An Exploring Tool for Simple Flat Origami based on Random Foldings

Naoya Tsuruta [1], Jun Mitani [1], Yoshihiro Kanamori [1] and Yukio Fukui [1]

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

Abstract

In flat-foldable origami, design techniques represented by the circle-river packing approach are effective for complex models but are not suitable for simple models. In this paper, we propose a system for generating simple origami pieces that can be made with a samll number of folds. A fold in origami can be classified into two categories: *exact fold*(if its position is based on exact reference landmarks) and *inexact fold* (if its location is not based on exact reference landmarks). Our system calculates exact folds based on origami constructions and inexact folds by moving the exact ones with the Monte Carlo method. The system displays dozens of tiny folded origami pieces at once. By selecting a desired model, the user can obtain the origami instructions to recreate it on real paper.

Keywords - Origami, Origami Constructions, Monte Carlo Method

1 Introduction

Origami is the art of paper folding. Folding objects from a sheet of paper has been a research topic in mathematics and engineering, resulting on the introduction of technical origami design. Using geometrical and mathematical principles to construct a crease pattern, it is possible to design complex origami models. An example of a robust algorithm for designing flat foldable origami is the circle-river packing [3] (also known as tree method). The algorithm places a weighted tree graph into a square paper and calculates a crease pattern for the base of the desired model. However, these origami design techniques usually result in complicated crease patterns, which are not suitable for simple origami models. To address this problem, we have proposed a system for generating simple origami pieces using enumeration of folded shapes. The system enumerates folded shapes that can be accomplished within four folds by limiting folding operations. Among them, we can find new computer generated origami models (Figure 1). It is hard to enumerate shapes constructed with more than four folds because of the combinatorial explosion and it is impossible to enumerate all possibilities when inexact folding is considered. In this paper we propose a new system for supporting creation of simple origami models with the Monte Carlo method.

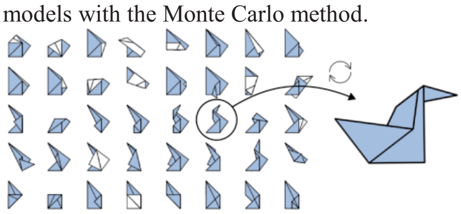

Fig. 1. Example of enumerated folded shapes and an origami model of a bird found within it. [2]

2 Our Proposed System

Our system randomly generates folded origami pieces, using the theory of origami geometrical constructions. Here we describe how to generate them automatically. The generation process takes three steps.

2.1 Arrangement of folds

First we decide where to fold by calculating the location of a fold. A fold is classified into two types by whether it has an exact location point or not. We refer them as an exact fold and an inexact fold respectively hereinafter. Exact folds, such as folds that place a corner onto another corner, are usually used in the beginning of the folding processes. Inexact folds are often used in latter steps. Therefore, fully random folding is not suitable for creating origami models and we have to mix these types. Our system decides the position of exact folds based on the theoretically known origami folding operations. With only two following folding operations it is possible to recreate any geometric construction [1].

1. Given some lines, to locate the points of intersection
2. Given two points $p1$ and $p2$ and two lines $l1$ and $l2$, to fold a line placing $p1$ onto $l1$ and placing $p2$ onto $l2$ (Figure 2).

By applying this origami operation, we obtain the exact folds given the available reference landmarks. An initial state of origami paper has four corners and four lines (contour edges) as reference landmarks. The system calculates the location of a fold by picking up two corner points and two lines at random. From a set of points and lines, at most three solutions are obtained and the system randomly chooses one. After the operation, the number of available landmarks will be increased. The intersections of contour edges are also used as landmarks.

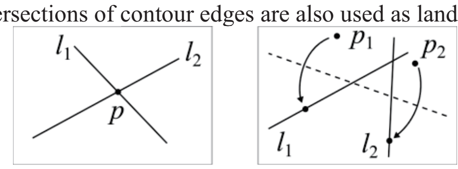

Fig. 2. Two origami operations.

2.2 Modifying Folds

In the latter steps, we slightly move the exact folds in order to make them inexact. We realize this by moving two endpoints, s and v, of the fold in random directions, θ_s and θ_v, by a specified amount r as shown in Figure 3. Our system applies this randomization to the latter N steps, with N specified by the user. The numbers of landmarks is greatly increased by the addition of inexact folds, because it causes the corners and lines not to coincide. Therefore, this method contributes to the variety of the folded shape.

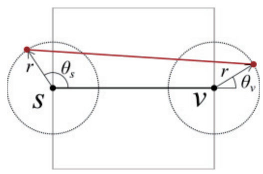

Fig. 3. Moving the endpoints of a fold in random directions.

2.3 Enumeration of Possible Folded Shapes

After the location of a fold is set, we decide how many layers to fold. Only simple valley and mountain folds are used in the system. In case of valley fold, our algorithm tries to perform the fold from the top layer of the origami model. Here, we have to take care about the probability of possible folded shapes. Figure 4 shows an example of a fold and its possible result shapes. Although the fold shown in Figure 4(a) intercepts three layers, there are only two possibilities for the folded shapes. For that reason, we cannot choose randomly the number of layers in which the fold will be applied, but we can enumerate the possible folded shapes and then choose one of them at random.

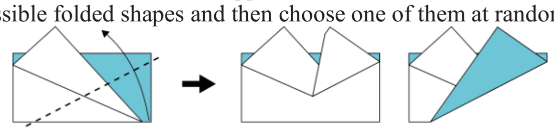

Fig. 4. Difference between the number of layers and the numbers of possible folded shapes. Figure (a) has three layers and two possible shapes.

2.4 User Operations

The user specifies the total number of folds, the number of inexact folds, and the length of the parameter *r* described in section 2.3. Our system displays the folded shapes as shown in Figure 5. The user can choose one among them if it looks like a desirable shape such as an animal, a flower, etc. If there is no interesting shape, the user requests another round of origami pieces. Initial colour of paper is randomly set by the system but it can be changed after the user's choice.

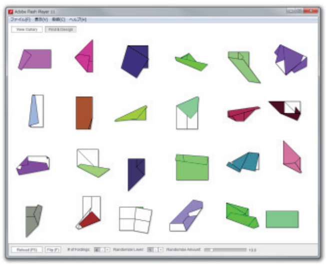
Fig. 5. Interface of the proposed system.

3 Results and Discussions

Figure 6 shows examples of models that we have found using our system. Our system keeps the folding sequences of each generated model and generates its diagram so that the user can reproduce a physical model. The system generates twenty-four pieces at once (by default) and runs in real time on a standard PC when the total number of folds is within six. The generation of models with six folds requires a few seconds to compute. Reducing the number of pieces that are displayed at once is a simple way to improve performance.

We have also tried to find shapes that resemble existing origami models designed by origami artists. Figure 7 shows a well-known simple origami model '2 fold Santa' by Paula Versnick and an origami piece generated by our system. The first fold of Paula's Santa is made with inexact folds but with a limited degree of freedom as shown in Figure 8. Although a fold which places a point onto a line has infinite solutions, it has only one degree of freedom and, for that reason, results on more formatted shapes than fully randomized folding. There are several types of limited inexact folds, and implementing them leads to generation of more attractive origami pieces.

4 Conclusion and Future Work

We have proposed a system for exploring new origami pieces based on the Monte Carlo method. Randomized folding makes a variety of folded shapes and the user can get his or her unique origami models by simply choosing a model from the generated pieces. We are now running our system on the web (http://www.npal.cs.tsukuba.ac.jp/~tsuruta/origaminista/) and collecting the data from the online users. Finding a strategy for generating meaningful models by analysing data is our future work.

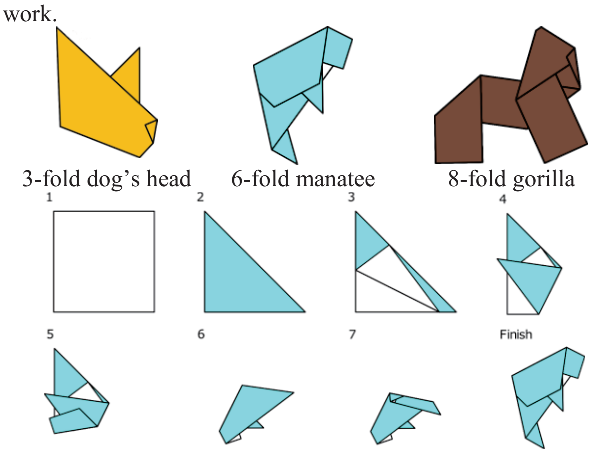

Fig. 6. Examples of generated origami models and a diagram.

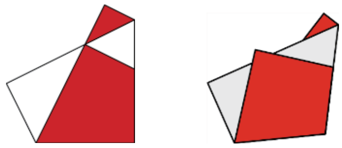
Fig. 7. Paula Versnick's 2 fold santa (left) and a generated piece similar to it (right).

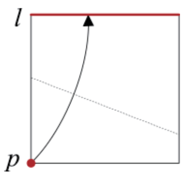

Fig. 8. A fold that places *p* onto *l* is an inexact fold but makes a more formatted result than a fully randomized fold.

Another future work is to modify generated pieces. An *n*- fold origami piece (origami piece constructed with *n* folds) is defined by an initial unfolded model and *n* folds. Several similar pieces can be generated by moving folds slightly so that the user can pick up the best one. A more intuitive user operation is that the user directly edits the contour of the generated piece. We show an image of contour editing in Figure 9. Trimming a contour can be solved by adding a fold (Figure 9(b)). On the other hand, modification of the contour (Figure 9(c)) is hard to solve because moving a fold may cause changes in whole shape of the model. We expect that this can be achieved with an optimization approach to minimize the changes.

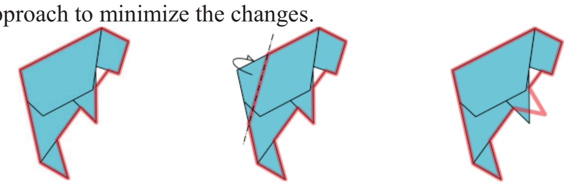

Fig. 9. Image of contour editing. (a) A piece generated by the system and its contour (thick red line). (b) Trimming of the contour and an additional folds fitting to it. (c) Modification of the contour.

References

[1] Koshiro Hatori, K's Origami : Origami Construction, http://origami.ousaan.com/library/conste.html

[2] Naoya Tsuruta, Jun Mitani, Yoshihiro Kanamori and Yukio Fukui, A Computer Aided Design system for Simple Origami Pieces Based on the Enumeration of Folded Shapes, Science of Origami, vol. 2 No. 1, pp33-44, 2012.

[3] Robert J. Lang, Origami Design Secrets: Mathematical Methods for an Ancient Art, A K Peters, Natick, MA, 2003.