of the new surface on the outer side of the additional crease is calculated and displayed (Figure 8-4(c)). Such internal process will be explained in detail in Subsection 8.1.2. At this step, the rulings are likely to be crossing. So, the user checks the result and adjusts the curve shape to resolve the crossing, if necessary (Figure 8-4(d)). This is done by moving the control points of the B-spline curve in 2D space. The user chooses one control point by clicking on the point shown in the 2D pane and then moves the position of the control point by the mouse drag. As the control point is moved, the B-spline curve and the paper shape is updated promptly so that the user can check the result and move the control point interactively. This manipulation, choosing a control point and moving it, is carried out until the shape of the additional curved crease and the paper is in the intended shape, with no rulings crossing. Note that the "control points" of the additional crease are different from the ones of the primary curved crease, placed evenly on the crease curve. The graphs of the parameters are also shown for the additional curved crease by specifying the corresponding crease by the scroll bar CREASE INDEX.

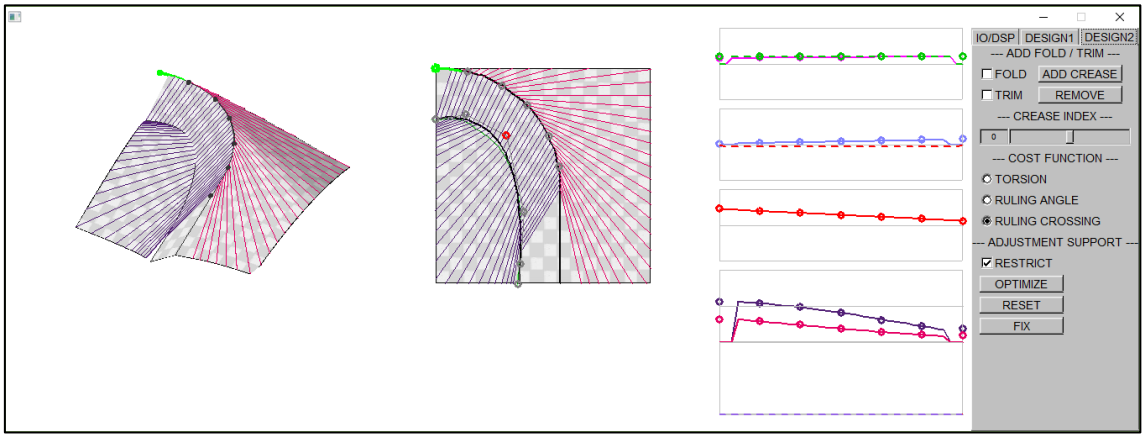


Figure 8-2 GUI for additional curved crease.

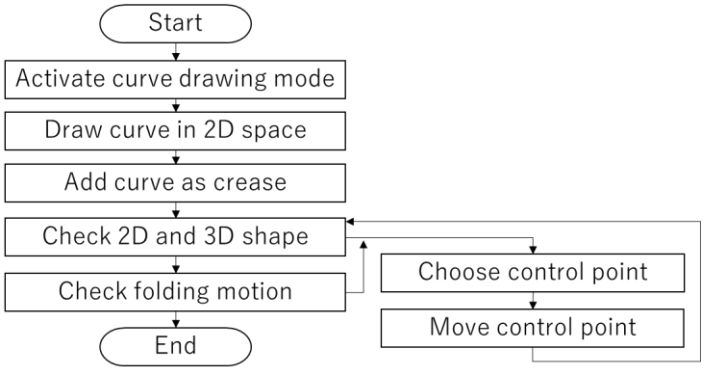
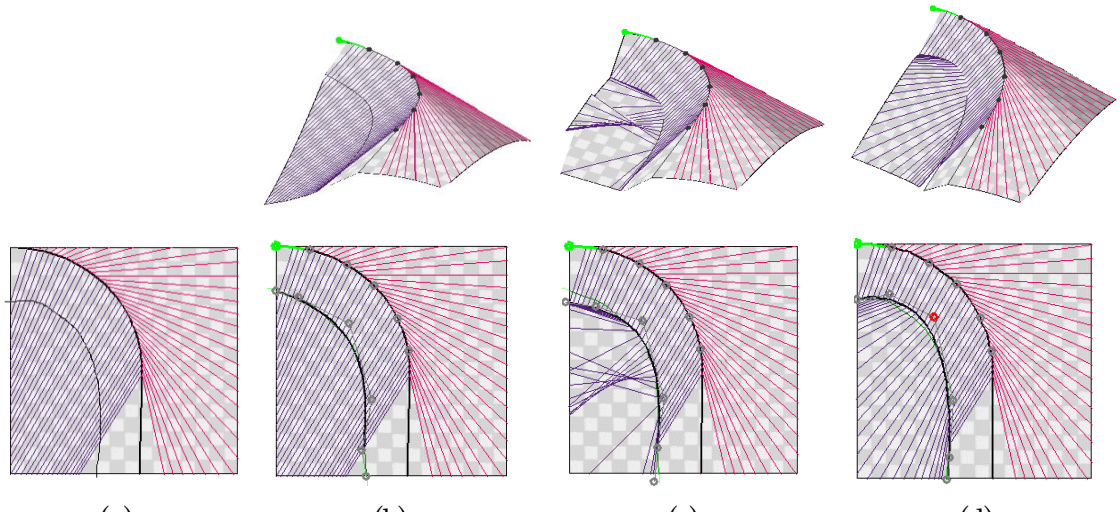


Figure 8-3 Iteration of user input for additional crease.



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

Figure 8-4 Procedure for adding a curve. (a) Free-form curve drawn by the user. (b) B-spline curve approximating the free-form curve. (c) Additional curved crease in initial state. (d) Additional curved crease after user adjustment.

By this procedure, the paper with multiple creases is created. But sometimes the task of adjusting the curve shape may be difficult. Because all other elements are derived from the shape of the crease curve on the 2D space, including the torsion, the curvature, the folding angle, and the vectors in the Frenet frame, a small unintended shift of a control point could affect the shape and the parameters of the additional curved crease, causing a large shift in the ruling directions calculated from them. To support the user input to adjust the curve shape effectively, the system is equipped with some functions of input restriction and the optimization for trial use. Three types of input modes are listed below.

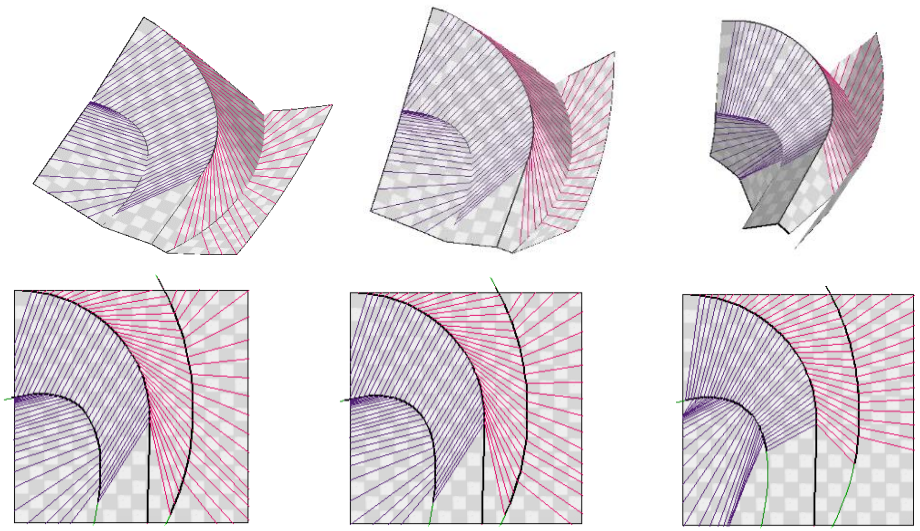
(i) The user edits the control points with no system support.
The user can move the control points on 2D space freely with no restriction. An inappropriate movement could make the state worse and cause the rulings to be crossing each other, but this may be good enough for an experienced user.

(ii) The user edits the control points with system support to encourage a better state.
The user can move the control points with some restrictions on the movement by checking the checkbox RESTRICT. The restriction is based on the cost functions chosen by the radio button under COST FUNCTION. The control points are moved according to the user's mouse drag only if the new position has a smaller cost than the previous state. The detail of each cost function will be explained in Subsection 8.1.3.

(iii) The optimization process based on the cost function.
The optimization process of an additional curved crease is executed by the button OPTIMIZE. In this process, each control point of the additional crease is shifted by a small amount in different directions, evaluated by the cost function, and the position with the smallest cost is adopted. This process is applied to all control points of the curve. If there is more than one additional crease, the crease to be optimized is specified by the scroll bar CREASE INDEX. The same cost function is used in the input mode (ii).

After the design and the adjustment of the curved fold, the folding motion of the paper is simulated. Figure 8-5 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. As in the generation of the folding motion in Chapter 6, the folding angle of the primary crease curve is changed linearly between zero, the flat state, and some value smaller than $\pi/2$, the completely folded state. The torsion is also changed between the original state and the flat state. On additional crease curves, the 3D curves, the folding angles, and the rulings on the other side are recalculated depending on the shape of the surface curved by the primary crease, with no direct control of the parameters by the user or the system. Given appropriate additional creases by the user, the folding and unfolding motion with ruling transition is simulated successfully in some range of folding angles. The crossing of the rulings may occur at some point of the motion, as a result of the ruling transition. In Figure 8-5(c), there are a gap on the additional crease on the 3D model where the crease is extended to the edge of the paper as a straight line. It is assumed to be caused by the difference in the

number of the rulings on the left and right sides of the additional crease. Because the rulings are derived from to the curvatures and the torsions calculated by equations (6.1) to (6.5) using consecutive three vertices, on the surface on the outer side, some rulings near the curve ends must be omitted. This causes a small difference in the orientations of the flat faces made by the extended straight line on the left and right sides, making a gap between the surfaces. The solution to this problem is to place the crease curve so that the ends of all crease curves are located outside the paper boundary. In the manipulation, the user may put more than one additional curve on either side of the primary curve but needs to make sure that there is no intersection of crease curves with the support of the system.



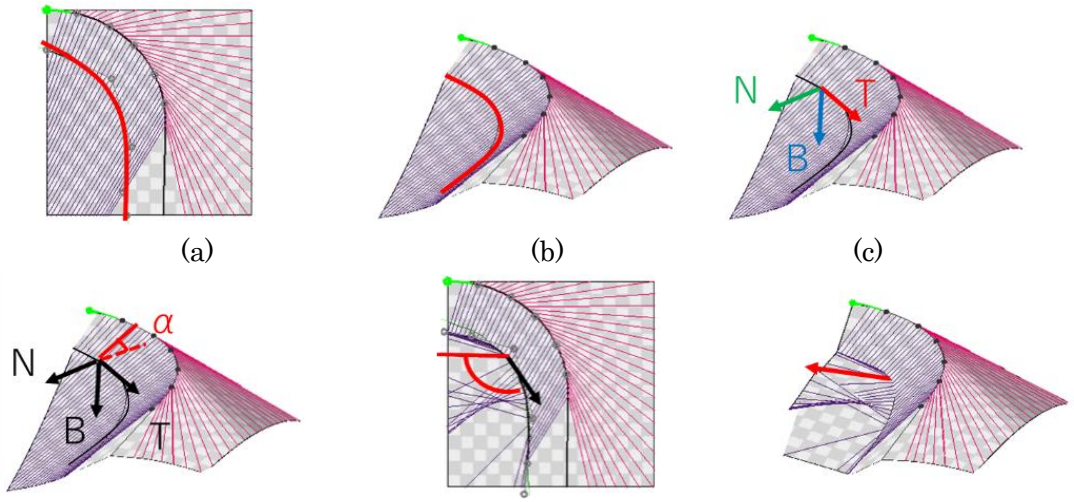
(a) (b) (c)

Figure 8-5 Simulation of folding motion. Folding angle of the primary curve is (a) 7 to 26 degrees, (b) 11 to 40 degrees, (c) 38 to 57 degrees.

8.1.2. Calculation of additional crease

Now, we will explain the internal process using the denotation $\{*^{Rj}{}_{i}\}$, indicating the elements of the j-th additional curved crease on the right side of the primary crease curve, for example. After the free-form curve is input as a new crease, the curve is smoothened by being approximated as a B-spline curve of degree-three, with six control points and the knot vector {0,0,0,0,1,2,3,3,3,3}. It is then discretized as a sequence of vertices in 2D space $\{\mathbf{X}_{2D}{}^{Rj}{}_{i}\}$ by plotting the intersections of the B-spline curve and the rulings stemming out from the primary curved crease (Figure 8-6(a)). By projecting the 2D vertices onto the 3D ruling vectors $\{\boldsymbol{r}_{\boldsymbol{R}_i}\}$, a 3D space curve of the additional curved crease $\{\mathbf{X}^{Rj}{}_{i}\}$ is obtained (Figure 8-6(b)). 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 equations (6.1)-(6.5) (Figure 8-6(c)). The folding angles $\{\alpha^{Rj}{}_{i}\}$ are calculated as the angle between the normal vectors $\{\mathbf{N}^{Rj}{}_{i}\}$ and the rulings

$\{\boldsymbol{r_L}^{Rj}{}_{\boldsymbol{i}}\}$ projected to the normal plane of the crease curve(Figure 8-6(d)). The ruling $\boldsymbol{r_L}^{Rj}{}_{\boldsymbol{i}}$ is obtained as $-\boldsymbol{r}_{R_i}$ because the ruling on the left side of the additional crease curve is identical to the rulings on the right side of the primary crease curve in the opposite orientation. At last, the ruling angles $\{\beta_R{}^{Rj}{}_{\boldsymbol{i}}\}$ on the new surface on the other side of the additional crease curve is calculated using equations (2.8) and (2.9) (Figure 8-6(e)), and the 3D ruling vectors $\{\boldsymbol{r_R}^{Rj}{}_{\boldsymbol{i}}\}$ by equations (2.10) and (2.11) (Figure 8-6(f)). In this step, the new rulings are likely to have crossings, 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 curve closest to the primary crease.



(a) (b) (c)

(d) (e) (f)

Figure 8-6 Process of calculating additional curved crease and the paper shape. (a) 2D curved crease $\{\mathbf{X}_{2D}{}^{Rj}{}_{\boldsymbol{i}}\}$. (b) 3D curved crease $\{\mathbf{X}^{Rj}{}_{\boldsymbol{i}}\}$. (c) Frenet-frame of additional curved crease. (d) Folding angle of additional curved crease. (e) Ruling angle $\{\beta_R{}^{Rj}{}_{\boldsymbol{i}}\}$ of 2D curved crease. (f) Ruling vector $\{\boldsymbol{r_R}^{Rj}{}_{\boldsymbol{i}}\}$ of the 3D curved crease.

8.1.3. Cost functions for input restriction and optimization

For the input mode (ii) and (iii) in Subsection 8.1.1, the input restriction and the optimization are implemented based on the cost functions. Our system provides three types of cost functions listed below, designed empirically for trial.

(a) The total area of the rulings crossing.
It is obvious that the rulings crossing in the 2D crease pattern do not make a good 3D polygon model. To improve such an undesirable state, we set the cost function to include the total area on the 2D crease pattern where the projection to the 3D space is

non-injective due to the ruling crossing. With si and ei indicating the indices of starting vertex and the ending vertex of the curved crease respectively,

$$Cost_A = \sum_{i=si}^{i<ei-1} \{Area\ of\ \triangle \boldsymbol{I}_{i,i+1}\boldsymbol{Y}_i\boldsymbol{Y}_{i+1}\}, \qquad (7.1)$$

where $\boldsymbol{I}_{i,i+1}$ is the intersecting point of two rulings of vertices $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$, and $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_{i+1}$ are the ends of the rulings for $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i+1}$. One area between the rulings crossing is shown in Figure 8-7.

(b) The difference between torsions.

Because the equation (7.1) is zero for the curved crease with no rulings crossing, we need some other factors to control the curve shape after the crossing of the rulings are resolved. One idea is that two curved creases placed side by side should have similar 3D shapes of the curve. The cost function is composed of the difference of torsion between the primary crease curve and the additional crease curve,

$$Cost_\tau = \frac{1}{ei - si}\sum_{i=si}^{i<ei} \left|\tau_i - \tau^{Rj}{}_i\right|. \qquad (7.2)$$

(c) The difference between the left and right ruling angles.

Another characteristic preferable for the curved crease is some type of stability. One simple index to measure this is the difference of ruling angles on the left and right side of the crease,

$$Cost_\beta = \frac{1}{ei - si}\sum_{i=si}^{i<ei} \left|\beta_L{}^{Rj}{}_i - \beta_R{}^{Rj}{}_i\right|. \qquad (7.3)$$

From equations (2.8) and (2.9), the ruling angles are the same for constant folding angle and differ if it is increasing or decreasing along the curved crease.

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 lowest cost is managed and updated separately for each cost function.

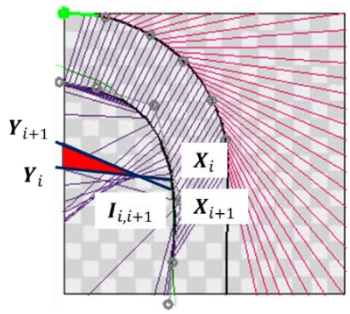
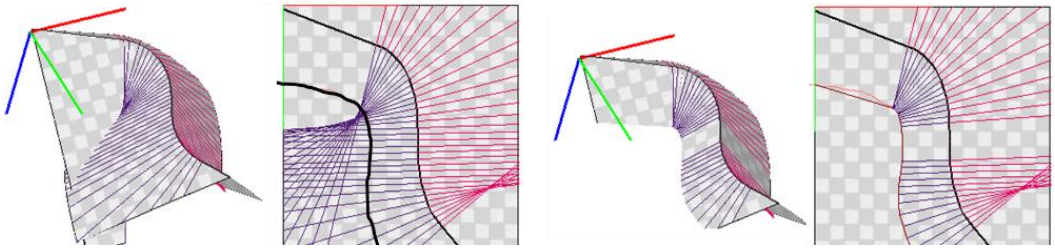


Figure 8-7 Non-injective area caused by rulings crossing.

8.1.4. Trimming the paper

In our GUI system, the user can trim the paper by a free-form curve input by the mouse drag, as shown in Figure 8-8. It is the same procedure as adding a new curved crease on a curved surface, except choosing the checkbox TRIM instead of FOLD in the initial step. This operation can eliminate the undesirable areas of the paper, such as crossings of the rulings.



(a) (b)

Figure 8-8 Trimming the paper. (a) Free-form curve input by the user. (b) The paper trimmed on the user input curve.

8.2. Result and discussion

A 3D polygon model generated using this prototype GUI system, shown in Figure 8-5, is tested by the evaluation method described in Chapter 6. The polygon model contains three curved creases: one primary crease, one additional crease on the left side of the primary crease, and another additional crease on the right side.

8.2.1. Evaluation of the developability

For the two additional creases, the indices for the developability, which are (a) the sum of the corner angles adjacent to a vertex and (b) the flatness of the quads, are summarized in Table 8-1 and Table 8-2. For (a), the error was below the tolerance of 0.05 degrees. For (b), the displacement was below the tolerance of 0.1mm in most of the cases, but for some quads on the outside of the additional crease, the displacement was above the tolerance.

Table 8-1 Evaluation of developability by the sum of corner angles adjacent to a vertex.

Folding angle	(a) 7 to 26 degrees		(b) 11 to 40 degrees		(c) 38 to 57 degrees	
crease	Left	Right	Left	Right	Left	Right
average [degrees]	0.000163	0.000140	0.000440	0.000389	0.000387	0.000419
maximum [degrees]	0.000401	0.000516	0.00120	0.00115	0.00132	0.00178

Table 8-2 Evaluation of developability by the flatness of quads.

Folding angle	(a) 7 to 26 degrees		(b) 11 to 40 degrees		(c) 38 to 57 degrees	
quads	inside	outside	inside	outside	inside	outside
average [mm]	0.00056	0.0255	0.00074	0.0080	0.00098	0.025
maximum [mm]	0.0033	**0.87**	0.0022	0.040	0.019	**0.71**

8.2.2. Comparison with real paper

For the comparison with the real paper, two samples of crease patterns are printed and folded. The shapes of the real paper are adjusted to be in similar shape with the 3D polygon model with a piece of wire on the back of the paper to fix the 3D shape of the primary curved crease. The results are shown in Figure 8-9. Their appearance seems acceptable although there seems to be some distortion near the straight segment of the additional crease in Figure 8-9(j) and (l), which the 3D polygon models contain some gaps. There are also differences such as the viewpoint, not strictly controlled.

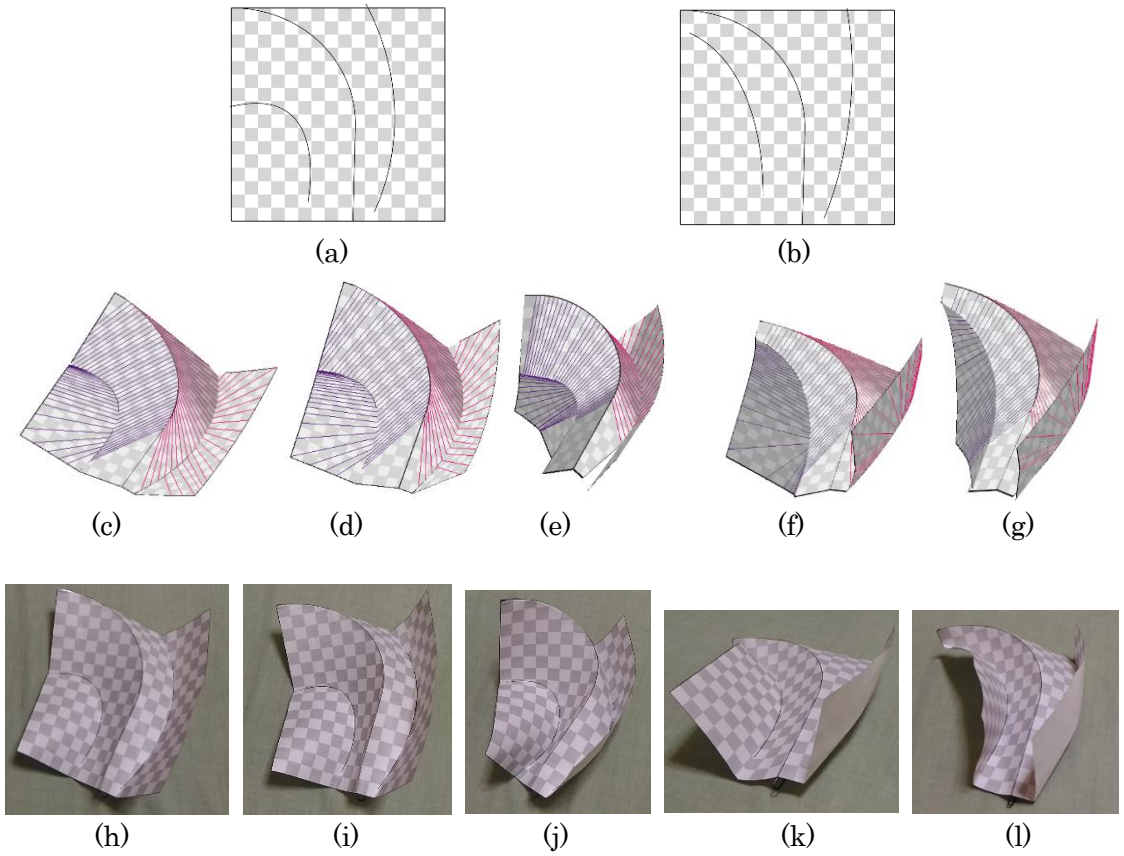


Figure 8-9 Visual comparison with real paper. (a), (b): Crease patterns printed. (c)-(e): 3D polygon models of crease pattern (a), in different folding angles, generated by our GUI system. (f), (g): 3D polygon models of crease pattern (b). (h)-(j): Photos of real paper folded by crease pattern (a), corresponding to the 3D model (c)-(e). (k), (l): Photos of real paper folded by crease pattern (b), corresponding to the 3D model (f), (g).

8.2.3. Other examples

Other examples designed by the system are shown in Figure 8-10, with its folded shape generated as a result of the folding simulation. This time, the faces adjacent to the ends of the curved creases extended as straight lines are omitted. For the samples in the top to the third row, the result seems to be acceptable. But for the last sample, the rulings start to cross each other on the newly generated curved surface quickly after the folding starts.

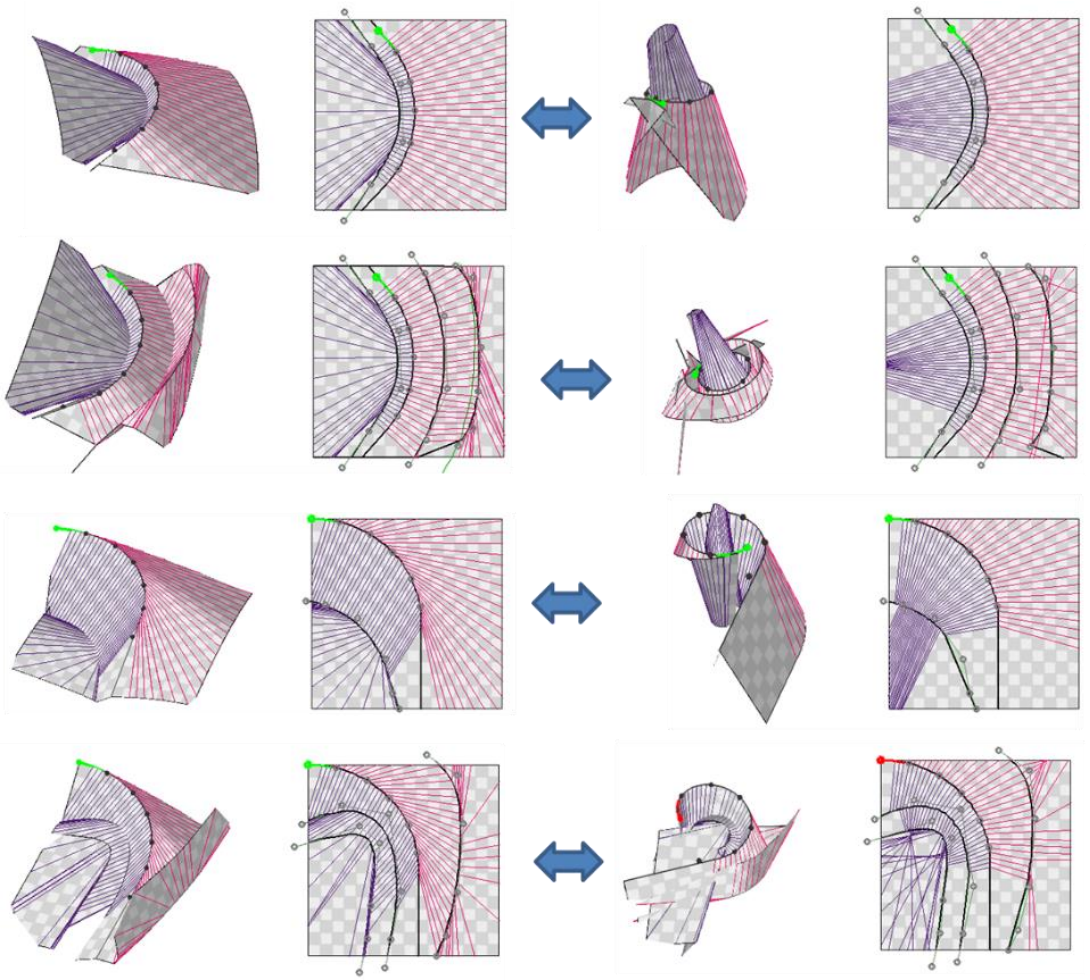


Figure 8-10 Other examples of 3D polygon model and the crease pattern with multiple curved creases. Left: Designed shape. Right: Folded state.

Chapter 9.
Curved folding with rotational symmetry

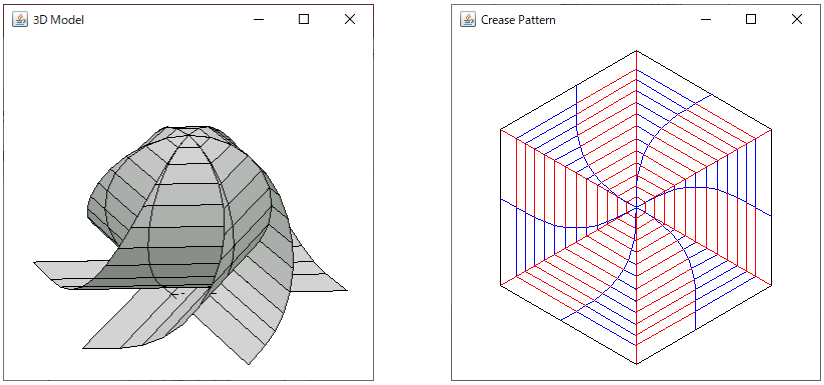
This chapter proposes a method to generate the folding motion of the rotationally symmetric curved folding with user manipulation. Rotationally symmetric curved folding is a type of origami that has curved creases placed in rotational symmetry, and the folded sheet of paper forms a three-dimensional shape, as shown in Figure 9-1. Various paper artworks of this type have been created by many artists and hobbyists, and software to design the 3D shape and the crease pattern has been developed [Mitani_09]. Although the crease patterns and their final folded 3D shapes are well known, their folding motions, or the paper shapes in between the flat unfolded state and the final folded state, are still not clear. Our system generates and visualizes the folding motion of some pre-defined models, with some user manipulation and the keyframe interpolation. The methods are explained in Section 9.1, and the result is shown in Section 9.2, with the discussion about folding a paper with a crease pattern, including some straight creases in Subsection 9.2.4.

(a) (b)

Figure 9-1 Example of rotationally symmetric curved folding from Flicker [Mitani]. (a) Sphere in cylinder type. (b) Sphere in disk type.

9.1. Method

In this section, we will explain the whole process by taking as an example the origami-sphere in disc type with six segments, which is a typical example of the rotationally symmetric curved folding designed by Mitani. The 3D model and the crease pattern designed by his software *ori-revo* [Mitani_09] is shown in Figure 9-2, and the photo of origami-sphere with eight segments is in Figure 9-1(b). It is composed of two types of creases: the curved creases in a valley-fold, or concave fold, that maintain the sphere shape with the adjacent curved surfaces, and the straight radial creases in a mountain-fold, or convex fold, that pull out the extra parts of the paper not on the sphere. The following subsections describe the method to generate the shape and the motion of the paper, followed by the explanations of the GUI system.

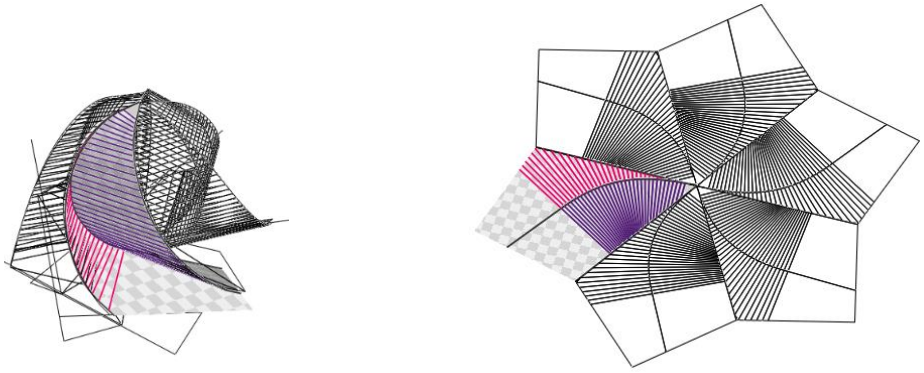


(a) (b)

Figure 9-2 *Origami-sphere* in disk type with six segments. (a) 3D model. (b) Crease pattern. Both are designed by *ori-revo* [Mitani_09].

9.1.1. Overall process

As a base of our method, we model the *origami-sphere* as a group of segments divided by the radial line creases, placed in rotational symmetry, as shown in Figure 9-3. Each segment contains only one curved crease. By editing one segment, the rest of the paper is cloned in accordance with the symmetric property of the crease pattern. Consequently, the curved folding with several creases is modeled based on the method proposed in Chapter 6.



(a) (b)

Figure 9-3 One segment of origami-sphere rendered with texture and rulings. (a) 3D model. (b) 2D crease pattern. Modeled by our GUI system.

As in the system described in the previous chapters, the system has a 3D model and a crease pattern. During the folding motion, the shape of the 3D model and the rulings in the 2D space changes continuously, while the crease pattern stays unchanged. Figure 9-4 shows the illustration of the folding motion generation. The top row describes the initial step with no user modification. Given the flat state and the final folded state to be the beginning and the end of the folding motion, two frames are set to be the initial keyframes. In the flat state, all creases have zero folding angles, i.e., the

3D model identical to the crease pattern. The final folded state is the complete form of the curved folding where, in this example, straight crease lines are folded completely, and the curved surfaces bent by the curved creases make a sphere. Initially, the intermediate frames are generated by linearly interpolating the two keyframes, as described in Subsection 9.1.3 (Figure 9-4 top row). As this generally causes large gaps between the segments, the user picks one frame, corrects it to make the gap smaller, and adds it as a new keyframe. The rest of the non-key frames are then re-interpolated (Figure 9-4 bottom row). This process is carried out until the user is satisfied with the output, where the segments are placed adjacently in all frames with no large gaps or self-collision. In this work, we assume that the paper shape is rotationally symmetric at all stages of the folding motion, though in the real world it may temporarily be asymmetric.

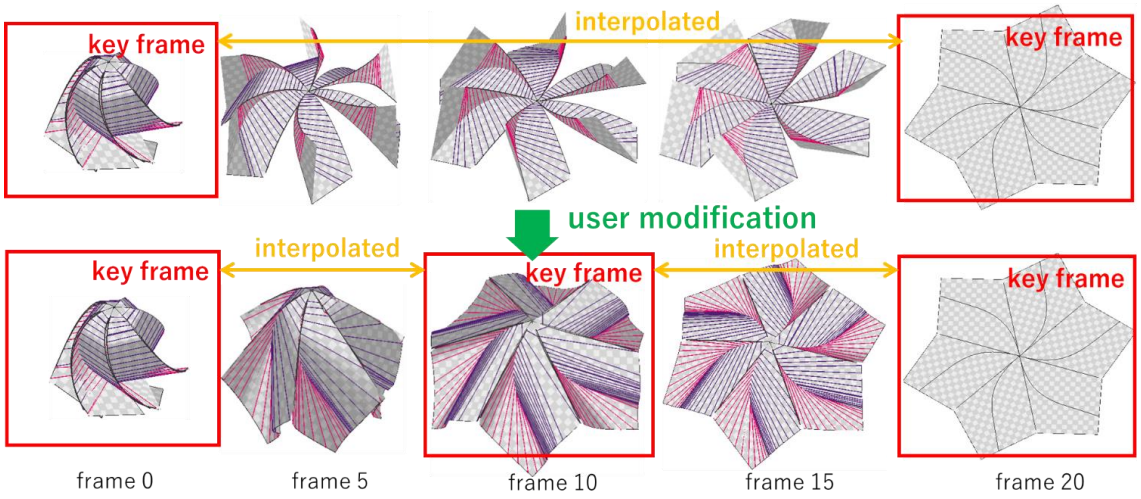


Figure 9-4 Process of folding motion generation. Frame 0 is the final folded state. Frame 20 is the flat state. One of the frames in the intermediate state is picked up, modified by the user, and set as a new keyframe. Other frames are interpolated from the adjacent keyframes.

9.1.2. Crease pattern generation

In the process of generating the folding motion, the crease pattern is set as the predefined data that may be designed by some other tools or methods. The whole crease pattern is composed of six identical segments placed side by side in rotational symmetry. As shown in Figure 9-5, the crease pattern of one segment consists of one curved crease in the center and two boundary creases on its sides, which are the boundaries between the adjacent segments. Theoretically, a segment may contain multiple curved creases that are not on the boundary, but our system supports only one crease for usability.

As in Chapter 6 and Figure 6-3, the curved crease is derived from 2D curvatures on the control points, sampled at equal intervals on the curve. They are interpolated throughout the curve, and the shape of the 2D crease curve is reconstructed from them.

The placements of two boundary creases are determined by the division number N of the rotationally symmetric design, which is equal to the number of the segments (Figure 9-6). Their shapes are defined by a constant value of 2D curvature k_{2D} (Figure 9-7). Whereas the *origami-sphere* has the boundary curves in a straight line shape, our system allows them to have a curved shape but defines them by only one value to make the implementation simple. The two congruent boundary creases are placed with a rotation of $2\pi/N$ so that, in the flat state, the segments are placed with no gap. Figure 9-3 and Figure 9-4 show an example with $N = 6$ and $k_{2D} = 0.0$, which means the boundary creases are the straight lines.

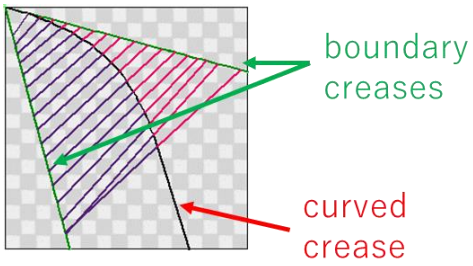
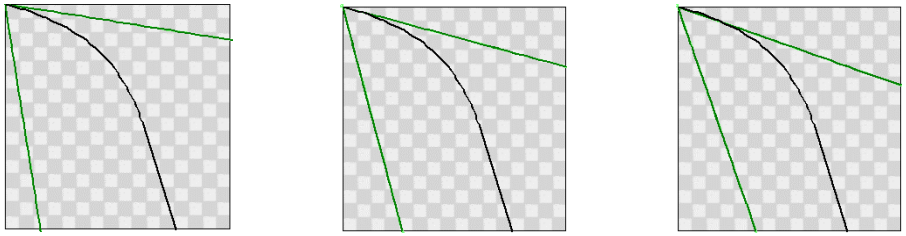
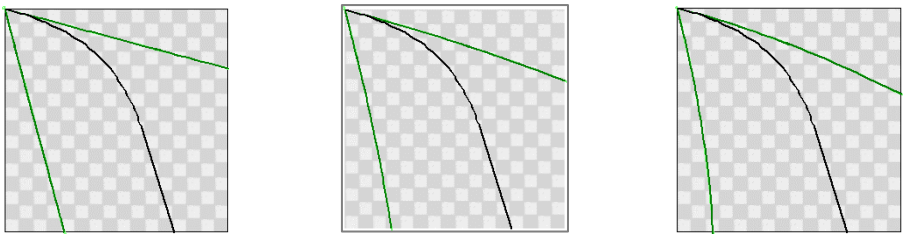


Figure 9-5 Crease pattern of one segment.



(a) (b) (c)

Figure 9-6 Placement of boundary crease. (a) $N = 5$ (b) $N = 6$ (c) $N = 7$



(a) (b) (c)

Figure 9-7 Shape of boundary crease. (a) $k_{2D} = 0.0$ (b) $k_{2D} = 0.0005$ (c) $k_{2D} = 0.001$

9.1.3. 3D model generation and interpolation

The process of 3D model generation and interactive modification is applied to the keyframes added by the user and to the final folded state. For the final folded state, the parameters of the curved crease should be calculated on other tool or software and given as an input, but if necessary, they may be refined on this system. For the keyframes, the 2D crease pattern is fixed, and the 3D shape of a segment is defined by

the torsion and the folding angle of the curved crease, following the case B in Subsection 2.3.2. The initial values of the torsion and the folding angle are given by interpolating the flat state and the final folded state. Then the user picks one frame, modifies the parameters, and set it as a new keyframe. The whole 3D model is then constructed with six segments placed in rotational symmetry with small gaps in between the adjacent boundary creases. The process of modifying or refining the 3D model is composed of two steps: (i) the 3D shape refinement of a segment, and (ii) the adjustment of the segment pose, or the 3D orientation of the segment.

In (i) the 3D shape refinement, the 3D shape refinement, the shape is controlled by the folding angle and the torsion of the curved crease. The parameters are modified by the user by changing the values on the control points, placed evenly on the curved crease. By the method described in Chapter 6, the rulings and the 3D shape of the curved folded surface is derived. An example of a modified segment shape is shown in Figure 9-8.

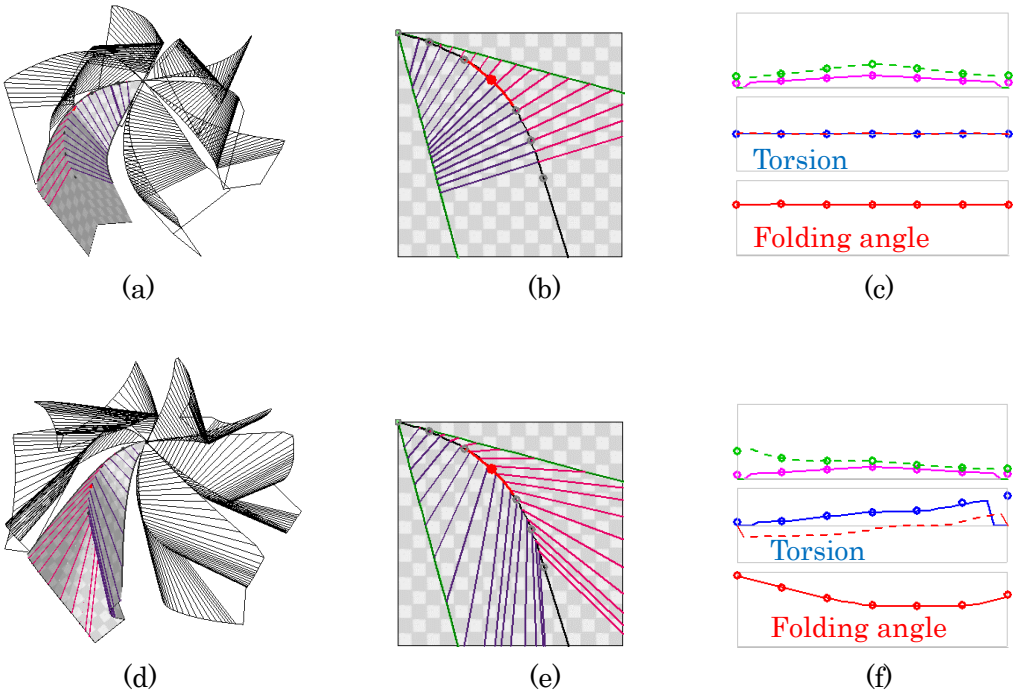
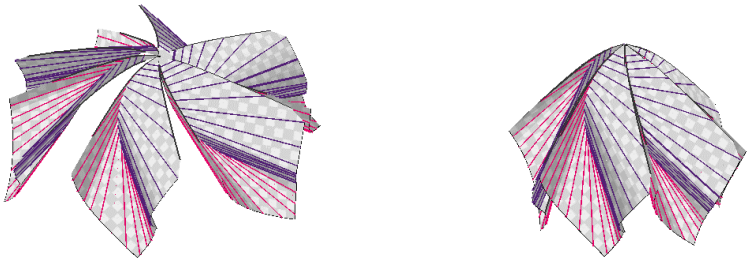


Figure 9-8 3D shape refinement of a segment. Top row: Before modification. Bottom row: After modification. (a), (d): 3D model. (b), (e): Crease and rulings mapped to 2D. (c), (f): Graphs of parameters.

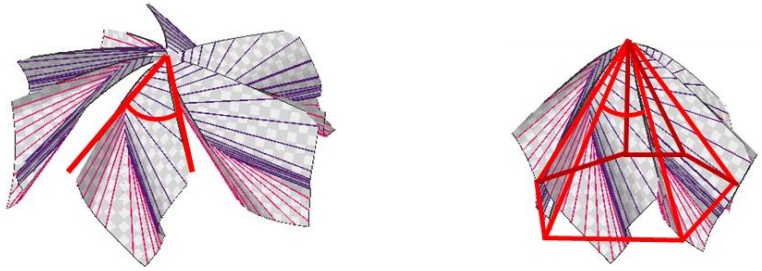
In (ii) the pose adjustment, the segment pose is adjusted through an aligning process to minimize the gaps between the boundary creases, or by user adjustment through the mouse drag interface, as shown in Figure 9-9. In the aligning process, segments are placed so that their boundary creases fit the lateral edges of an equilateral pyramid, as illustrated in Figure 9-10. The shape of the equilateral pyramid (Figure 9-10(b)) is derived from the number of segments N and the relative 3D orientations of a pair of boundary creases on a segment. That is, the base of the pyramid has the same number of edges as the number of the segments, and the angle between the adjacent lateral edges

of the pyramid is equal to the angles between the lines approximating the boundary crease in 3D space: passing through the center of the paper and its orientation calculated by the least square method using the sample points on the boundary crease (Figure 9-10(a)). In the user adjustment process, as the pose of one segment is adjusted with mouse drag, all other segments are also placed in rotational symmetry, or the i-th segment rotated $2\pi i/N$ around the vertical axis.



(a) (b)

Figure 9-9 Adjustment of segment pose. (a) Before adjustment. (b) After adjustment.



(a) (b)

Figure 9-10 Adjustment of segment pose. (a) Lines approximating segment boundaries. (b) Lateral edges of equilateral pyramid.

To integrate the six segments to be one sheet of paper, (i) the target segment shape must have the 3D shapes of the two boundary creases to be as congruent as possible and (ii) the segment pose is to be adjusted to minimize the gaps, without considerable self-collision. (i) and (ii) are improved alternately until the user is satisfied with the result by a visual check. After the keyframe is modified, other interpolated frames are updated by calculating the folding angles, the torsion, and the pose by the linear interpolation of the keyframes, followed by the same process as described above.

9.1.4. GUI of the prototype system

As in the GUI system in the previous chapters, the prototype GUI system for the folding motion generation is composed of four panes: the 3D model, the 2D crease of a segment, the parameters of the curved crease shown in the graphs, and the control panel, shown in Figure 9-11. The graphs of the folding angles and the torsion are modified to show the parameters in multiple keyframes. Their graphs are shown, with the current frame in thick colored lines, the keyframes in thin colored lines, and the other interpolated frames in thin gray lines. For the other parameters, only the current frame is depicted.

On the control panel, a card tab is added which carries widgets to design the rotational symmetric curved folding, such as scroll bars to choose the segment number N, to choose the current frame to be checked and modified, and to adjust the torsions and the folding angles of the current frame.

Figure 9-12 shows the flowchart of the user operation. After the 2D and 3D creases of the final folded state are input, the user first defines the boundary curve by the method described in Subsection 9.1.2. Then the user checks the generated shapes and modifies the 3D model if necessary, as described in Subsection 9.1.3. The segment shape is refined by changing the torsion or the folding angle through the widgets while checking the generated shape visually on the screen. The pose is aligned by choosing the button or adjusted by mouse drag on the 3D model. When the 3D model in the frame is satisfactory by the visual check, the user registers the frame as a new keyframe and checks the folding motion by showing each frame specified by the scroll bar. The user picks a frame and modifies it until the 3D models in all frames are satisfactory, under a smooth deformation in the folding and the unfolding motion. As well as the shape of the 3D models, the transition of rulings and the parameters can also be observed through the 2D crease pattern and the graphs of the parameters.

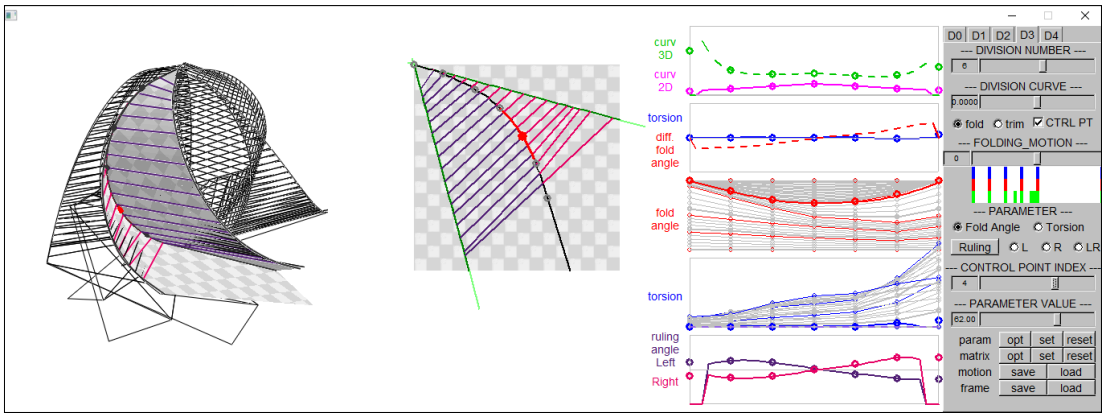


Figure 9-11 GUI of the prototype system.

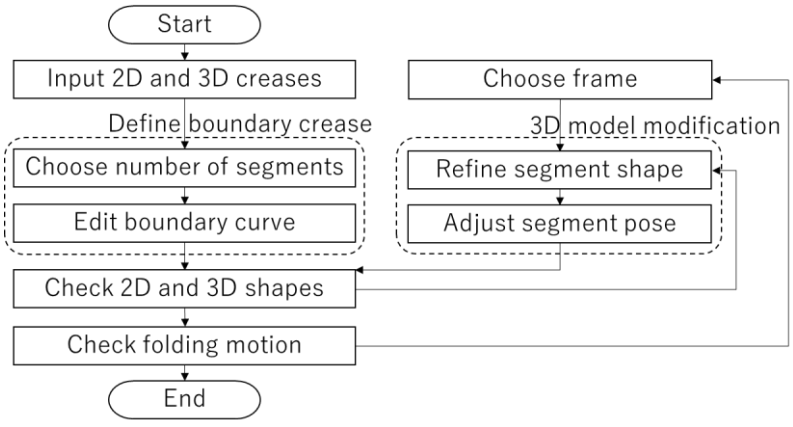


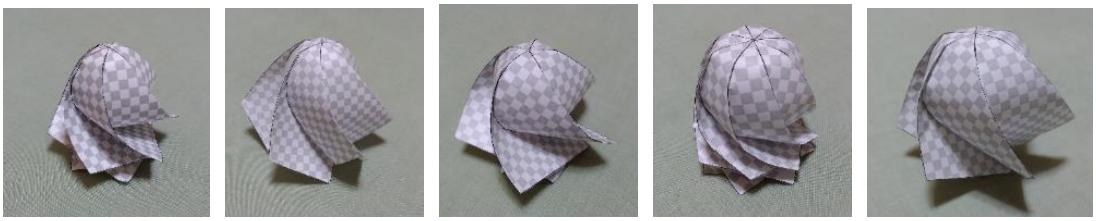
Figure 9-12 Flowchart of user operation.

9.2. Result and discussion

As an evaluation of our method, the 3D models of the rotational symmetric curved folding with the folding motion, generated by our prototype system, were evaluated in terms of (i) the developability of one segment, (ii) the connectivity between the segments, and (iii) a visual comparison with real paper. The examples used for the evaluation are listed in Table 9-1 and shown in Figure 9-13.

Table 9-1 3D models used for evaluation.

Model Name	N6	N6_k1	N5	N8	N6_skew
Number of segments N	6	6	5	8	6
Curvature of boundary crease k_{2D}	0.0	0.001	0.0	0.0	0.0
Shape	sphere	sphere	sphere	sphere	skewed sphere



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

Figure 9-13 Curved folding corresponding to the 3D models in Table 9-1. (a) N6, (b) N6_k1, (c) N5, (d) N8, and (e) N6_skew.

9.2.1. Developability of one segment

The developability of a segment was evaluated in the same method as in Chapter 6, (a) the sum of the corner angles adjacent to a vertex and (b) the flatness of the quads. The result is summarized in Table 9-2. For (b), the units are in millimeters while the crease pattern of the 3D model used for the evaluation has a radius of approximately 300 mm. The results show that for all items, the error is below the tolerance, and the segments are sufficiently developable for all 3D models. This is a result well expected from Chapter 6, as one segment is identical to a curved folded surface with one curved crease.

Table 9-2 Developability of segment. Units are radian for (a) and mm for (b).

Model Name	N6	N6_k1	N5	N8	N6_skew
(a) Average	3.5×10^{-5}	1.2×10^{-5}	4.2×10^{-5}	4.4×10^{-5}	6.2×10^{-5}
(a) Maximum	3.47×10^{-4}	1.12×10^{-4}	2.45×10^{-4}	3.93×10^{-4}	4.92×10^{-4}
(b) Average	7.19×10^{-3}	5.01×10^{-3}	7.1×10^{-3}	8.34×10^{-3}	1.09×10^{-2}
(b) Maximum	1.19×10^{-1}	3.85×10^{-2}	7.46×10^{-2}	1.08×10^{-1}	6.69×10^{-2}

9.2.2. Connectivity between the segments

The connectivity of the segments is calculated as the gap between the adjacent boundary creases in 3D space. To measure the gap between two curves, we first sample the reference points in an equal interval on one boundary crease. Then the distances from the reference points to the nearest edges on the other boundary crease are calculated. The average distance at all the sample points is evaluated as the gap. Because the segments are always placed in rotational symmetry in this system, the gaps are the same for all pairs of adjacent boundary creases. Figure 9-14 shows the gaps on each frame for the 3D models listed in Table 9-1, with frame 0 being the final folded state and frame 20 the flat state. The circles on the graph indicate the keyframes where both the shape and the pose of the segments are modified by the user. The squares indicate the frames where only the segment pose is modified. The rest of the points is the interpolated frames. The results show that, by the user manipulation, the gaps are made to be below 2.5 mm for most frames, with the 3D models whose radiuses of the crease patterns are approximately 300 mm. We expect these gaps are small enough to be covered by a small deformation, such as the sliding of the creases while processing with real paper. With more user effort, the gaps may be reduced.

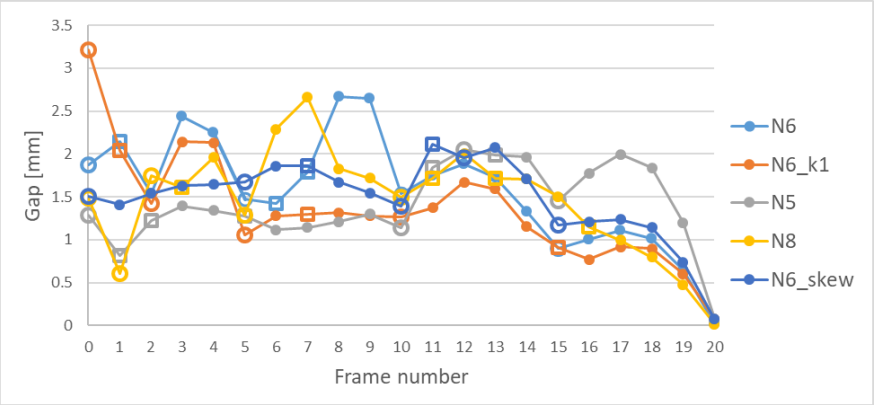
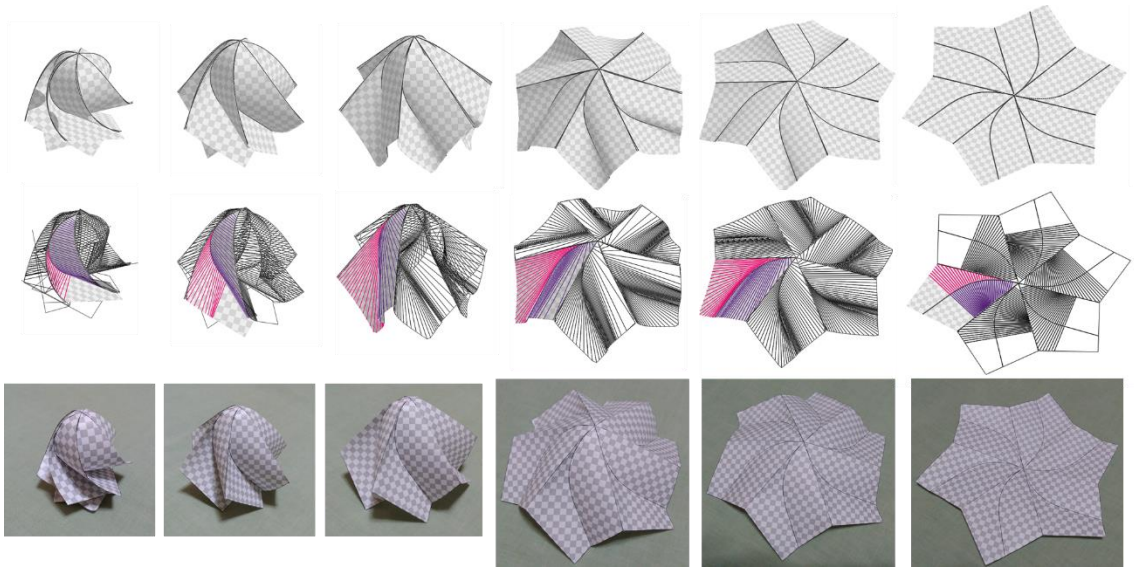


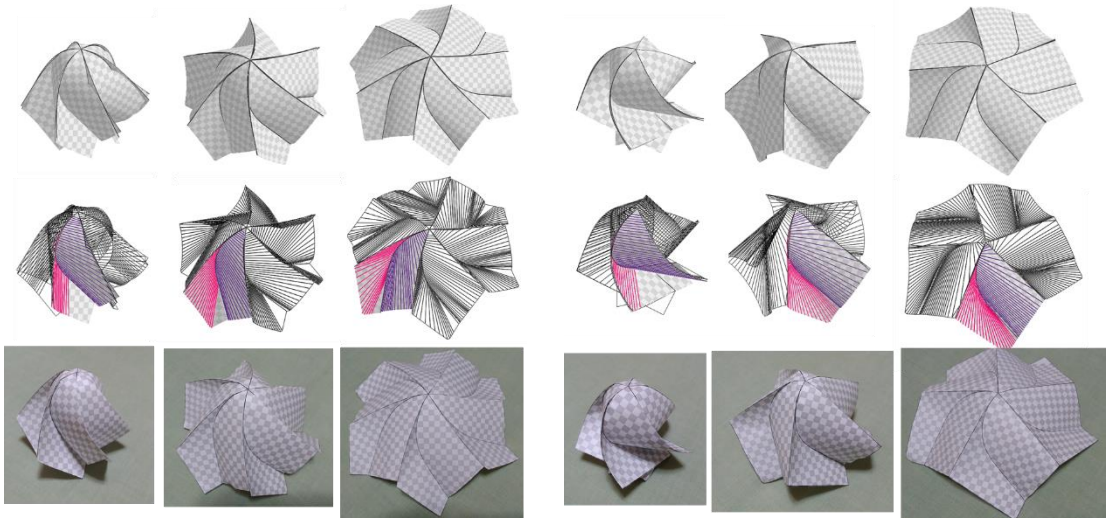
Figure 9-14 Connectivity of segments. Circles on the graph indicate keyframes whose shape and pose are modified by the user. Squares indicate the frames where only the pose is modified.

9.2.3. Comparison with real paper

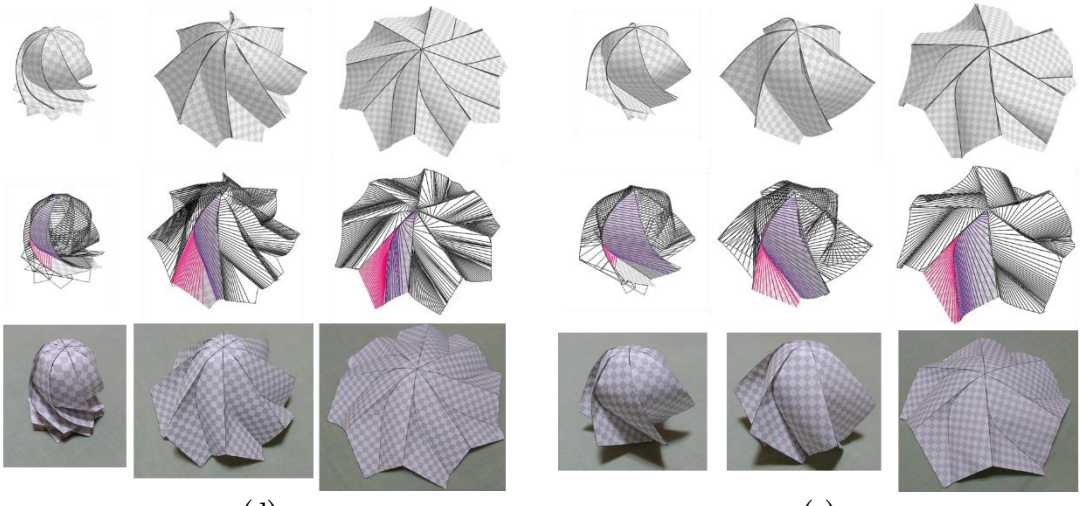
As the visual comparison, the 3D models and the photos of the models are shown in Figure 9-15. The behavior of the paper shape and the rulings were visualized successfully, with only small differences between the photos. The 3D model shows that even if, for the final folded state, the angles between the crease curve and the rulings are designed to be nearly right angles, they become acute in the intermediate state. This makes most rulings end at the edge of the paper instead of the boundary crease, making the boundary creases to have small curvatures while being folded.



(a)



(b) (c)



(d) (e)

Figure 9-15 Visual comparison with real paper. (a) N6, (b) N6_k1, (c) N5, (d) N8, and (e) N6_skew. For each sample, top: rendered 3D models, middle: wireframe models with one segment rendered with rulings, bottom: photos of real paper.

9.2.4. Discussion

By the method of modeling a rotational symmetric curved folding with some simple segments having one curved crease, the folding motion is modeled and visualized successfully, with small gaps between the segments. This new idea expands the possibility of modelling various curved folded surfaces against the result of Chapter 8, where we found that a curved folding with several creases was difficult to manage.

However, from the point of mathematics, this is not a correct result. A straight crease is able to bend or to be in a curved shape only when it is flat or folded completely. In other words, the surface adjacent to a straight crease should always be a flat surface in the intermediate state. On the other hand, the behavior of the real paper shows that the *origami-sphere* is actually folded, forming similar shapes with the 3D model generated by our method. This is thought to be realized by a small deformation of the paper, such as the compressions, stretching, and the shifting of the creases. Our model supports straight creases in the curved folding by (i) the shapes of the boundary crease becoming nearly straight as a result of the ruling transition as shown in Subsection 9.2.3, and (ii) by allowing a small gap on the creases located between the segments, working in similar manner as the small deformation of real paper. By supporting the ruling transition and dividing the crease pattern into segments, the method is able to model the curved folding including straight creases.

As to the usability, we still have a lot to improve, especially on the modification of the 3D shape. It relies heavily on the user's effort to decrease the gap between segments. Even for an expert user, who knows how the torsion and the folding angle should be modified, it may take approximately 5-10 minutes. Some optimization method should help, as in the method of adjusting the shape of the additional curved crease, introduced in Chapter 8. The improvement in the user interface is also expected for better use.

Chapter 10.
Conclusion and future work

In this thesis, we have studied the methods to model and visualize the shape of the paper folded along the curved creases. Towards the goal of clarifying the shape and the ruling configuration of the paper being folded, we developed some prototypes of GUI system which models a curved folded surface as a discretized polygon model, according to the interactive user input, and simulate its folding motion by gradually changing the folding angle and the torsion of the crease curve.

10.1. Summary of contributions

The contributions of the thesis are as follows:

Interactive GUI system
The proposed GUI system supports the user to design a curved crease in two means: (i) It enables interactive manipulation of the shape of the crease curve and the folding angle with an instant visualization of the paper shape, and (ii) the simple user input to adjust the parameters of the crease curve only on a small number of control points on the curve to keep the paper shape smooth. To design a curved crease, the user can modify the parameters defining the crease curve in 2D and 3D, and the folding angle, while checking the resulting paper shape immediately to resolve the rulings crossing and to avoid self-collision. The parameters are input only on some control points on the curved crease and calculated throughout the curve by spline interpolation of degree-three. Since the curvatures and the torsion transit continuously along the crease curve, the generated crease curves are guaranteed to be G2 continuous. Consequently, the Frenet-Serret frame is defined on any point of the curve except for the two ends, which is required for the calculation of the rulings. The continuity of the parameters contributes to the smooth shapes of the curved crease and surfaces, helping to avoid the corruption of the 3D model, such as the crossing of the rulings.

Simulation of the folding motion
Using the same method of designing the paper shape, the system generates the folding motion of the curved crease by the simple interpolation of the folding angle and the torsion, with the 2D crease curve being fixed. As the parameters change from the original value to zero, the paper shape is deformed to be unfolded, keeping the developability of the surfaces by re-calculating the ruling direction and updating the geometry in every time step.

Handling a curved crease with an inflection point
By observing the generated rulings and examining the mathematical principles used in the method, we have noticed the problem that rulings cannot be calculated on an inflection point, or a point with zero curvature, on the crease curve, using the equation

of Fuchs and Tabachnikov [Fuchs_99]. The system handles the problem by rectifying the torsion and the folding angle to meet the properties of a straight crease line locally. To be specific, the torsion of the 3D crease curve is rectified to be zero and the folding angle to be set constant around the zero-curvature point. In the adjacent sections, the parameters are interpolated smoothly to meet the original value at their ends. With the rectification, the curved folding with an inflection point in the center of the crease is modeled, and its folding motion simulated successfully under the same procedure.

Adding creases on the curved surfaces

To design a curved folding with more than one curved crease, we developed another prototype GUI system to add curved creases on a curved surface generated by the first curved crease. For making a user intended paper shape, the task of adjusting the curve shape is quite difficult, because for the added crease, the curvatures, torsion, and the folding angle to calculate the rulings cannot be controlled directly by the user, as the curved surface on one side of the crease being fixed. To reduce the difficulty, we proposed the input restriction and optimization process based on some cost functions. This method was useful to the paper containing only a few creases. As the user adds more creases, adjusting the curve shape to resolve the rulings crossing would become extremely difficult.

Generation of the folding motion of a rotationally symmetric curved folding

At last, the method is applied to generate the folding motion of curved folding with rotational symmetry, with user input. The paper is modeled as a group of identical segments with only one crease, placed in rotational symmetry, so that the curved folding with several creases is modeled by our previous methods. Moreover, by dividing the crease pattern into segments, the method is able to model the curved folding including straight creases, which is mathematically impossible. We assume it is enabled by the small gaps between the segments, representing the small deformation which occurs on the real paper, such as the compressions, stretching, and the shifting of the creases.

10.2. Future work

We state the future work for this thesis as follows:

The intuitive and direct user interface to control the paper shape

In the current system, the user controls the paper shape by adjusting the curvatures, the torsion, and the folding angle of the curved crease. But for more intuitive manipulation, it is necessary to have the interface to control the paper shape directly. An example of such an interface may be the user moving a point on a curved surface in 3D space. Though this is not supported in the current framework, one possible approach may be the optimization process to estimate the parameters of the curved crease to meet the desired conditions. Such an interface will reduce the user's task and will contribute to work with various types of crease patterns.

Propagation of deformation between two curved creases

In adding another crease on the curved surface generated by the first crease, it was quite difficult to control the parameters on the added crease and the rulings on the newly generated developable surface because the curved surface on one side of the crease is fixed. By relaxing this constraint and enabling the first crease to deform according to the newly added crease, easier conditions may be expected. To propagate the deformation of one crease to another, the rulings and the developable surface between the two creases must be consistent. As one curved crease is deformed and the rulings and the curved surface are updated, the other curved crease needs to be deformed to meet the surface between them, which affects the rulings and the surface on the other side of the original crease. So, the process of crease deformation and the propagation should be done through some iteration to make all surfaces to be in existable states.

Folding motion in more variation

For rotational symmetric curved folding, we assumed that the paper shape is rotationally symmetric in every time step in the folding motion. For representing real paper with more reality, it would be necessary to generate the folding motion in more variation, such as going through a non-symmetric shape in some time steps during the folding. One possible method to realize this may be shared from the work on the direct user interface, listed firstly in this section. By directly giving the target positions to the points on the boundary creases, the shape of each segment may be deformed respectively to share the boundary with the adjacent segments, allowing segments to be in different shapes.

Comparison with captured data

In Chapter 5, we introduced our initial attempt to measure the paper shape using some partial low-resolution scanned data. Though this data was not sufficient for the measurement by itself, by combining with the 3D model developed by our method, it would be useful in reconstructing the actual shape of the paper being folded. To realize this, the idea of the optimization process would be effective. By identifying some points on the paper in the images, using color patterns for example, the parameters on the curved crease should be optimized to minimize the errors between the model and the images.

Support for self-collision

In our method, self-collision of the paper is avoided by the visual check and the manipulation of the user. Because the system does not detect or avoid the collisions, the faces on the paper model may intersect with another face through naive manipulation. The first step to this problem is the detection of the collisions and to show a warning to the user so that the user can move the faces to avoid the collisions. Then, the second step would be to deform the paper model by the system, such as faces being pushed and moved by another face. This may also be realized by the direct user interface, listed firstly, to push the point of collision to resolve this.

Finally, we will refer to the long-term goal of this thesis. As the shapes and the behavior of the curved folded paper may be modeled and visualized by our system, we shall now

use the result in the real world, such as processing a thin sheet of material to make a curved folded surface by some automated system, or by a set of very simple operations. Such a curved folding may be applied in, for example, a flattenable structure, furniture, and greeting cards, to list just a few.

References

[Bo_07] Bo, P., Wang, W.: Geodesic-controlled developable surfaces for modeling paper bending.in Computer Graphics Forum 26(3):365-374.2007

[Burgoon_06] Burgoon, R., Wood, Z. J., Grinspun, E.: Discrete shells origami.21st International Conference on Computers and Their Applications, CATA-2006, Seattle, Washington, USA, March 23-25, 2006, Proceedings.

[Chen_04] Chen, H.-Y., Lee, I.-K., Leopoldseder, S., Pottmann, H., Randrup, T., Wallner, J.: On surface approximation using developable surfaces.Graphical Models and Image Processing Volume 61, Issue 2, 1 March 1999, Pages 110-124.

[Choi_16] Choi, I.: Curved folding and planar cutting of simple closed curve on a conical origami.Kodai Math. J. Volume 39, Number 3 (2016), 579-595.

[Chu_02] Chu, C.-H., Sequin, C. H.: Developable Bézier patches: properties and design. Computer-Aided Design Volume 34, Issue 7, June 2002, Pages 511-527.

[Demaine_10a] Demaine, E. D., Demaine, M. L., Koschitz, D.: Reconstructing David Huffman's legacy in curved-crease folding.Origami 5: The 5th International Meeting on Origami in Science, Mathematics and Education, 2010.

[Demaine_10b] Demaine, E. D., Demaine, M. L., Hart, V., Price, G. N., Tachi, T.: (Non)existence of pleated folds: how paper folds between creases.Graphs and Combinatorics May 2011, Volume 27, Issue 3, pp 377–397

[Demaine_11] Demaine, E. D., Demaine, M. L., Koschitz, D., Tachi, T.: Curved crease folding: a review on art, design and mathematics.In: Proceedings of the IABSE-IASS Symposium, 2011.

[Demaine_14] Demaine, E. D., Demaine, M. L., Huffman, D. A., Koschitz, D., Tachi, T.: Characterization of curved creases and rulings: design and analysis of lenz tessellations.Origami 6: The Sixth International Conference on Origami in Science, Mathematics, and Education (2014) 209–230.

[Demaine_18] Demaine, E. D., Demaine, M. L., Huffman, D. A., Koschitz, D., Tachi, T.: Conic crease patterns with reflecting rule lines.Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education, 2018, 573-589.

[Dias_12] Dias, M. A., Santangelo, C. D.: The shape and mechanics of curved-fold origami structures.EPL (Europhysics Letters), Volume 100, Number 5. 2012

[Duncan_82] Duncan, J.P., Duncan, J.L.: Folded developables.Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 383(1784),

pp.191–205.

[Epps_14] Epps, G.: Robofold and robots. IO.Made by Robots: Challenging architecture at a larger scale, Architectural Design, 2014, 68–69.

[Farmer_05] Farmer, S.M., Calladine, C.R.: Geometry of “developable cones”. International Journal of Mechanical Sciences 47 (2005) 509–520

[Frey_04] Frey, W. H.:Modeling buckled developable surfaces by triangulation. Computer-Aided Design 36 (2004) 299–313

[Fuchs_99] Fuchs, D., Tabachnikov, S.: More on paperfolding. The American Mathematical Monthly, Vol. 106, No.1, pp. 27-35,1999.

[Ghassaei_18] 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.

[Gonzalez-Quintial_15] Gonzalez-Quintial, F., Barrallo, J., Artiz-Elkarte, A.: Freeform surfaces adaptation using developable strips and planar quadrilateral facets. Journal of Facade Design and Engineering 3 (2015) 59–70

[Hoffmann_16] Hoffman, T., Sageman-Furnas, A. O., Wardetzkyy, M.: A discrete parametrized surface theory in R3. International Mathematics Research Notices, Volume 2017, Issue 14, July 2017, Pages 4217–4258

[Huffman_76] Huffman, D.: Curvature and creases: a primer on paper. IEEE TRANSACTIONS ON COMPUTERS, VOL. C-25, NO. 10, OCTOBER 1976.

[Hwang_15] Hwang, H.-D., Yoon, S.-H.: Constructing developable surfaces by wrapping cones and cylinders. Computer-Aided Design 58 (2015) 230–235.

[Kilian_08a] Kilian, M., Fl¨ory, S., Chen, Z., Mitra, N. J., Sheffer, A., Pottmann, H.: Developable surfaces with curved creases. Advances in Architectural Geometry, 2008.

[Kilian_08b] Kilian, M., Fl¨ory, S., Chen, Z., Mitra, N. J., Sheffer, A., Pottmann, H.: Curved folding. ACM Transactions on Graphics 27:3 (2008), 75:1-9.

[Kilian_18] Kilian, M., Monszpart, M., Mitra, N. J.: String actuated curved folded surfaces. ACM Transactions on Graphics, 36(3), 2017, 25:1-13.

[Lawrence_11] Lawrence, S.: Developable surfaces: their history and application. Nexus Network Journal 13 (2011) 701–714

[Lee_13] Lee, N., Close, S.: Curved pleat folding for smooth wrapping. Proceedings of the Royal Society A Mathematical, Physical and Engineering Science 8 July 2013 Volume 469 Issue 2155

[Lee_18] Lee, T. U., Chen, Y., Gattas, J. M.: Curved-crease origami with mutiple states. Origami 7: The 7th International Meeting on Origami in Science, Mathematics and Education, 2018, 849-864.

[Leopoldseder_03] Leopoldseder, S., Pottmann, H.: Approximation of developable surfaces with cone spline surfaces. Computer-Aided Design Volume 30, Issue 7, June 1998, Pages 571-582

[Liu_06] Liu, Y., Pottmann, H., Wallner, J., Yang, Y.-L., Wang, W.: Geometric modeling with conical meshes and developable surfaces. ACM Trans. Graph. 25, 3 (2006).

[Liu_07] Liu, Y.-J., Lai, Y.-K., Hu, S.-M.: Developable strip approximation of parametric surfaces with global error bounds. 15th Pacific Conference on Computer Graphics and Applications (PG'07), 2007.

[Mitani_04] Mitani, J., Suzuki, H.: Making papercraft toys from meshes using strip-based approximate unfolding. ACM Transactions on Graphics (TOG) TOG Homepage Volume 23 Issue 3, August 2004, Pages 259-263

[Mitani_06] Mitani, J.: Strip creation for designing curved papercraft models adopting mesh subdivision scheme.

[Mitani_09] Mitani, J.: A design method for 3D origami based on rotational sweep. Computer-Aided Design & Applications 6(1), 2009, 69-79.

[Mitani_11] Mitani, J., Igarashi, T.: Interactive design of planar curved folding by reflection. Pacific Conference on Computer Graphics and Applications - Short Papers, 2011, 77–81.

[Mitani] flickerhttps://www.flickr.com/photos/jun_mitani/

[Miyashita_15] Miyashita, S., DiDio, I., Ananthabhotla, I., An, B., Sung, C., Arabagi, S., Rus, D.: Folding angle regulation by curved crease design for self-assembling origami propellers. J. Mechanisms Robotics 7(2), 021013 (May 01, 2015) Paper No: JMR-14-1216; doi: 10.1115/1.4029548

[Narain_13] Narain, R., Pfaff, T., O'Brien, J. F.: Folding and crumpling adaptive sheets. ACM Transactions on Graphics, 32(4):51:1–8, July 2013. Proceedings of ACM SIGGRAPH 2013, Anaheim.

[Nelson_16] Nelson, T. G., Lang, R. J., Magleby, S. P., Howell, L. L.: Curved-folding-inspired deployable compliant rolling-contact element (D-CORE). Mechanism and Machine Theory Volume 96, Part 2, February 2016, Pages 225-238

[Perriollat_13] Perriollat, M., Bartoli, A.: A computational model of bounded developable surfaces with application to image-based 3D reconstruction. Computer Animation and Virtual Worlds, 2013.

[Peternell_04a] Peternell, M.: Developable surface fitting to point clouds. Computer Aided Geometric Design, 2004.

[Peternell_04b] Peternell, M.: Recognition and reconstruction of developable surfaces from point clouds.Geometric Modeling and Processing, 2004

[Polthier_06] Polthier, K., Schmies, M.: Straightest geodesics on polyhedral surfaces. Proceeding SIGGRAPH '06 ACM SIGGRAPH 2006 Courses Pages 30-38

[Rabinovich_18a] Rabinovich, M., Hoffmann, M., Sorkine, O.: Discrete geodesic nets for modeling developable surfaces. ACM Transactions on Graphics, 37(2), 2018.

[Rabinovich_18b] Rabinovich, M., Hoffmann, M., Sorkine, O.: The shape space of discrete orthogonal geodesic nets. ACM Transactions on Graphics, 37(6), 2018.

[Roeschel_16] Roeschel, O.: Curved folding with pairs of cylinders. 17TH INTERNATIONAL CONFERENCE ON GEOMETRY AND GRAPHICS

[Rose_07] Rose, K., Sheffer, A., Wither, J., Cani, M.-P., Thibert, B.: Developable surfaces from arbitrary sketched boundaries. Eurographics Symposium on Geometry Processing (2007)

[Seffen_18] Seffen, K. A.: Spherical images and inextensible curved folding. PHYSICAL REVIEW E 97, 023004

[Solomon_12] Solomon, J.; Vouga, E.; Wardetzky, M.; Grinspun, E: Flexible developable surfaces. Eurographics Symposium on Geometry Processing 2012

[Stein_18] Stein, O., Grinspun, E., Crane, K.: Developability of triangle meshes. ACM Trans. Graph. 37, 4 (2018)

[Sternberg_09] Sternberg, S.: Curves and flats. Origami4: Proceedings of the 4th International Meeting of Origami. Natick, AK Peters Ltd, Robert J. Lang (editor), 2009.

[Tachi_09] Tachi, T.: Simulation of rigid origami. Origami 4: The Fourth International Conference on Origami in Science, Mathematics, and Education (2009) 175–187.

[Tachi_11a] Tachi, T.: One-DOF rigid foldable structures from space curves. in Proceedings of the IABSE-IASS Symposium 2011, London, UK, September 20-23, 2011.

[Tachi_11b] Tachi, T., Epps, G.: Designing one-DOF mechanisms for architecture by rationalizing curved folding. Proceedings of the International Symposium on Algorithmic Design for Architecture and Urban Design, ALGODE TOKYO 2011, March 14-16, 2011, Tokyo, Japan

[Tachi_13] Tachi, T.: Composite rigid-foldable curved origami structure. in Proceedings of Transformables 2013, Seville, Spain, Sep.18-20, 2013.

[Tang_16] Tang, C., Bo, P., Wallner, J., Pottmann, H.: Interactive design of developable surfaces. ACM Transactions on Graphics, 35(2), 2016, 12:1-12.

[Wang_08] Wang, C. C. L.: Flattenable mesh surface fitting on boundary curves. Journal of Computing and Information Science in Engineering 8(2) · January 2008

[Zhu_13] Zhu, L., Igarashi, T., Mitani, J.: Soft folding. Computer Graphics Forum, Volume 32, Issue 7, pages 167–176, October 2013.