Recognition of Folding Process from Origami Drill Books

Hiroshi Shimanuki
Graduate School of Engineering,
Nagoya University, Japan
simanuki@watanabe.nuie.nagoya-u.ac.jp

Jien Kato   Toyohide Watanabe
Graduate School of Information Science,
Nagoya University, Japan
{jien, watanabe}@is.nagoya-u.ac.jp

Abstract

This paper describes a framework to recognizing and recreating folding process of origami based on illustrations of origami drill books. Illustration images acquired from origami books are motley and not sequenced. Moreover, the information obtained from 2D illustrations is so superficial and incomplete that the folding operations cannot be determined uniquely. To solve these problems, a highly flexible and reliable recognition mechanism is proposed in this paper.

This paper additionally includes the content as follows. Firstly, an algorithm for revising the positions of folding operations extracted from illustrations is proposed so as to make our recognition approach more reliable. Secondly, the outline of the methods which enable feasible folding operations to be generated based only on superficial and incomplete information extracted from illustrations is described. Finally, some updating procedures are proposed to maintain consistency of data (called internal model) which record the transformation of origami models in 3D virtual space during a folding process. Several examples that prove the validness of proposed algorithms/methods are also given in this paper.

1. Introduction

Origami has been played in Japan since ancient times. Its fineness and amusement are acknowledged by not only Japanese but also entire world. Various origami models have been created, folded and handed on for hundreds of years. And in addition, many origami models are still announced by creative origami writers as origami drill books now.

However, it is somewhat difficult to understand origami drill books for some people, especially for children, because in origami drill books a folding process for an origami model is usually explained by a sequence of illustrations that cannot show all the portions of the model. To deal with this problem, it is desirable to represent how an origami model is folded and how the situation of the model changes in three-dimensional virtual space.

Miyazaki et al. [3] propose a visual simulation system of origami which realizes interactive folding operations in 3-dimensional virtual space. From a totally different viewpoint, in this paper we propose a system which is able to recognize folding process of origami from a sequence of origami illustrations automatically, and moreover recreate all the folding steps during the process with 3D CG animation.

2. Framework

Figure 1 shows the framework of recognizing folding process in our system. This system is primarily composed of three units, extraction of graphic elements, generation of folding operations and recognition of folding operations, together with an internal model and a CG simulator. As to extraction of graphic elements, we have proposed a method [5] that successfully extracts five types of graphic elements (i.e. illustrations, explicative sentences, step numbers, special symbols and sketches) from a page-image of origami books, and groups different types of elements so that each cluster is enabled to correspond with a single folding step (see Fig.2).

Another method has also been developed to extract edges of origami and fold-lines within the scope of one illustration. The fold-lines extracted within one illustration are essential to generating folding operations, because ideally the fold-lines tell positions of folding operations. However since they are usually inaccurate, a revision processing, described further in Section 3, has to be applied.

Generating folding operations means to create all the resulting creases on faces of an origami model in 3D virtual space. Obviously, the information obtained from 2D plane figures (illustrations) such as the positions of folding operations described above is superficial and incomplete. Given such information, many interpretations about the way of folding including infeasible ones can be made. As the result, the creases cannot be determined uniquely. From this viewpoint, the unit of generation of folding operation is de-

signed to play the role to create only feasible ways and exclude all infeasible ways of folding. Some effective methods by maintaining consistency of crease patterns under geometrical constrains have been proposed in [4] and the outline of these methods will be concisely described in Section 4.

The unit of recognition of folding operations is designed to find out most likely folding operation from multiple candidates that are selected by the unit of generation of folding operation. The potential ways of folding (candidates) are simulated against an application-specific data structure named internal model, which records the transformation of a 3D origami model under folding operations step by step. Updating methods of this data structure will be described in Section 5.

The result of simulation, called ISG (Ideal Shape Graph), is compared with the result of graphics recognition, called USG (Unrefined Shape Graph) which corresponds to the next origami illustration. ISG is generated by projecting the internal model onto a 2D plane from the same viewpoint as that of the next illustration, while USG is straightforwardly composed of the edges of origami obtained at the phase of extracting graphic elements. The folding operation relative to the ISG that most agrees with the USG is finally chosen as the way of folding for the next step. Details about this unit has been described in [2].

In such an approach, a sequence of "right" folding operations defined on the internal model (i.e. a folding process) can be generated repeatedly and eventually recreated by a CG simulator.
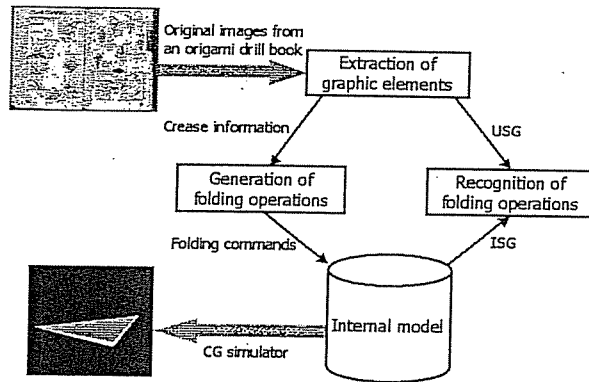

Figure 1. Framework of recognizing folding process of origami.

3. Revising Positions of Folding

Recall that the information about the position of a folding operation obtained by matching an ISG to a USG is not necessarily correct, because this information essentially comes from an illustration. In the situation of dealing with
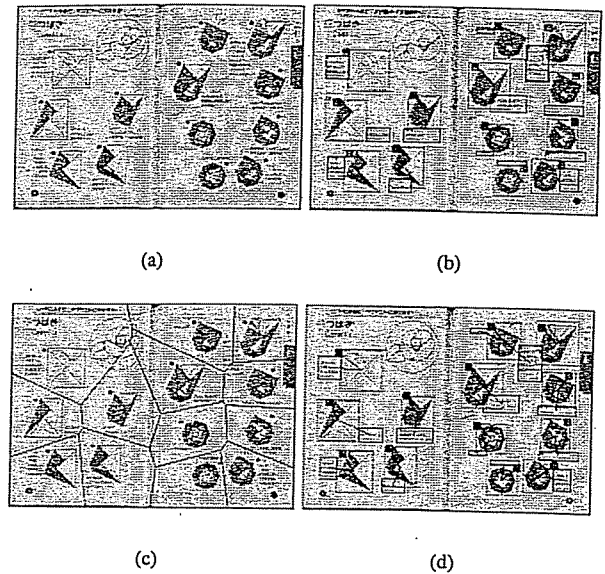

Figure 2. The result of extracting graphic elements. (a) One original page-image of origami drill books. (b) Extraction of various graphic elements. (c) Division into small regions using Voronoi diagram. Each region approximately corresponds to a single folding step. (d) Grouping different graphic elements for one folding step, using the results of (c) together with spatial relationships among the elements.

complicated ways of folding, a little error may lead to such an awful precision degeneration of operation generated that the next folding becomes infeasible. On the other hand, even if the error is so small that there is nothing the matter with the next operation, a revision process is needed, in order to build beautiful origami models which can be good examples for users.

In practice, folding operations are usually performed in accordance with some rules, for example, "fold up the paper so that the two creases fit each other" or "fold back the paper so that the two vertexes fit each other". Irregular ways of folding without any restraints probably exist, but must be rare. From this viewpoint, we formulate the relationships among creases and vertexes within an internal model into four types and describe each type with a rule as follows. These rules are based on origami axioms [1], similar to geometrical operations that can be performed by using only a straight edge and a compass (SE & C). The obtained positions of folding operations using graphics recognition methods are revised by applying one or more available ones of these rules.

Revision Rules

1. A crease has two end points p_1 and p_2.

2. A crease is a perpendicular bisector of two points p_1 and p_2.
3. A crease is a bisector of an angle α.
4. Given a line l_1 and a point p_1, a crease is perpendicular to l_1 and passes through point p_1.

Four examples of the position revision which are related to rules 1-4 are shown in Fig.3, respectively. Figure 3(a) means that if the initial position of a folding operation, namely, the end points of a fold-line extracted are very close to two vertexes of the internal model (ISG), the position of folding operation will be moved so that the resulting crease will pass through these vertexes. Other examples can be interpreted in an analogous way.

Notice that in some examples several rules are available, for example, rules 1-3 are all available for the example shown in Fig.3(c). The order of applying multiple rules is intractable. To solve this problem, we propose an algorithm given in Fig.4 which provides a realizable policy to decide the order of priority of competing rules, based on the number of end points of a fold-line coincided with vertexes of the internal model. In Fig.4, C1, C2, C3 and C4 stand for the revision processing according to rules 1-4, respectively.
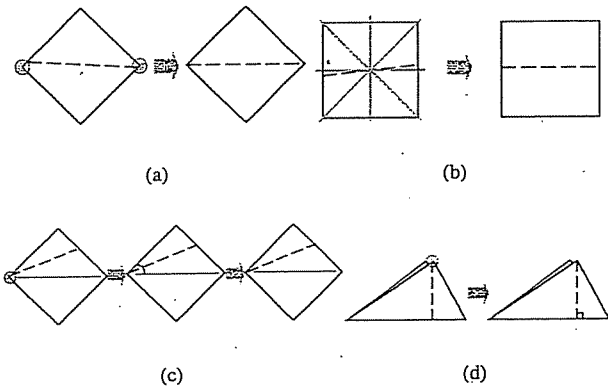

Figure 4. Revision algorithm.

(a) (b)

(c) (d)

Figure 3. Revision rules. The examples (a)-(d) correspond to rule 1-4, repectively.

4. Generating folding operations

We have already proposed some methods for generating folding operations using only incomplete position information of fold-lines extracted from origami illustrations [4]. By use of these methods, all the feasible folding operations can be constituted.

Generation of creases resulting in a folding operation is performed through an unfolded plan, a sheet of square paper with creases made by simply extending an origami model. The creases on an unfolded plan caused by one folding operation are always origami-symmetric with other creases.
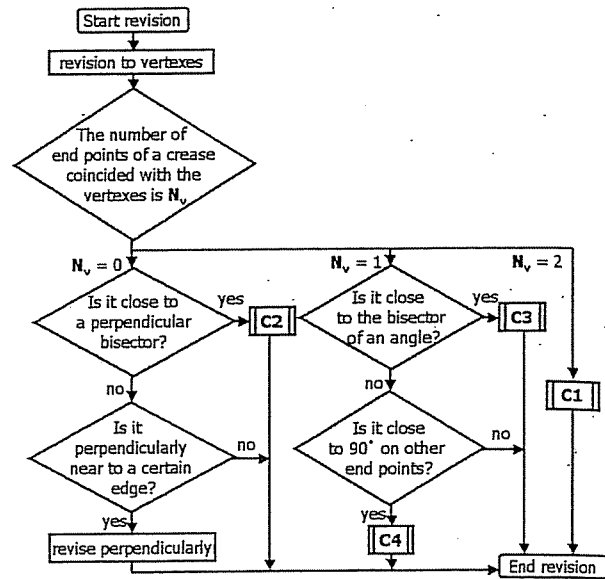
So, although the position information of fold-line extracted is superficial and incomplete, the creases on some invisible faces can be generated using our methods because our methods have taken advantage of origami-symmetry between the creases.

Figure 5 shows a simple example of "valley folding" and generated creases on unfolded plans. If the fold-line which indicates the position of "valley folding" can be extracted from the illustration, the crease resulting on $Face1$ can be generated with ease, however, the crease on the face connected with $Face1$ is invisible from related 2D illustration or related ISG. So, in our approach folding operations are generated by the following steps. The creases whose position can be decided by extracted fold-line information, such as l_0 which is decided directly by the end points p_0 and p_1 of the fold-line, are generated first. Then, the creases origami-symmetrical with generated new creases against some existing creases, such as l_1, are also generated. In Fig.5, we obtained a set of generated creases $L = \{l_0, l_1\}$.

There is another possibility that from the extracted fold-line information we can not determine the direction of the folding operation. So, the resulting crease pattern corresponding to "mountain folding" may possibly be generated. Applying a procedure for checking feasibility called method of origami-section [4] to the example shown in Fig.5, the hypothesis of "mountain folding" turns out to be false (infeasible).
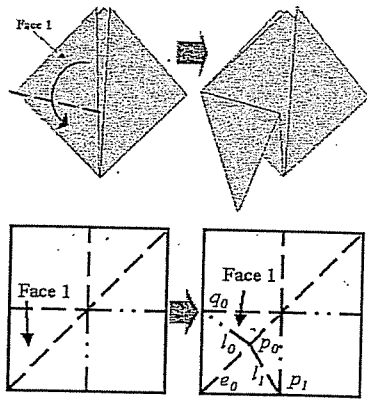
Figure 5. An example of generating "valley folding" and its corresponding unfolded plans.

Figure 6. Updating a face list in case of "valley folding".

5. Updating Internal Model

The internal model is primarily composed of a face tree, a node indicates a face and a branch indicates connectivity relationships among the faces separated by one folding operation. In addition to connectivity relationships, another kind of important information is overlapping relationships, which describe the order of faces that overlap each other and furthermore locate at the same plane surface. This information is also maintained by the face trees.

When a folding operation is simulated against to the internal model, the corresponding face tree has to be renewed suitably for the new situation of the model. That means not only some terminal nodes (leaves) will be broken up (grow), but also a data structure called face list that holds the overlapping relationships among faces (nodes) has to be updated. The latter updating is usually much more complicated than the former updating. In this section, we describe updating procedures for face lists with respect to basic folding operations and several typical complex folding operations. The basic operations include "mountain folding" and "valley folding", while the typical complex operations include "tucking-in", "covering".

5.1. Updating Procedures for Basic Operations

"Mountain folding" and "valley folding" are simplest and most frequently used operations in origami. Obviously, "mountain (valley) folding" is the reverse of "valley (mountain) folding" in the sense of moving faces in opposite directions. Figure 6 illustrates how an origami model changes and how the face trees vary under a "valley folding". From the face tree given in fig.6, it can be easily observed that the initial situation of the face tree possesses only two faces F1 and F2, and moreover F2 is located behind F1. So, the
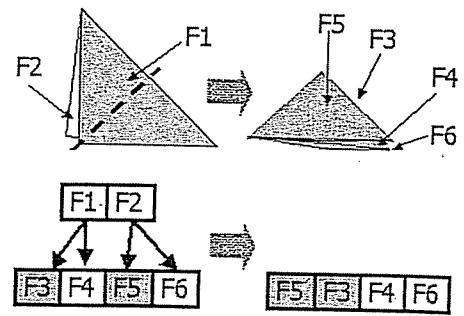
face list corresponding to the upper level of the tree holds the order of overlapping as F1→ F2 (from left to right). After the "valley folding" is performed, the node of F1 split into F3 and F4 because face F1 has been separated into two smaller faces. For the same reason, the node of F2 split into F5 and F6 simultaneously. Additionally, during this folding operation, two faces (F3 and F5) are moved and the others (F4 and F6) are stationary. We indicate the moved faces by hatch.

Now, we have four faces (F3, F4, F5 and F6) overlapping each other. The new face list corresponding to the lower lever of the face tree has to be updated so that it maintains the overlapping relationships among the present faces. We update the face list by the following steps. Firstly, a new sub-list is generated by picking out all moved (hatched) nodes and reversing the order of them. In the example of Fig.6, this sub-list is F5→ F3. Secondly, the remaining nodes in the old face list are just added to the end of the sub-list made at previous step. The updated face list for the example in Fig.6 is F5→ F3→ F4→ F6 as shown at the right of the face tree.

It is not difficult to prove above-mentioned updating procedure to be true even for the case that "valley folding" is applied to several faces at the same time. On the other hand, the updating procedure for "mountain folding", can be simply obtained by modifying above procedure a little, namely, at the second step we do not add the remaining nodes to the end of the sub-list but do the reverse: add the sub-list to the end of the remaining nodes.

5.2. Updating Procedures for Typical Complex Operations

5.2.1. "Tucking-in" and "Covering"
Just like the relation between "mountain" and "valley" folding, "tucking-in" is also the reverse of "covering" at the sense of moving faces in opposite directions. So, here we only introduce the updating procedure for "tucking-in", because that for "covering" can be straightforwardly obtained through analogy.
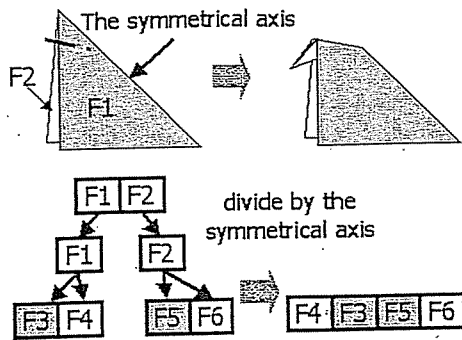
553

Figure 7. Updating a face list in case of "tcking-in".

Figure 8. Updating a face list in case of "expanding".

There exist a symmetrical axis when "tucking-in" or "covering" happens to an origami model. The symmetrical axis is the existing crease to which resulting creases are origami-symmetric [4]. In Fig.7, the crease between F1 and F2 is the symmetrical axis because two new crease generated during operation "tucking-in" will be origami-symmetric to this crease. The updating procedure for a face list is very simple. Fist, we divide the face list in two along the symmetrical axis. The remaining things to do are then simply applying the updating procedures for "mountain folding" and "valley folding" to the separated face lists. For the example shown in Fig.7, the updating procedure related to "mountain folding" is applied to the front face F1 and that related to "valley folding" is applied to the back face F4.

5.2.2. "Expanding" "Expanding" is a quite special operation because this operation not only makes some creases (more than one), but also cancels (opens) some existing creases.

Figure 8 shows an example of "expanding". By applying "expanding" in the way indicated in Fig. 8, two creases are generated on F1 and F2, so that F1 and F2 are split into (F5, F6) and (F7, F8), respectively. Additionally, there is a common crease between F6 and F7, also between F5 and F4 before "expanding", however, they are both canceled after "expanding". So, two faces (F6, F7) and (F5, F4) have to be integrated into a single face, respectively. The updating procedure can be described as follows. As for a moved face (hatch nodes in Fig. 8), if the moved face connects with a stationary one, we integrate the both at the position where the stationary face originally locates. If the moved face connects with another moved one, we integrate these two faces and put them at the beginning of the face list. According to this updating procedure, in the example of Fig. 8 F5 is moved to the position of F4 and integrated with F4, while F6 is integrated with F7 and moved together with F7 to the beginning of the face list.
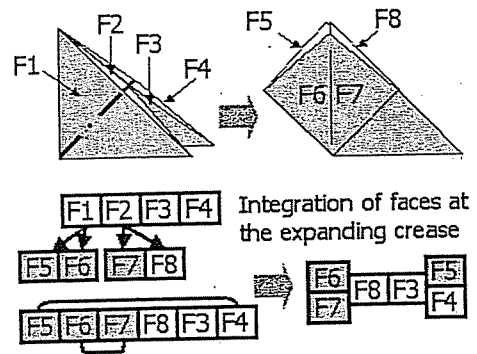
6. Conclusion

This paper presented a system to recognizing and recreating folding process from origami drill books. Proposed revision methods for the positions of folding operations based on origami axioms make the system performance more reliable, and furthermore proposed updating algorithms for basic and typical complex folding operations make it possible to maintain consistency of data (internal models) which describe the transformation of origami models in 3D virtual space during a folding process.

As the future work, it is necessary to carefully determine various threshold values for the revision methods through experiments, and also necessary to make sure that proposed updating algorithms are also applicable to other more complicated folding operations. For this reason, an appropriate way for evaluating the validness of our methods based on experimental results is indispensable.

References

[1] H. Huzita. Understanding Geometry through Origami Axioms. In J. Smith, editor, *Proc. of the First International Conference on Origami in Education and Therapy (COET91)*, pages 37–70. British Origami Society, 1992.

[2] J. Kato, T. Watanabe, H. Hase, and T. Nakayama. Understanding Illustrations of Origami Drill Books. *J. IPS Japan*, 41(6):1857–1873, 2000.

[3] S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaki. INTERACTIVE MANIPULATION OF ORIGAMI IN 3D VIRTUAL SPACE. *J. IPS Japan*, 34(9):1994–2001, 1993.

[4] H. Shimanuki, J. Kato, and T. Watanabe. Constituting Feasible Folding Operations Using Incomplete Crease Information. In *Proc. of IAPR Workshop on Machine Vision Applications*, pages 68–71, 2002.

[5] T. Suzuki, J. Kato, and T. Watanabe. Extraction of Contextual Information Existing among Composite Elements of Origami Books. In *Proc. of 4th IAPR International Workshop on Graphics Recognition*, pages 195–207, 2001.