Simulation of Rigid Origami

Tomohiro Tachi
The University of Tokyo
tachi.tomohiro@gmail.com

Abstract

This paper presents a system for computer based interactive simulation of origami, based on rigid origami model. Our system shows the continuous process of folding a piece of paper into a folded shape by calculating the configuration from the crease pattern. The configuration of the model is represented by crease angles, and the trajectory is calculated by projecting angles motion to the constrained space. Additionally, polygons are triangulated and crease lines are adjusted in order to make the model more flexible. Flexible motion and comprehensible representation of origami help users to understand the structure of the model and to fold the model from a crease pattern.

1 Introduction

Simulation of origami is very important for representing or modeling origami in computer. It helps paper folders to understand the structures of origami models, and can be used as a tool for designers to draw diagrams. We propose a system for simulating folding process from crease pattern to folded base, based on rigid origami simulation (Figure 1).

There have been many approaches for simulating origami. One way is to simulate origami by a sequence of simple folding steps as shown in [1] by Miyazaki et al. Since the final state of an origami model is represented by folding steps, it is easy to reconstruct an animation from a sheet of paper to the model. However, the origami models that can be represented in this system are limited by the simplicity of folding steps, and not suitable for many complex origami models whose folding process cannot be divided into simple folding steps.

Figure 1: Screenshots of the simulation program. Left: base of "Kamehameha Wave"(by the author). Middle: pleated hypar. Right: base of "Tachikoma" (by the author).

Thus representing origami models by crease pattern seems suitable. ORIPA is a crease pattern editor with estimation of folded figure, developed by Mitani [2]. Final state with stacking order is estimated, though sometimes the estimation fails. Thickness of paper is represented for better understanding of the structure of the model, but there are no intermediate state from crease pattern to the final shape.

It is possible to simulate intermediate state of folding, by using rigid origami model (i.e. plates connected by hinges constrained around vertices). Balkcom [3] uses rigid origami model for origami simulation. His method is based on virtual cutting and combination of forward and inverse kinematics. The calculation of the trajectory runs fast in this method, but the mountain-valley values of some creases are not taken into account for the folding motion.

Our system is based on rigid origami simulation, and uses crease angles of all fold lines as variables to represent configuration of an origami model. The simulation method is based on projection to the constraint space, thus possible to use folding motion of all crease lines as driving force of folding. This results in a robust overall motion which is independent of the positions of the virtual cutting. Additionally, our system can add flexibility to origami models by adding and adjusting crease lines. In this method, folding motion of non rigidly foldable models can also be simulated.

The system provides a smooth and comprehensible interactive animation of folding from a crease pattern to the folded base, which helps users to understand the structure of the origami model and the way to fold the model from the crease pattern. This helps paper folders to fold models from a crease pattern, and also encourages designers to use a crease pattern as a medium for publishing origami models.

2 Kinematics of Rigid Origami

In our model, an origami configuration is represented by folding angles of edges. Folding angles change according to mountain-valley values of the fold lines, when the paper get folded. Those edges are connected with facets and form closed loops, and for each closed loop we have a constraint on angle movement. If there is no hole inside the loop, then the constraint is satisfied by intersection of constraints by single vertices inside the loop (Figure 2). Our model assumes that the paper is without a hole. Folding motion is numerically calculated using linear approximation of constraints by single vertices.

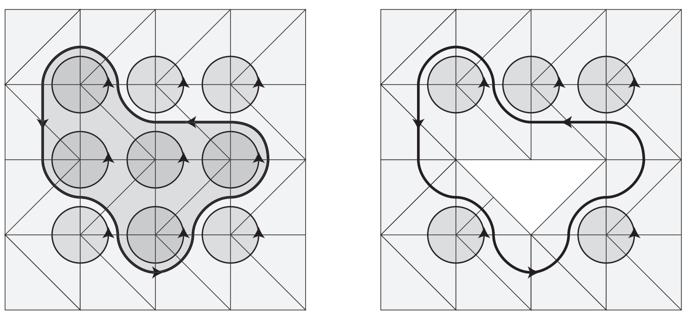

Figure 2: Left: No hole is inside the loop and the constraint is satisfied by the intersection of single-vertex constraints inside. Right: A hole is inside the loop and we have to think about the constraint around the hole.

Necessary conditions for single-vertex rigid origami shown in [4] by Belcastro and Hull are used as constraints of multi-vertex rigid origami model. Suppose there are n edges $\ell_1, \cdots, \ell_n$ connected to the vertex. For a single-vertex origami, the 3×3 matrix constraint function $\mathbf{F}$ of fold angles $\rho_1, \cdots, \rho_n$ is given by

$$\mathbf{F}(\rho_1, \cdots, \rho_n) = \chi_1 \cdots \chi_{n-1} \chi_n = \mathbf{I} \tag{1}$$

where matrices $\chi_1, \cdots, \chi_n$ represent rotation by fold lines $\ell_1, \cdots, \ell_n$ respectively (Figure 3). Derivative of the equation is given by

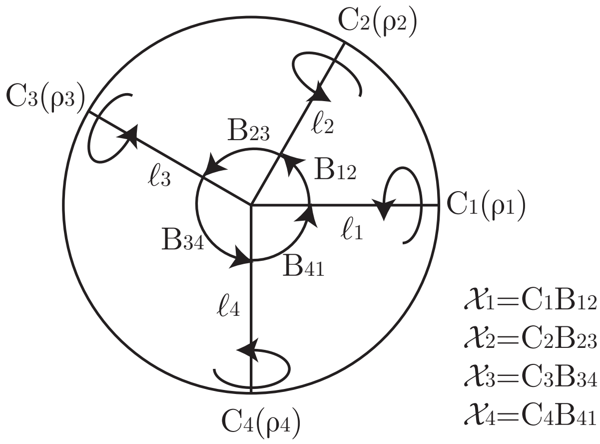

Figure 3: An example of rotation matrices of a single-vertex origami.

$$\frac{d\mathbf{F}}{dt} = \frac{\partial \mathbf{F}}{\partial \rho_1}\dot{\rho_1} + \cdots + \frac{\partial \mathbf{F}}{\partial \rho_i}\dot{\rho_i} + \cdots + \frac{\partial \mathbf{F}}{\partial \rho_n}\dot{\rho_n} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2}$$

We get 9 equations (i.e. one for each element of $\mathbf{F}$) for angles movement $\rho_1, \cdots, \rho_n$ represented using $9 \times n$ matrix.

$$\underbrace{\begin{bmatrix} \frac{\partial \mathbf{F}}{\partial \rho_1}_{(1,1)} & \cdots & \frac{\partial \mathbf{F}}{\partial \rho_n}_{(1,1)} \\ \frac{\partial \mathbf{F}}{\partial \rho_1}_{(1,2)} & \cdots & \frac{\partial \mathbf{F}}{\partial \rho_n}_{(1,2)} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{F}}{\partial \rho_1}_{(3,3)} & \cdots & \frac{\partial \mathbf{F}}{\partial \rho_n}_{(3,3)} \end{bmatrix}}_{9 \times n \text{matrix}} \begin{bmatrix} \dot{\rho_1} \\ \vdots \\ \dot{\rho_n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3}$$

However, since $\mathbf{F}$ is a rotation matrix (i.e. orthogonal matrix) the equations are redundant and only 3 of 9 equations are independent. Partial derivative of an orthogonal matrix $\mathbf{F}$ at $\mathbf{F} = \mathbf{I}$ is a skew-symmetric matrix because

$$\frac{\partial \mathbf{F}}{\partial \rho_i} + \left(\frac{\partial \mathbf{F}}{\partial \rho_i}\right)^{\mathrm{T}}\Bigg|_{\mathbf{F}=\mathbf{I}} = \frac{\partial \mathbf{F}}{\partial \rho_i}\mathbf{F}^{\mathrm{T}} + \mathbf{F}\frac{\partial \mathbf{F}^{\mathrm{T}}}{\partial \rho_i}\Bigg|_{\mathbf{F}=\mathbf{I}} = \frac{\partial}{\partial \rho_i}\left(\mathbf{F}\mathbf{F}^{\mathrm{T}}\right)\Bigg|_{\mathbf{F}=\mathbf{I}} = \mathbf{0} \tag{4}$$

Partial derivative of the constraint function is represented in the following form using 3 independent variables a, b, c.

$$\frac{\partial \mathbf{F}}{\partial \rho_i} = \begin{bmatrix} 0 & -a & c \\ a & 0 & -b \\ -c & b & 0 \end{bmatrix} \tag{5}$$

We get a $3 \times n$ matrix instead of a $9 \times n$ matrix shown in (3),

$$\begin{bmatrix} a_1 & \cdots & a_n \\ b_1 & \cdots & b_n \\ c_1 & \cdots & c_n \end{bmatrix} \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6}$$

Then we rewrite the matrix using global edge number. Assume that edge i is connected and edge j is not connected to vertex k. Constraint by vertex k is represented by the matrix $\mathbf{C}_k$, whose ith column is identical to the column of the previous matrix that corresponds to the edge, and jth column is zero vector.

$$[\mathbf{C}_k] \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_N \end{bmatrix} = \underbrace{\begin{bmatrix} \cdots & a_{ki} & \cdots & 0 & \cdots \\ \cdots & b_{ki} & \cdots & 0 & \cdots \\ \cdots & c_{ki} & \cdots & 0 & \cdots \end{bmatrix}}_{3\times N\text{matrix}}^{\quad i\text{th} \qquad j\text{th}} \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_i \\ \vdots \\ \dot{\rho}_j \\ \vdots \\ \dot{\rho}_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

where N is the number of edges. Global constraints by multiple of vertices are given by the intersection of the conditions.

$$\mathbf{C}\dot{\boldsymbol{\rho}} = \underbrace{\begin{bmatrix} [\mathbf{C}_1] \\ \vdots \\ [\mathbf{C}_M] \end{bmatrix}}_{3M\times N\text{matrix}} \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_N \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{8}$$

where M is the number of vertices inside the paper.

If and only if the rank of matrix $\mathbf{C}$ is less than N, the linear equation has non-trivial solutions. The solution of this linear equation is calculated using pseudo-inverse matrix $\mathbf{C}^+$ of matrix $\mathbf{C}$.

$$\dot{\boldsymbol{\rho}} = \left[\mathbf{I}_N - \mathbf{C}^+\mathbf{C}\right] \dot{\boldsymbol{\rho}}_0 \tag{9}$$

where $\dot{\boldsymbol{\rho}}_0$ is unconstrained value of the angles velocity calculated by mountain-valley value of the fold edges. The idea is to project unconstrained angles movement to constrained space by using orthogonal projection matrix $[\mathbf{I}_N - \mathbf{C}^+\mathbf{C}]$. Trajectory can be calculated by Euler integration.

$$\Delta\boldsymbol{\rho} = \dot{\boldsymbol{\rho}}(t)\Delta t = \left[\mathbf{I}_N - \mathbf{C}^+\mathbf{C}\right] \Delta\boldsymbol{\rho}_0 \tag{10}$$

However this Euler integration results in accumulation of numerical error, so we modify equation(8) using the residual vector $\mathbf{r}$.

$$\mathbf{C}\dot{\boldsymbol{\rho}} = -\mathbf{r} \tag{11}$$

where

$$\mathbf{r} = \begin{bmatrix} r_{1a} & r_{1b} & r_{1c} & \cdots & r_{ka} & r_{kb} & r_{kc} & \cdots r_{Ma} & r_{Mb} & r_{Mc} \end{bmatrix}^{\mathrm{T}} \tag{12}$$

where r_{ka}, r_{kb}, r_{kc} are the residual of elements of $\mathbf{F}$ corresponding to a_k, b_k, c_k respectively.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 - r_a & 0 + r_c \\ 0 + r_a & 1 & 0 - r_b \\ 0 - r_c & 0 + r_b & 1 \end{bmatrix} \tag{13}$$

The modified solution is

$$\Delta\boldsymbol{\rho} = -\mathbf{C}^{+}\mathbf{r} + \left[\mathbf{I}_N - \mathbf{C}^{+}\mathbf{C}\right]\Delta\boldsymbol{\rho}_0 \tag{14}$$

Pseudo-inverse is calculated as $\mathbf{C}^{+} = \mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{C}^{\mathrm{T}}\right)^{-1}$ if $\mathbf{C}$ is full rank and $3M < N$. When this condition is satisfied, the linear equation is solved as follows.

$$\begin{aligned}\Delta\boldsymbol{\rho} &= -\mathbf{C}^{+}\mathbf{r} + \left[\mathbf{I}_N - \mathbf{C}^{+}\mathbf{C}\right]\Delta\boldsymbol{\rho}_0 \\ &= -\mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{r} + \left[\mathbf{I}_N - \mathbf{C}^{\mathrm{T}}\left(\mathbf{C}\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C}\right]\Delta\boldsymbol{\rho}_0 \\ &= \Delta\boldsymbol{\rho}_0 - \mathbf{C}^{\mathrm{T}}\left\{\left(\mathbf{C}\mathbf{C}^{\mathrm{T}}\right)\backslash\left(\mathbf{r} + \mathbf{C}\Delta\boldsymbol{\rho}_0\right)\right\}\end{aligned} \tag{15}$$

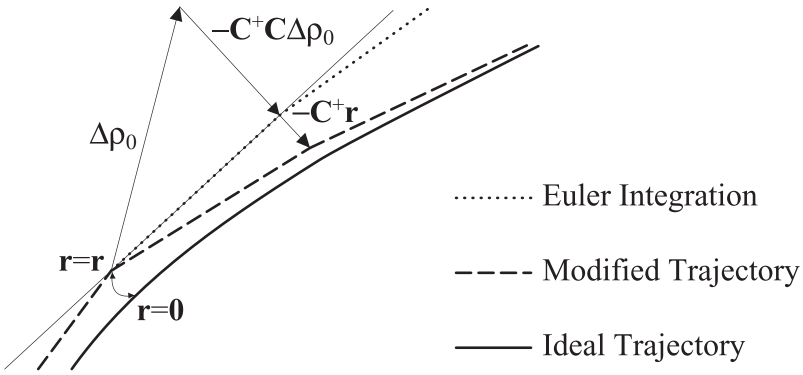

Figure 4: Idea of the trajectory. $[\mathbf{I}_N - \mathbf{C}^{+}\mathbf{C}]$ is a projection matrix that project unconstrained angles movement vector $\Delta\boldsymbol{\rho}_0$ to the constraint linear space. The angles movement vector is further modified to reduce residual.

3 Adding and Adjusting Crease Lines

3.1 Closed Vertex

If we simply fold a paper on already folded edges, the intersection of a new line and an old line is a vertex with fold lines whose motion is locked by the new fold. We call this kind of a singular vertex a "closed vertex." In our system, a vertex is detected as a closed vertex if there is a set of three adjacent crease lines connected to the vertex whose outer two of are symmetrical about the center crease and their mountain-valley assignments are opposite.

There are two ways to avoid closed vertex. One way is to split the "inside angle", which we call the angle that comes inside when folded, by adding crease lines, and another way is to reduce the inside angle (Figure 5).

3.2 Adding Crease Lines by Triangulation

Many rigid origami models with valency 4 vertices are not rigidly foldable because of the lack of the degrees of freedom (DOF). Total degrees of freedom of the model are $N - 3M$

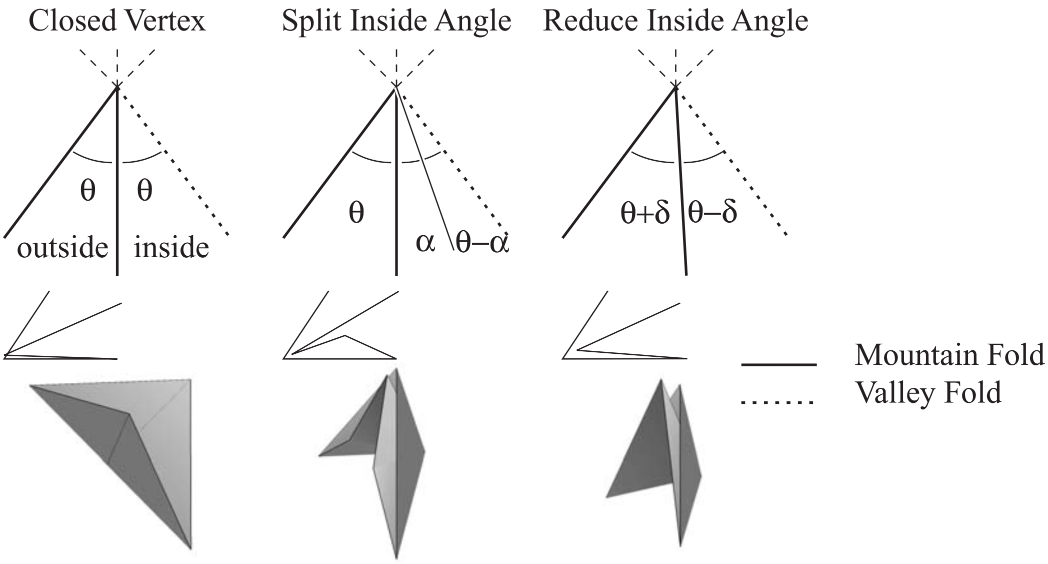

Figure 5: Two ways to avoid avoid closed vertex.

if the constraint matrix is full rank (i.e. if there is no singularity) where M and N are the number of vertices and edges inside the paper (i.e. not on the perimeter) respectively.

Polygons with more than 3 vertices are triangulated. $k-3$ degrees of freedom are added to the model per one k-gon by adding $k-3$ crease lines on each polygon. If all the polygons are triangulated, the total degrees of freedom of the model is $N_0 - 3$ where N_0 is number of edges on perimeter of the paper (Figure 6). There are multiple ways to add crease lines when triangulating polygons. Appropriate crease lines are added so that as many inside angles as possible are split.

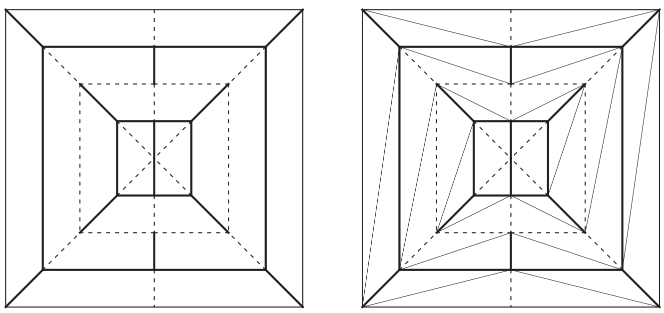

Figure 6: An example of triangulation. Left: Not triangulated model is over constrained ($M = 19, N = 42, \mathrm{DOF} = N - 3M < 0$). Right: Triangulated model has 3 DOF ($M = 19, N = 60, N_0 = 6, \mathrm{DOF} = N - 3M = N_0 - 3 = 3$).

3.3 Adjusting Crease Lines

Crease lines are adjusted to avoid singularity by closed vertices. Position of each vertex is moved to the direction that reduces inside angles of closed vertices. Amount of displacement for each vertex is set according to the "angle unevenness" of the vertex, which is defined as $\{\max(\rho_i) - \min(\rho_i)\}$, for better balance of folding process of fold lines. This reasonably results in no adjustment of the crease pattern when both not folded and completely folded state.

Most origami models have many flat foldable vertices. Flat foldable vertices are kept flat foldable during the adjustment of crease lines so that an adjusted model can be folded to the final state. Kawasaki's theorem is used as the constraints on displacement of the vertices [5]. Here is Kawasaki's theorem:

$$\sum_{i:\text{odd}} \theta_{k,i} = \sum_{i:\text{even}} \theta_{k,i} = \pi \tag{16}$$

where $\theta_{k,i}$ represents ith angle between the edges around vertex k (Figure 7).

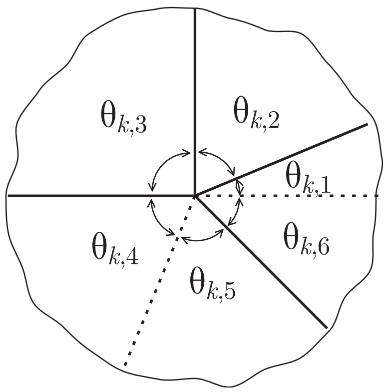

Figure 7: A flat-foldable vertex k

Displacement of the vertices are represented as a $2M_{all} \times 1$ vector $\begin{bmatrix} \Delta\mathbf{X} \\ \Delta\mathbf{Y} \end{bmatrix}$ where M_{all} is the number of all vertices. Constraint for the displacement is given by

$$\underbrace{\begin{bmatrix} \frac{\partial \mathbf{G}}{\partial \mathbf{X}} & \frac{\partial \mathbf{G}}{\partial \mathbf{Y}} \end{bmatrix}}_{[\mathbf{C}_{adj}]} \begin{bmatrix} \Delta\mathbf{X} \\ \Delta\mathbf{Y} \end{bmatrix} = \mathbf{0} \tag{17}$$

where $\mathbf{G}$ is a vector function of $\mathbf{X}$ and $\mathbf{Y}$ whose kth element is

$$\mathbf{G}\left(\mathbf{X}, \mathbf{Y}\right)_k = \sum_{i:\text{odd}} \theta_{k,i} - \sum_{i:\text{even}} \theta_{k,i} \tag{18}$$

Constrained displacement of the vertices are calculated in the same way as calculating trajectory of the angles.

$$\mathbf{C}_{adj}^{+}\mathbf{r}' + \left[\mathbf{I}_{2M_{all}} - \mathbf{C}_{adj}^{+}\mathbf{C}_{adj}\right] \begin{bmatrix} \Delta\mathbf{X}_0 \\ \Delta\mathbf{Y}_0 \end{bmatrix} \tag{19}$$

where $\begin{bmatrix} \Delta\mathbf{X}_0 \\ \Delta\mathbf{Y}_0 \end{bmatrix}$ are the unconstrained vertex displacement and $\mathbf{r}'$ is the residual vector of function $\mathbf{G}$.

3.4 Result of Adding and Adjusting Crease Lines

Figure 8 shows the results of adjusting crease lines after triangulating the models. Some models become flexible enough to show how the facets are stacked when all the crease lines are simultaneously folded. Other models are possible to be folded only when their crease lines are adjusted. Overall animation of folding from crease pattern to final shape is smoother and more comprehensible when the crease lines are adjusted.

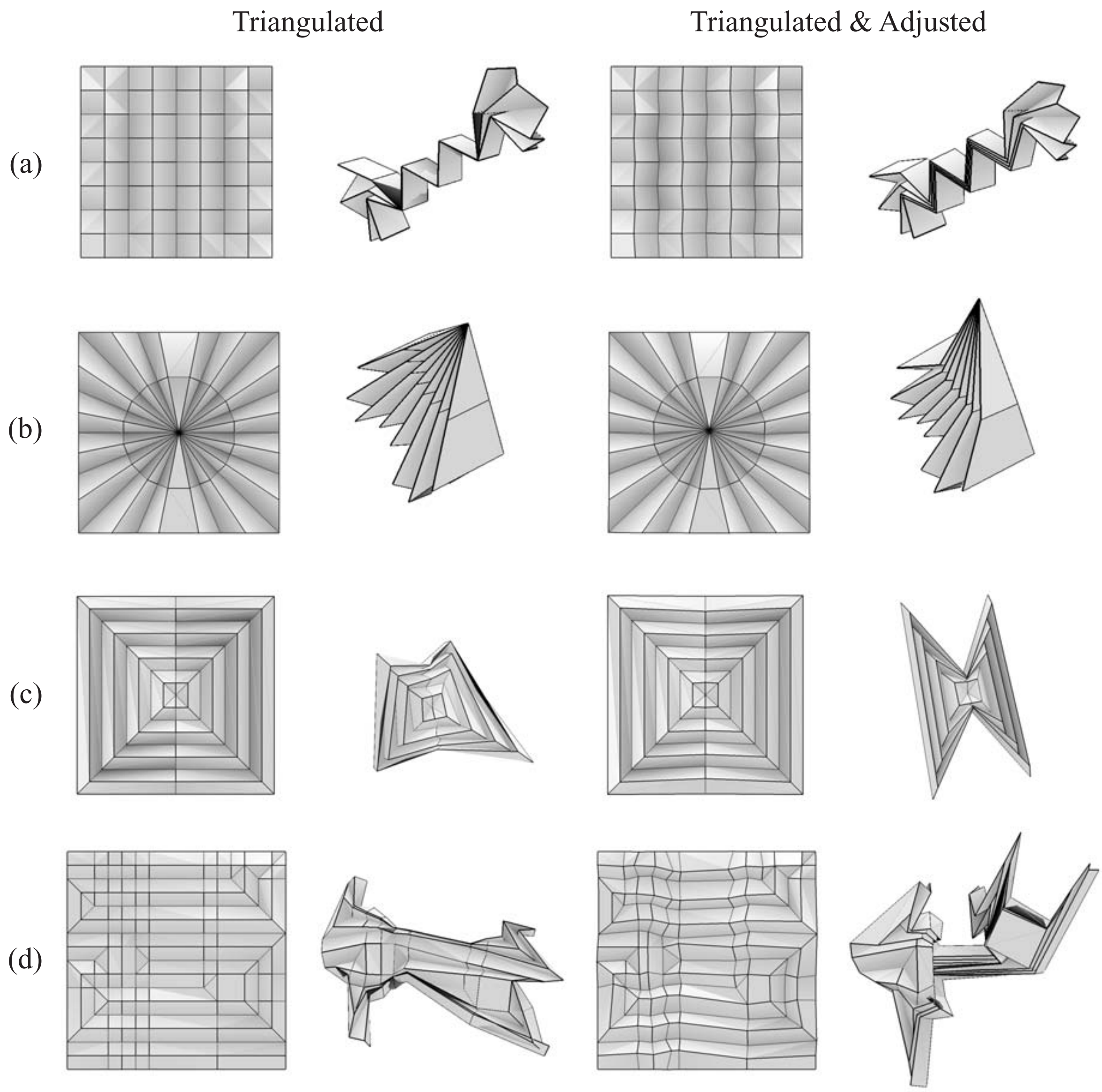

Figure 8: (a)(b) Adjusting crease lines adds flexibility to the model and results in more natural and comprehensible origami representation. (c)(d) Some models cannot be folded to the final state without adjustment. (Model in (d) is base of "Teatime" by the author.)

4 Implementation and Result

The system was implemented as a program written in C with OpenGL. The user can interactively simulate folding and unfolding of origami models, whose crease pattern data is given in ORIPA format. Conjugate gradient method was used for solving the linear equation. The program runs in an interactive speed for many complex origami models (ex. 16×16 box-pleats models). Screenshots of the program is shown in figure 1.

Two kinds of penalty force are used to avoid local self-intersection. One limits the folding angles of mountain or valley crease lines to $[-\pi, \pi]$ to make two adjacent facets not intersect each other. The other limits the folding angles of added crease lines according to folding angles of adjacent crease lines to sufficiently make connected three facets which share one vertex do not intersect each other. Although this self-intersection avoidance is not sufficient to avoid global or complicated local self-intersection, it works for many models.

The system also supports transition from one folded state to another folded state so that it can be used as a tool for drawing diagrams. The folding process can be controlled by a set of multiple crease patterns. However, it is observed that many origami folding steps are not rigidly foldable. Sink folds and some reverse folds are not possible to be executed without unfolding all crease lines (Figure 9).

(a)

(b)

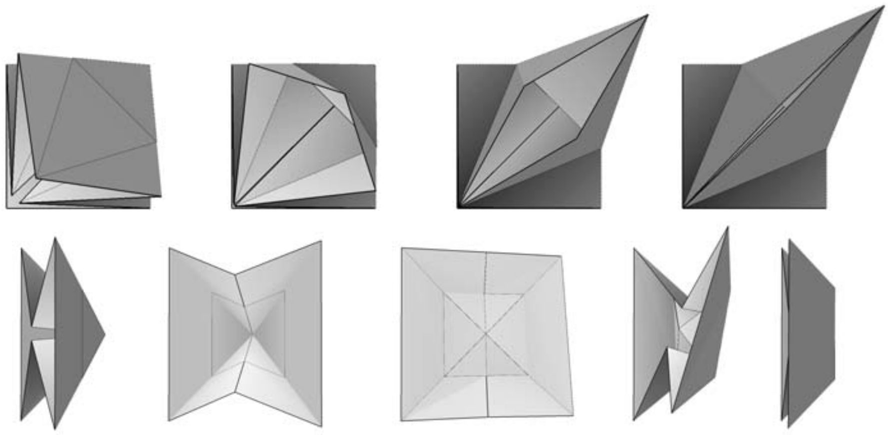

Figure 9: Transition from one folded state to another. (a) Petal fold can be simulated. (b) Simple open sink can be folded, but need to unfold all crease lines during the folding process.

Many models can be folded to the desired state in this system, while many cannot be folded. We have observed that foldability of the model is determined by how the model have closed folds in the structure rather than complexity of the model. In general, origami models that is actually easy to be opened into a sheet of paper in real world can be easily folded and unfolded. For example, Miura fold, waterbomb tessellation, and pleated hypars are smoothly folded in the simulation, while a crane with finishing, four-crane base or tatou cannot be not folded to the final state (Figure 10).

The simulation program is planned to be available in the author's website (`http://www.tsg.ne.jp/TT/`).

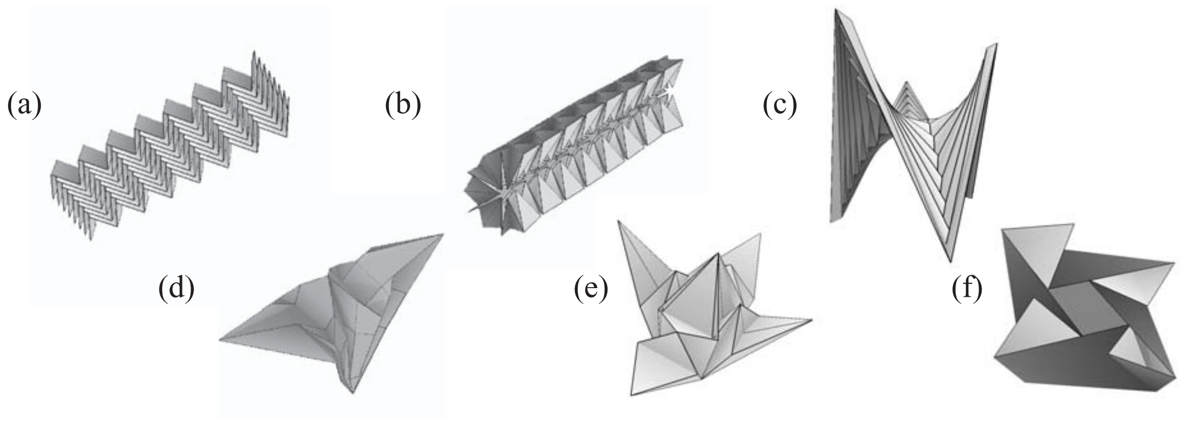

Figure 10: Example of simulation. (a)Miura fold (by Koryo Miura), (b)waterbomb tessellation and (c)pleated hypar are foldable, and (d)crane with finishing, (e)four-crane base and (f)tatou get stuck or intersect themselves before folded.

5 Conclusions and Future Works

We proposed a system for simulating folding motion of origami by calculating the trajectory by projection to the constrained space based on rigid origami model. An additional method was proposed for making origami models more flexible by triangulating polygons and adjusting crease lines. Moreover we investigated local self-intersection avoidance and transition from one state to another state. As a result we could get an interactive system that user can fold and unfold relatively "open" origami models from crease pattern in a robust, smooth, and comprehensible way.

Global self-intersection avoidance problem and stacking order problem are not solved in this paper and remain to be the future works. Another work to be done in the future is to control the surface of high-DOF origami models through user interaction, which is useful for understanding how an origami surface three-dimensionally changes its form.

References

[1] S. Miyazaki, T. Yasuda S., Yokoi, and J. Toriwaki. An origami playing simulator in the virtual space. *The Journal of Visualization and Computer Animation*, 7(1):25–42, 1996.

[2] J. Mitani. Oripa; origami pattern editor. `http://mitani.cs.tsukuba.ac.jp/pukiwiki-oripa/`, 2005.

[3] D. Balkcom. *Robotic Origami Folding*. PhD thesis, Carnegie Mellon University, 2002.

[4] S. Belcastro and T. Hull. A mathematical model for non-flat origami. In *Origami3: Proceedings of the 3rd International Meeting of Origami Mathematics, Science, and Education*, pp. 39–51, 2002.

[5] T. Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In *Proceedings of the 1st International Meeting of Origami Science and Technology*, pp. 229–237, 1989.