Origami Diagrams and 3D Animation from Flat-Foldable Crease Patterns Sequences

Hugo Akitaya [1], Jun Mitani [1], Yoshihiro Kanamori [1], and Yukio Fukui [1]

[1] Graduate School of Systems & Information Engineering, University of Tsukuba, Japan

Abstract

Origami is the art of folding paper. It is a simple way of constructing geometric shapes from a single sheet of paper without using any cuts. There are two usual forms to document an origami model; diagrams and crease patterns. Diagrams are the step-by-step sequences that can be found in traditional origami books while crease pattern is the pattern of creases left on the paper after folding an origami model. The disadvantage of crease patterns is that it is difficult to use them to re-create the design, since crease patterns show only where each crease must be made and not a step-by-step instruction. However, drawing diagrams is tedious and very time-consuming. We propose a method to autonomously generate the corresponding diagram for a crease pattern sequence. Each element of the sequence represent the state of the paper in a certain origami step. We construct a sequence of steps, also generating an animation to show the transition between steps, thus helping novice origamists.

Keywords – computational origami, origami rendering, rigid origami simulation

1 Introduction

The most common form to convey origami is through origami diagrams, which are step-by-step sequences, composed of figures that represent the state of the folded paper, in combination with lines and arrows indicating the position of the folds and the movement of the paper. An example of a typical origami step represented as a diagram can be seen in Figure 1a (second figure from left to right). It is usual for diagrams of an origami model to have a large number of drawings. For that reason, drawing diagrams is tedious and very time-consuming.

With the development of modern techniques of origami design, the range of achievable shapes increased drastically and the crease pattern (the pattern of creases left on the paper after folding an origami model) has gained importance as an efficient method of documenting origami pieces. Figure 2 shows an example of crease pattern (2a) and its folded form (2c). Several authors publish only crease patterns of their creations. However, the disadvantage of crease patterns is that it is difficult to use them to re-create the design, because they show only where each crease must be made and not a step-by-step instruction. There are two types of fold in origami; mountain folds and valley folds. The mountain fold produces a convex surface and is represented here as dashed red lines. The valley fold produces a concave surface and is represented here as solid blue lines.

We introduce a method to generate diagram notations from sequences of crease patterns of flat origami comparing the position of vertices and creases between two consecutive crease patterns, thereby providing a tool to accelerate origami documentation. Distortions can also be added to the drawings in order to clarify the layer arrangement of the origami. Our system provides 3D animation of simple steps, as an aid to those inexperienced in folding origami.

2 Related Work

Modern origami diagrams have notations based on the system developed by the origami artist Akira Yoshizawa. Originally, they were hand written, but, gradually computer-aided origami diagramming became the standard way to represent origami instructions. Although there are variations on diagramming symbols, diagrammers usually follow some standards [1].

The rendering of a flat origami from its crease pattern, which is a NP-complete problem, has been implemented by a software program called ORIPA [2], which uses brute force to determine the ordering of the layers. Our program uses this method to obtain the folded form of the crease pattern. A method for simulating rigid origami (origami in which the faces are made of a rigid material) was developed [3] by using affine transformations to map the movement of the faces, while the dihedral angles between the faces are calculated by using Euler integration to solve a system of differential equations for each vertex. The relationship between spherical geometry and the dihedral angles of an origami model with a single vertex was found to produce noncrossing motions [4].

3 Origami Diagrams

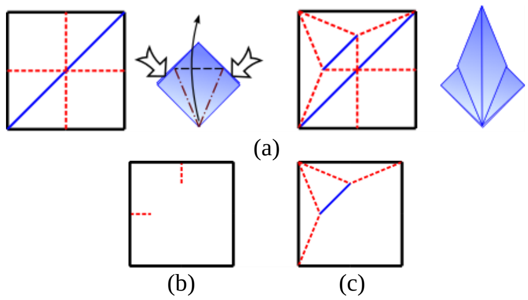

Fig. 1. Generation of origami notations. (a) Crease pattern and folded form for two consecutive steps. Folding arrows are obtained by comparing the position of vertices. (b) Removed creases will be marked with a push arrow. (c) Added creases will appear as fold lines.

3.1 Diagrams` Notations

Origami diagrams notations are the arrows and lines showing where the folds must be performed as well as how the paper should move. In the diagram notation used in this work, the mountain folds are represented as dark red dot-dashed lines and the valley folds as black dashed lines.

By comparing the crease pattern of a step with the crease pattern of the subsequent step we can obtain a set of removed creases (creases present only in the current step) and a set of added creases (creases present only in the next step). An added crease should appear in the diagrams as a fold line and rendered just above the face in which it is embedded. The orientation and position of such fold will follow the affine transformation that controlling the position of the host face. A removed crease is marked in the diagrams with a *push arrow*. Push arrows are the white fat arrows that indicate push action used to unfold a certain crease while inverting the orientation of the faces divided by this crease.

We can also compare the position of vertices after the execution of the step. We denote the movement of the paper with a *folding arrow* (slim black arrow), as shown in Figure 1a. Our system will generate as many arrows as the number of moving vertices. If fewer arrows are desirable, the user can simply delete the unnecessary notation by editing the results.

3.2 Distortions

Usually, diagrams have some distortions to show more clearly how the layers are disposed in the origami model [1]. If the result form of the model shown in Figure 2a were ideal, its rendering would show only a square (Figure 2b), and information about how many layers are piled up would be lost. To perform such distortions, we consider the layer ordering of the origami. The layer ordering is obtained using ORIPA methods, by brute force. Then, we can attribute a z-index to each face considering that successive faces that do not intercept in the x-y plane should have the same z-index. Each vertex must also have a z-index, calculated as the arithmetic mean of the z-indexes of the faces that contain such vertex, and then adjusted to avoid self-penetrations (Figures 2d, 2e and 2f). After this adjustment, the position of the vertex is calculated using the following formula, in which p_v^* represents the distorted position of vertex v, p_v is the original position of vertex v, p_o is the vector representing the viewing direction, z_v is the z-index of vertex v and z_{max} is the maximum z-index.

$$p_v^* = p_v + p_o \cdot z_v / z_{max} \qquad (1)$$

Figure 2c shows the result obtained by our system with distortions. The user, in order to generate different perspectives and show the desired layer configuration, can change the viewing direction and the amount of the distortion by inputting a different p_o.

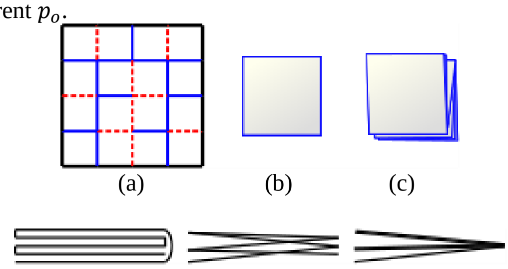

Fig. 2. (a) Crease pattern and its folded form: (b) ideally mathematical model and (c) distorted model. (d) Transverse section of an origami showing the layer z-index. (e) Vertex z-index as the mean of faces z-indices. (f) Adjusted vertex z-index to avoid self-penetrations.

4 3D Animation

We use the method described in [3] to determine the affine transformations that govern the movements of the origami faces. First, we determine the set of faces that will move by comparing the position of faces in two consecutive steps. Then, we can gradually vary the dihedral angles between the faces until their desired positions. Each face is considered to be rigid and the solution of the necessary condition for the angles (to maintain the continuity of the paper) is calculated using spherical geometry, in a process similar to the one described in [4].

The degree of freedom for the movement at a vertex is the number of moving faces minus two [3]. When the degree of freedom is less than one, our system allows deformations on the neighbour fixed faces in order to represent the movement preserving the continuity of paper. Figure 3 shows screenshots of an example of animation.

The animations can contain self-interceptions, since the conditions for angles are not sufficient to prevent collisions and some steps cannot be performed with the rigid origami approach. The results of the animation can also be exported to a 3D computer graphics software, so that the position of vertices can be edited in order to create more realistic animations or intermediate 3D steps for origami diagrams.

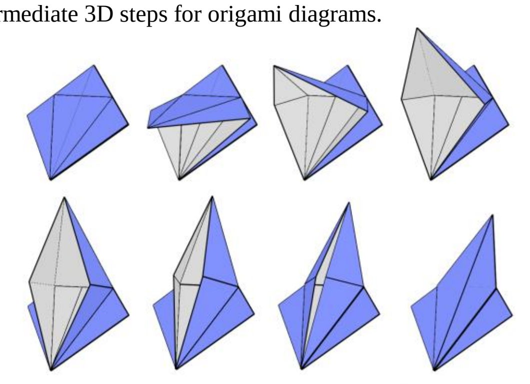

Fig. 3. Generated 3D animation of an origami step.

5 Conclusion

We presented a method to autonomously generate origami diagrams from crease pattern sequences. The crease pattern sequences can be obtained by a simplification method as described in [5], or manually created using ORIPA. The use of the output can decrease substantially the time spent to draw origami diagrams. The results are exported in a vector graphics file format, allowing easy alterations and adjustments to the desired notation. However, some notations are not generated by our method such as mountain fold arrows and, as this method is only applied to flat origami forms, the 3D intermediate steps that appear in some diagrams cannot be generated using ORIPA. To obtain such intermediate steps, the user can use screenshots of the 3D animation.

As future work, we plan to add support for more notation symbols and implement improvements in the distortions. In a symmetrical model, for example, there might be different vertices in the same position with the same z-index on the symmetry axis. These vertices' distorted position will be the same, although they are usually represented slightly apart from each other in diagrams. Creating a dynamic p_o pointing outwards from the symmetry line is a possible solution.

The z-index stacking of faces only work if there are no cycles in the stacking order of faces, which occurs when different regions of the same face requires different stacking order to be correctly rendered. The division of faces in smaller regions for the calculation of z-index and rendering is a way to produce correct representations for these models.

References

[1] R. J. Lang, Origami Diagramming Conventions, available at: http://www.langorigami.com/diagramming/diagramming.php.

[2] J. Mitani. Development of origami pattern editor (oripa) and a method for estimating a folded configuration of origami from the crease pattern. IPSJ Journal 48, pp. 3309-3317, 2007.

[3] T. Tachi. Simulation of rigid origami. Origami 4, Fourth International Conference on Origami in Science, Mathematics, and Education (4OSME), A K Peters, pp. 175-187, 2009.

[4] I. Streinu and W. Whiteley. Single-vertex origami and spherical expansive motions. Proc. of the 2004 Japanese conference on Discrete and Computational Geometry, Springer-Verlag, Berlin, Heidelberg, pp. 161-173, 2005.

[5] H. Akitaya, J. Mitani, Y. Kanamori and Y. Fukui. Generating Folding Sequences from Crease Patterns of Flat-Foldable Origami. To appear in ACM SIGGRAPH 2013 Posters, Anaheim, CA. 2013.