Modeling and animation of 3D Origami using spring-mass simulation

Yohsuke FURUTA † Jun MITANI †,†† and Yukio FUKUI †

† Graduate School of Systems and Information Engineering, University of Tsukuba.

†† PRESTO, Japan Science and Technology Agency.

Abstract Origami is known as a Japanese art and play that creates a shape by folding a square sheet of paper. Before folding Origami, we have to know the process of it. Although Orizu (a diagram that presents the way of folding) or movies are used as instruction manuals usually, the view points of them are fixed and we cannot change them. So, it is sometimes difficult to understand the process of folding with these manuals. To solve this problem, 3D CG technology can be used. In this paper, we propose the methods for generating the 3D Origami model and the animation using spring-mass models. We propose two different models and we call them the *spring-mesh model* and the *spring-hinge model*. As these models can represent flexible models, it is possible to simulate non-Rigid Origami. The spring-mesh model is an Origami model that represents faces with spring networks and the spring-hinge model is a model that represents all hinges with springs and faces with rigid polygons. We compared these models and studied each characteristic.We also propose a method for generating seamless 3D Origami animation. In our method, the positions of faces are generated by key-frame interpolation and then revisions are added to them. We made some 3D Origami animations with our method, and we found that our method could generate natural movement of flaps in the animation.

Keywords Origami, Spring-mass model, simulation, animation

1. Introduction

Origami is known as a Japanese art and play that creates a shape by folding a square sheet of paper. Since folding a square is compatible with geometrical problems, many studies about Origami had been done in the field of mathematics. The fruits of the studies have been applied to recent engineering works such as designing foldable structures. We can say that Origami offersus various possibilities.

Before folding Origami, we have to know the process of it. Although Orizu (a diagram that presents the way of folding) or movies are used as instruction manuals usually, the view points of them are fixed and we cannot change them. So, it is sometimes difficult to understand the process of folding with these manuals. To solve this problem, it is required to view the folding process from arbitrary angles. This may be achieved by using 3D CG technology. In this paper, we propose a method for generating the 3D Origami model and the animation.

In the past, some methods were proposed that simulated Origami folding by assuming that Origami was constructed with rigid planes and hinges, called Rigid Origami model. In this model, the shape of folding Origami is calculated in the manner of rigid kinematics. Although this model works fine with limited shapes, there are many cases that cannot be folded rigidly.

To solve this problem, we apply the spring-mass model[1]. As our model can represent a flexible model, it is possible to simulate non-Rigid Origami. The spring-mass model requires appropriate spring arrangements and the setting of parameters. We propose two different models and we call them the spring-mesh model and the spring-hinge model

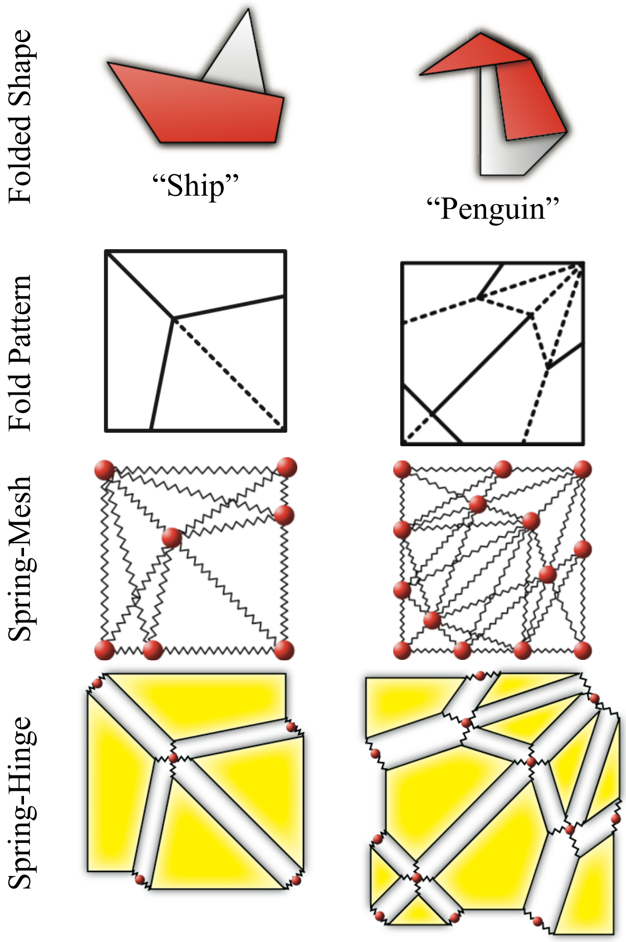

Fig 1:Examples of the spring-mesh model and the spring-hinge model.

(Fig 1). The spring-mesh model is an Origami model that represents faces with spring networks. The spring-hinge model is a model that represents all hinges with springs and faces with rigid polygons. We compared these models and studied each characteristic.

We also propose a method for generating seamless 3D Origami animation. The motions of flaps during folding operation are very complex and we cannot get the animation by simple key-frame approach. In our method, the positions of flaps are generated by key-frame interpolation and then revisions are added to them. The revisions are calculated by using spring-mass simulation during the animation.

We made some 3D Origami animations with our methodand we found that our method can generate natural movement of flaps during the animation.

We introduce related works in chapter 2 and describe the spring-hinge model in chapter 3. In chapter 4, we discuss the animation of Origami and reach the conclusion in chapter 5.

2. Related works

There are various ways of approach to model the restriction and the transformation of Origami. The famous simulating system of Origami, called "Origami Simulation" by Miyazaki[2], applied the 3D system controlled by pointing devices (drag-and -drop function). The fold line isautomatically created when a corner of the paper is pointed with a cursor and then moved, which can move flexibly and the face attached to this follows the movement of the cursor. Further, since the system records each folding process and the shapes of the paper, it can reproduce the series of process with 3D animation afterwards. Thesefold lines, however, can only be created on the same plane. It means that the fold lines across the several planes cannot be simulated precisely, especially under the condition of representing the complicated folding process caused by the complex interaction of each plane.

Tachi suggests one method called Rigid Origami Model regarding the folding paper as the rigid panel and the flexible hinge part[3]. This model enables us to create the animation that all of the planes can be moved at a time by solving problems of rigid kinematics. On the other hand, there are some Origami peaces they cannot be folded rigidly and this system cannot treat these cases.

Although the key frame method is used to fix the time for the animation which divides the process into several frames and calculates the shapes of the mean point at each frame, the simple use of this method cannot avoid making additional errors when target object is Origami since the transformation of the interacting works like Origami cannot be estimated by simple interpolation.

Fujii created the 3D animation of Origami folding using the VRML system[4]. However, handling complex models such as that multiple planes need to move toward different directions at a time are still difficult.

In this way, it turns out that the flexibility of paper is the important key to realize the natural moving in Origami animation.

3. Spring-hinge model

In this paper, we regard a plane surrounded by folding lines or boundary lines as a two-dimensional rigid plane and propose the model in which vertices correspondent to each plane are connected with springs. We call this model "the spring-hinge model" and describe the details in the following subsections.

3.1. Simulation

We define an-sided polygonal plane$R$in three-dimensional space as a set of 4x4 affine matrix **M** and a polygon P placed on two-dimensional plane. **M** contains a transform vector $T=(t_x,t_y,t_z)$ and a 3x3 rotation matrix **Q**. The polygon P is consisted from n two-dimensional coordinates of the unfolded pattern. We assume the scale factor is 1. The coordinates of i-th vertices of R can be expressed as follows.

$$\begin{pmatrix} r_{i,x} \\ r_{i,y} \\ r_{i,z} \\ 1 \end{pmatrix} = \mathbf{M}p_i = \begin{pmatrix} q_{11} & q_{12} & q_{13} & t_x \\ q_{21} & q_{22} & q_{23} & t_y \\ q_{31} & q_{32} & q_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_{i,x} \\ p_{i,y} \\ 0 \\ 1 \end{pmatrix} \tag{1}$$

Each polygonis designated to be connected as much as possible by adjusting the value of the affine matrixduring the movement. To put it concretely, vertices which should be topologically connected are attached with the springs having initial length of 0 (Fig 1). The force f_{ij} acting on the vertices of the polygon R_i is described as follows.

$$f_{ij} = k(r'_j - r_j) + Dv_{ij} \tag{2}$$

Here, k, D and v_{ij} indicate the spring constant, the damper constant, and the relative speed of the vertex r_i to the vertex r'_j respectively and r'_j shows the center of mass of the vertex which should be existed within the same coordinates. The force added to the polygon R_i by every spring connected to this polygon and the torque τ_i around the center of mass are described as follows [5].

$$F_i = \sum_j f_{ij} - m_i g \tag{3}$$

$$\tau_i = \sum_j (r_{oi} \times f_{ij}) - r_{oi} \times m_i g \tag{4}$$

m_i, g and r_{oi} in the equation indicate the mass of the polygon R_i, the gravitational acceleration, and the relative coordinates of the vertex r_i considered from the center of mass, respectively. Each of F_i and τ_i shows the three-dimensional vector and τ_i / $|\tau_i|$ indicates the axis of rotation, while $|\tau_i|$ shows the rotation angle. By giving a time integral to these equation based on Euler method, the speed v_i and the location r_i of the polygon, the angle speed ω_i, and the angle λ_i are calculated as follows.

$$v_i(t+\Delta t) = v_i(t) + \frac{F_i(t)}{m_i}\Delta t \tag{5}$$

$$r_i(t+\Delta t) = r_i(t) + v_i(t)\Delta t \tag{6}$$

$$\omega_i(t+\Delta t) = \omega_i(t) + \frac{\tau_i(t)}{I_i}\Delta t \tag{7}$$

$$\lambda_i(t+\Delta t) = \lambda(t) + \omega_i(t)\Delta t \tag{8}$$

I_i indicates the value of inertia tensor of the polygon R_i. Translation r_i and rotation λ_i are composed into a 4x4 matrix Π_i as follows.

$$\prod_i = \begin{pmatrix} s\mathbf{Q}(\lambda_i) & (1-s)T_i \\ \mathbf{0} & 1 \end{pmatrix} \tag{9}$$

$$T_i = \begin{pmatrix} r_{i,x} \\ r_{i,y} \\ r_{i,z} \end{pmatrix} \tag{10}$$

s in (10) shows the ratio of rotation components held in the matrix and the extent is $0 \le s \le 1$. Our system designates 0.7 since rotation component takes high percentage in Origami movement. **Q** is the function that replaces rotational vector with a 3x3 matrix, which is defined as the following equation [5].

$$\mathbf{Q}(v) = \begin{pmatrix} C + v_x^2(1-C) & v_x v_y(1-C) - v_z S & v_x v_z(1-C) + v_y S \\ v_x v_y(1-C) + v_z S & C + v_y^2(1-C) & v_y v_z(1-C) - v_x S \\ v_x v_z(1-C) - v_y S & v_y v_z(1-C) + v_x S & C + v_z^2(1-C) \end{pmatrix}$$

$$C = \cos|v|$$

$$S = \sin|v| \tag{11}$$

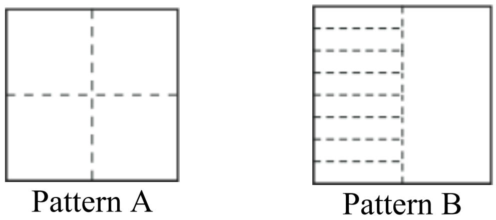

Fig 2:The crease patterns of our evaluation

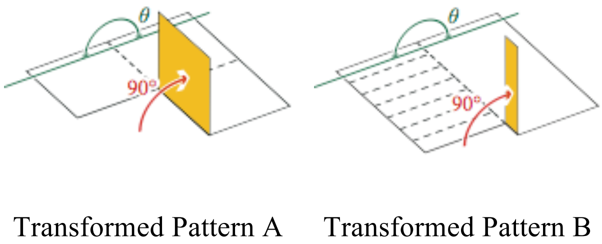

Fig 3:Transformed patterns that they folded at the crease line with 90°

Affine matrix $\mathbf{M}_i(t+\Delta t)$ of the polygon R_i is calculated with the following equation.

$$\mathbf{M}_i(t+\Delta t) = \Pi_i \mathbf{M}_i(t) \tag{12}$$

By alternately calculating the above equation in a very short interval, the moved positionof the polygon is calculated.

3.2. Evaluation

We implemented the simulation programs of both of “the spring-mesh model” and “the spring-hinge model” with Java and used Java3D Library for 3D expression. We compared the degree of their convergence after transformation by inputting the transformed figures such as Figure 3 that have the crease patterns of Figure 2, respectively and also calculating the angle θ of the opposite plane. The angle of θ is expected to convergeto 90 degrees as time passes.

The paper is a unit square and the spring constant, the damper constant, and the time interval are designated as 1, 1, and 0.01, respectively. Though the concept of a particle is different between these models, the total mass existing in the paper is adjusted to 1. That is, the weight of the vertex in the spring-mesh model is $1/n$ and that of the polygon in the spring-hinge model is S_i, same value with area.

The following computational environments are used for the experiment.

Model	Apple MacPro
CPU	Intel Dual-Core Xeon 5150 (2666MHz) x2
RAM	PC2-5300 DDR2-FBDIMM (667MHz 9GB)
GPU	NVIDIA GeForce 7300GT (VRAM: 256MB)
OS	Mac OS X 10.5.1
	JDK 1.5.0_13 + Java3D 1.5.1

3.3. Result

Figure 4 and 5 show the graphed time variation of θ. The following tendency can be read from this graph.

1. The solution converges in faster speed in the spring-mesh model.
2. The time needed for one step is shorter in the spring-mesh model.
3. The spring-hinge model converges the solution to the expected value.

We can confirm from the graph A and B that the spring-mesh model converges rapidly. The short interval of the marker tells us that the time per one step is short. On the other hand, though the operation time is long on the

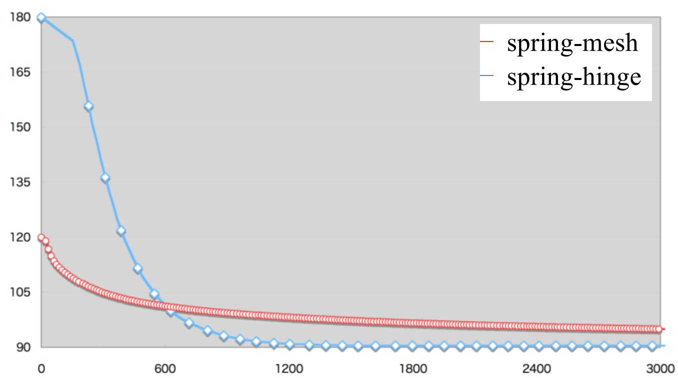

Fig 4: The graph of the angle θ of the Transformed Pattern A

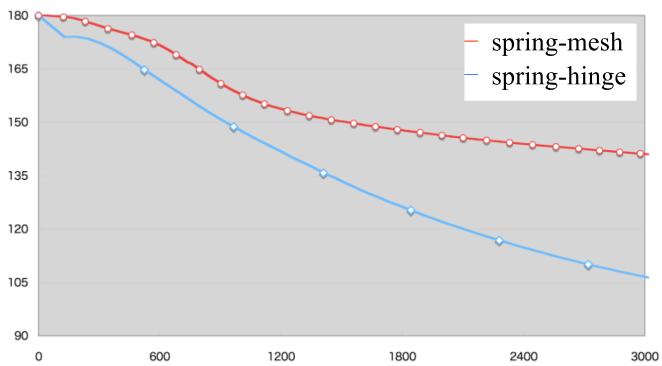

Fig 5: The graph of the angle θ of the Transformed Pattern B

grounds that the algorithm of the spring-hinge model is more complicated than that of the spring-mesh model, the solution converges in fewer repetitive times. Moreover, it becomes clear that the ideal solution close to 90 degrees can be obtained in the spring-hinge model.

4. Animation

The key frame method is generally used for the creation of the animation which requires predefined shapes and time intervals. Though the key frame method can create the animation with less input, it brings about the problem that we need to adjust the key frame and the interpolation function through trial and error. Without this correction, it often creates the unnatural result such as that the figure of the plane may be twisted or some space may be found between the polygons, if the interactive transformation like Origami, especially, is interpolated with the simple function. In this chapter, we discuss how to solve this problem.

4.1. Method

When some space is found between the polygons at the time of interpolating of the key frame, we propose to adopt the method to fill in the space caused by the stretch of the spring among vertices, which is described in the chapter 3, at every frame. The flow of our method is as follows.

1. Based on the next key frame and the current intermediateframe, calculate the position and the direction of the mass in the intermediate frame by interpolation.
2. Bring the position of the plane close to the correct position using the spring force.

The quaternion of the mass position of the polygon and the direction of movement is supposed to be linearly interpolated here.

4.2. Evaluation

We prepared the three frames expressing *Petal fold* as shown in Figure 6 and with dividing them into 60 parts, we calculated the mass position and the direction of the polygon by the simple linear interpolation. After that, we compared the result in which rendering is done using each frame without any correction and that in which rendering is done after the error correction using the spring operation of ten times.

4.3. Result

The result obtained from the above experiment is shown in Figure 7 and 8. In the simple linear interpolation, penetrationsof the polygons from the frame 2 to the frame 3 and the space between the polygons can be seen. On the contrary, we see no penetrations of the polygons and hardly find any space in Figure 8 which was corrected with the spring operation of ten times. We think that such a little space can be filled with the simple filling process at the time of rendering. The time needed for the spring operation was 4-7ms and it finished in half time of 16.67ms or less which is the general frame interval of 60fps CG animation. We can think that the space can be made smaller with longer operation time.

5. Conclusion

In chapter 3, we proposed “the spring-hinge model” based on the spring-mass model and compared it with the existing method of “the spring-mesh model.” By the experiment, it became clear that the spring-mesh model can calculate in higher speed, while the weakness for the twists was confirmed. Moreover, in this spring-mesh model, it is supposed that the calculation becomes difficult when we try to add collision detection of the polygonsbecause of the twists of it. In the spring-hinge model, as the polygons are always confirmed to be flat, it will be possible to add collision detection with adding thickness to the plane.

In chapter 4, we proposed the method to animate the process for transformation of Origami with 3DCG by the key frame interpolation. As each polygon of Origami makes transformation with complicatedly affecting each other, we need to adjust the motion at all times in the existing method. By using our method, the natural and smooth animation can be created by a little information. But in case of the complicated figure, it sometimes occurs that the created animation becomes unnatural because the correction to the interpolated value is insufficient. The closer interpolation function to the transformation of

Fig 6: The key-frames

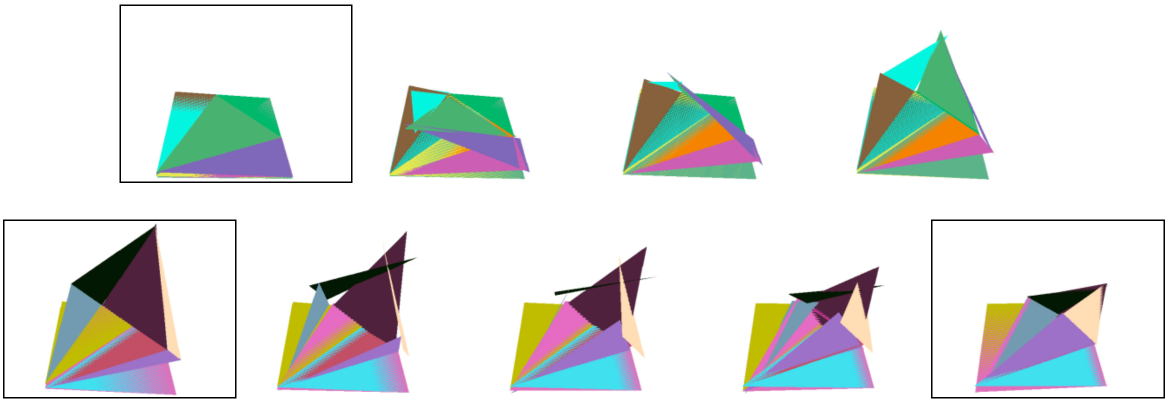

Fig 7: The positions of flaps generated by simple key-frame interpolation.

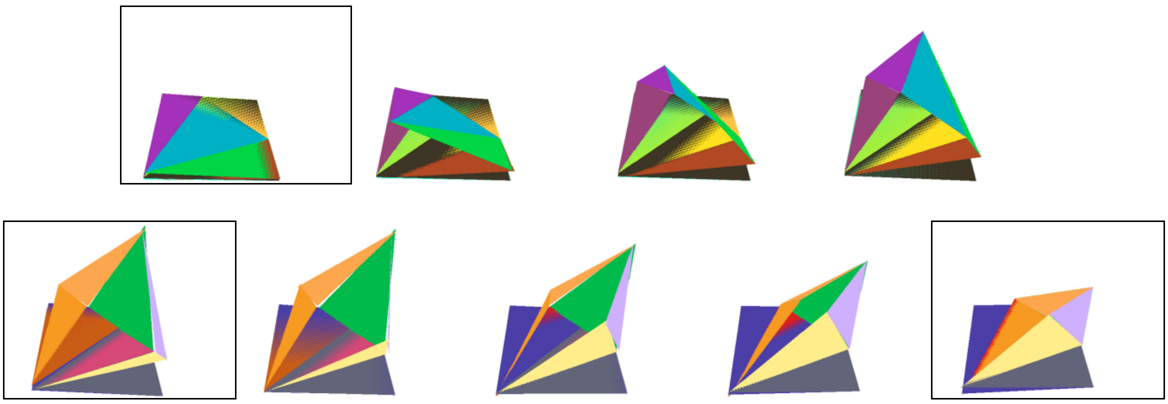

Fig 8: The positions of polygons generated by key-frame interpolation and then revisions are added to them.

Origami is needed to be found.

Reference

[1] FURUTA Y. , KI MOTO H. , MI TANI J . , F UKUI Y. : Computer model and mouse interface for interactive virtual origami operation. IPSJ Journal 48, 12 (Dec. 2007), 3658–3669.

[2] MIYAZAKI S. , YASUDA T. , YOKOI S. , TORIWAKI J . : An origami playing simulator in the virtual space. The Journal of Visualization and Computer Animation 7, 1 (1996), 25–42.

[3] TACHI T. : Simulation of rigid origami. appear in proceedings of the 4OSME, Pasadena USA.

[4] FUJII Y., KISHI N.: Origami Browser Using VRML on the Web. IPSJ SIG Notes 97, 43 (1997), 19-24

[5] Eric Lengyel: Mathematics for 3D Game Programming and Computer Graphics, Charles River Media, 2001