Estimation of Folding Operations Using Silhouette Model

Yasuhiro Kinoshita * Toyohide Watanabe †

Abstract—**In order to recognize the state of origami, there are only techniques which use special devices or require much time. In this paper, we focus on a folding operation that user applied, from which the state of origami can be understood. We propose the method of estimating the folding operation by using one camera. The folding operation can be estimated by comparing origami shapes before and after a folding operation. We introduce a silhouette model to represent the shape of an origami. Comparing the silhouette model with the shape of origami in a camera image, the folding operation can be estimated. Our method can estimate the folding operation at high speed.**

Keywords: Origami, Camera Image, Feature Points, Silhouette Model, Estimation of Folding Operation

1 Introduction

In an origami work, generally the book called origami drill book has been often used. The drill book indicates the folding operation step by step with corresponding illustration. However, since folding operations are not fully expressed, it is not always easy for beginners to perform an origami work successfully on the way. We propose an interactive folding operation support system which explains the folding operations clearly according to the procedural states of origami.

In order to support folding operations, it is necessary to recognize and maintain the state of origami by the system. Moreover, it is necessary to recognize at high speed in order to construct an interactive system adapting to the operation of user. There are several studies for recognizing the state of the origami sheet. Mitani, et al. proposed the method which uses the sheet with printed 2D bar-code. However, the recognition of the 2D bar-codes was time consuming[1]. Ju, et al. suggested the method which embeds radio tags in paper. However, necessarily use a special device[2]. This method is not adoptable to develop the computer support system of origami which is used by general users. These are unsuitable to support the user's folding operation. Additionally, a data structure for illustrating the state of origami was proposed by Miyazaki, et al.[3]. The data structure can be updated by the folding operation. Therefore, if a folding operation which a user applied to origami is known, it is possible to judge what the state of the origami is. In this paper, we focus on the feature of origami, and propose a method for estimating the folding operation rapidly. Our system uses only one camera to construct the experimental environment.

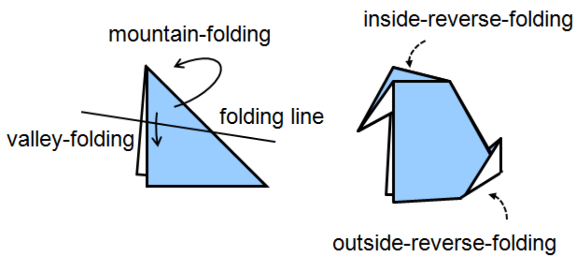

Figure 1: Folding operations

In the following sections, we first discuss our approach, and the outline of our system. Next, we describe the method of estimating folding operations by a series of camera images, and show experiment at results. Finally, we conclude a summary and give our future work.

2 Approach

2.1 Characteristics of Origami

Origami deals with the two-dimensional object, and is bound by geometric constraint. An origami state is changed by each folding operation. In this paper, we call the origami before applying folding operation as $pre-origami$ and the origami after applying folding operation as $post-origami$. The change of origami state depends on the $pre-origami$ state and the folding operation. Therefore, we can understand the $post-origami$ state from the $pre-origami$ state and folding operation.

A folding operation is represented by the location of folding line, folding direction and folding type such as "valley-folding" and "mountain-folding". An ex-

*Department of Systems and Social Informatics, Graduate School of Information Science, Nagoya University, Furo-cho, chikusa-ku, Nagoya, 464-8603, Japan Email: kinoshita@watanabe.ss.is.nagoya-u.ac.jp
†Email: watanabe@watanabe.ss.is.nagoya-u.ac.jp

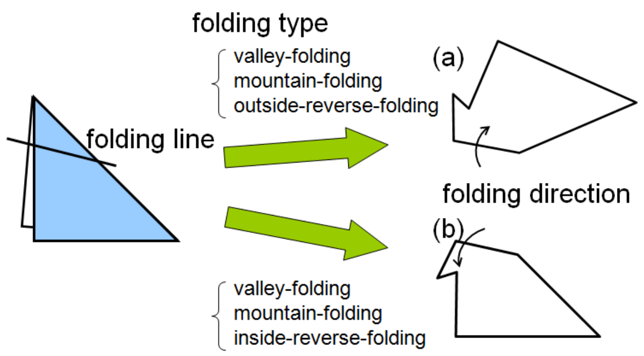

Figure 2: Defferent folding processes

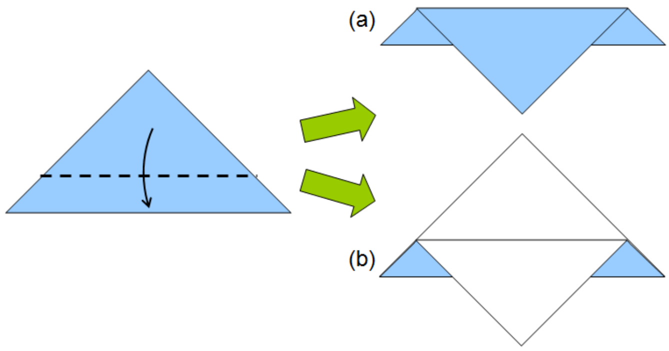

Figure 3: Two kinds of "valley-folding"

ample of folding type is shown in Fig.1. For instance, according to the folding line as illustrated in Fig.2, there are two folding directions and four folding types, "valley-folding", "mountain-folding", "inside-reverse-folding" and "outside-reverse-folding". The origami shapes 2(a) and 2(b) are two kinds of shapes decided by folding direction.

There is a relationship between a folding operation and the change of the origami shape. Therefore, the folding operation can be estimated by comparing the shape of $pre-origami$ and $post-origami$. Since the information about the origami shapes can be obtained from camera images, we estimate the folding operation from camera images. To this end, we introduce a silhouette model. By this model, we achieve to estimate the folding operation at high-speed.

2.2 Silhouette Model

Several researches about recognizing objects in a camera image by maintaining the shape of object have been performed. Kameda, et al. proposed a method to recognize human posture by 3D models which express parts of a human body[4]. In this method, one camera image is matched with a 3D model. However, in order to recognize three-dimensional object, there is a problem that computational cost increases. Therefore, in order to reduce computational cost, we limit folding operations for simplification, an origami must be handled as a two-dimensional object.

The silhouette model describes the shape of an origami with vertexes and edges. Two edges connect with one vertex, and all vertexes are connected by an edge. All edges and vertexes constitute a loop. Moreover, the edges do not cross. A vertex has vertex-ID and coordinates. An edge has edge-ID and two vertex-IDs of both ends. When a folding operation is applied, the shape of silhouette model is changed and becomes the same shape as the $post-origami$. The changed silhouette model obtained from translating the shape of previous silhouette model (called pre-silhouette). The shape of silhouette model in the initial state is maintained in advance because the original shape is square.

Our system can estimate the location of the origami in camera image (called camera-origami) by corresponding with the silhouette model. For corresponding to them, we use the vertexes of the silhouette model and feature points extracted from the camera image. Since there are only small number of vertexes and feature points, comparison frequency is decreased. Therefore, our method can correspond to the silhouette model and the camera-origami with less computational cost than others. From the result of estimation, feature points can be corresponded with the vertexes.

In order to estimate the folding operation, correspondence between the pre-silhouette and the camera-origami after folding operation (call post-camera-origami) is necessary. As a result of correspondence, our system can estimate the folding operation, because the information about the folding operation can be applied to the silhouette model.

When a folding operation is applied and the shape of the origami changes, a new silhouette model (called post-silhouette) which is similar to the current origami shape is needed. The post-silhouette is obtained by applying the folding operation to the pre-silhouette. Therefore, our system is enable to estimate the free folding operation of user.

3 System Outline

3.1 Precondition

Our system uses a fixed camera and makes a series of camera images. The camera is set up to just above an origami. To make the image processing simple, we assume that a background is black and two side of the origami is represented by white and light blue.

Additionally, we restrict the folding operations which user can apply. The folding operation is represented by the location of a folding line, folding direction, and folding type. The folding line is in the origami before folding operation, and becomes the edge constructing the $post-origami$ shape. This is because the folding line has to be the edge of the $post-origami$ shape so that it can be acquired from a camera image after the folding operation. An example is shown in Fig.3. Fig.3(a) is an available folding operation. The folding line is the edge of the $post-origami$ shape. On the other hand, since in the folding line is contained inside the origami shape, in Fig.3(b), it is difficult to detect the folding line from the origami shape. Moreover, we deal with folding types, as "valley-folding", "inside-reverse-folding" and "outside-reverse-folding". Since "mountain-folding" is the same folding operation as "valley-folding" except the folding directions, only "valley-folding" is used.

The end of folding operation is pointed out by a user, because it is difficult to find out the end of folding operation from camera images. In addition, there are folding back and inversion as the user inputs. These operations that cannot be estimated from a silhouette model. However, our system permits those folding operations because they do not need information from the camera image.

3.2 Processing Flow

Our system estimates the location of camera-origami fundamentally. When user indicates that a folding operation is applied, the folding operation is estimated. When drawing up the origami work, it is necessary to estimate two or more folding operations. Therefore, as shown in Fig.4, the shape of silhouette model is changed at each folding operation. The processing flow shown in Fig.4 is summarized as follows.

1. After changing the size of (b) according to (a), both of them are corresponded.
2. After user indicate the end of a folding operation, we compare (c) with (b). Then a folding operation like (d) is estimated.
3. A new silhouette model as (e) is obtained from the estimated the folding operation.
4. After the end of second folding operation, we compare (f) with (e). Then a folding operation like (g) is estimated.
5. A new silhouette model as (h) is obtained.

4 Estimation of Folding Operation

4.1 Estimation of Location of Origami

Our system estimates the location of camera-origami by using the silhouette model. The camera-origami fundamentally assumes fixed form. As a result of the location estimation, the silhouette model position moves to the location where the camera-origami is. However, when a folding operation is applied, the origami shape is different from in the silhouette model shape. In this case, the silhouette model position moves to the location where the origami which is not applied the folding operation exists. Next, we describe the procedure for estimating the location of the camera-origami.

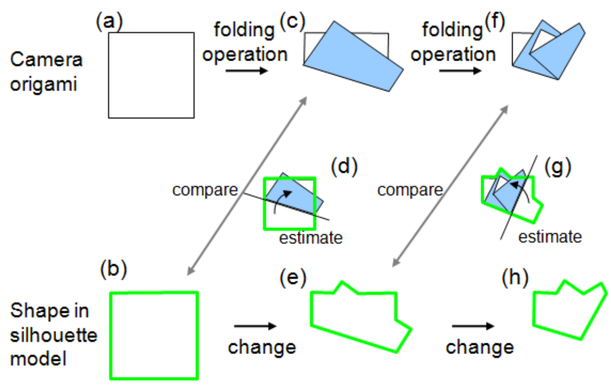

Figure 4: Processing flow

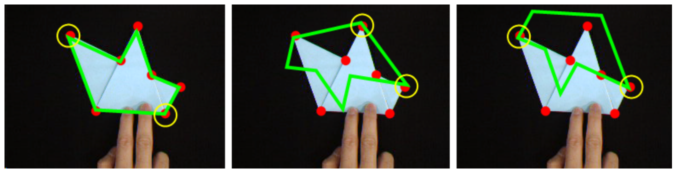

Figure 5: Example of superimposing the silhouette model on the origami in camera image

Firstly, our system extracts feature points from a camera image[5, 6] because feature points are used to corresponde a silhouette model with a camera-origami. Vertexes of the camera-origami can be extracted mostly, although unnecessary points are also extracted from the boundary of a hand and the camera-origami. Therefore, feature points around the area of a hand are removed by skin detection. An extracted feature point also has vertex-ID. If it does not correspond to a vertex of the silhouette model, the vertex-ID becomes -1.

Secondly, our system tracks feature points. When a user folds an origami, it is rarely that he/she moves the origami quickly. For this reason, feature points which represent the same vertex are very near in a series of camera images. Then, as a result of comparing to the previous frame, the nearest point within a predefined threshold is assumed as the same point, our system succeeds to vertex-ID. In this method, it is possible to reduce the comparison frequency between the silhouette model with the camera-origami according to succeeded feature

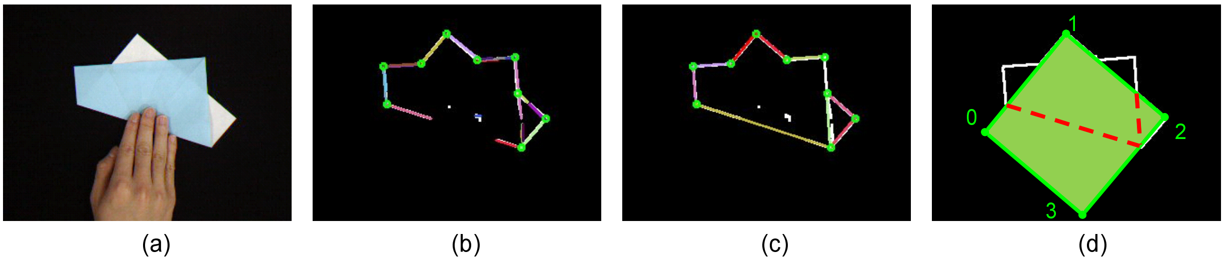

Figure 6: Detection of folding line

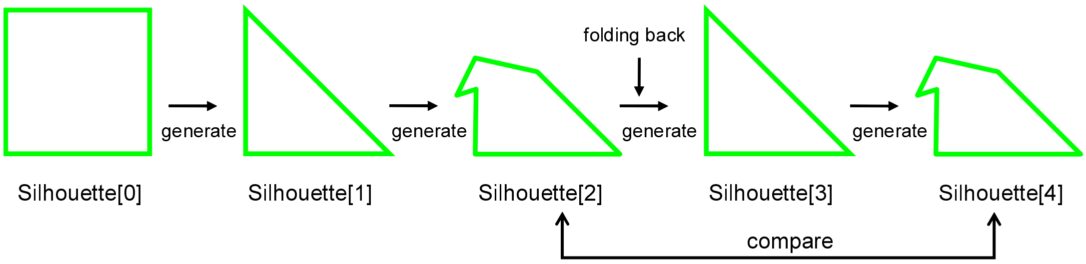

Figure 7: Shape transition in silhouette model

points.

Thirdly, our system carries out matching in order to estimate the location of the camera-origami. After selecting two vertexes and two feature points whose distance are the same as shown in Fig.5, the silhouette model position is moved so that they may overlap. Moreover, our system calculates the matching degree which describes the similarity between the silhouette model and the camera-origami, and estimates that the origami exists in the location where most similar. In matching, comparison frequency decreases by raising the priority of the pattern in which a vertex and a feature point are the same vertex-ID as the result of tracking feature points.

Finally, after the silhouette model position is moved to the estimated location, labeling is carried out. A feature point near each vertex in the silhouette model is assumed to be a vertex of the camera-origami and a vertex-ID is succeeded. In addition, as a result of labeling, feature points can be tracked by the next frame.

When the silhouette model corresponds to the camera-origami correctly, the matching degree becomes the smallest. To obtain such the matching degree, we focus on the gap between the silhouette model and the camera-origami, and the moved distance of the silhouette model position from a previous frame. These are calculated from vertexes of the silhouette model and feature points extracted from a camera image. The gap d_1 between the silhouette model and the camera-origami is an average from each vertex of the silhouette model to a feature point in the neighborhood, and is calculated like in equation (1). C is a set of feature points which are extracted from a camera image, and c shows a feature point included in C. And, n is the number of vertexes in the silhouette model, and v is each vertex. Euclid distance between v_1 and v_2 is expressed in $D(v_1, v_2)$. The gap d_2 between the silhouette model moved the position and the silhouette model with previous frame is an average of the moving distance of each vertex, and is calculated by equation (2). A moved vertex in the current silhouette model is expressed by v, and a vertex of the silhouette model in the previous frame is expressed by v'. Because an origami is not a thing to move quickly, it is thought that the origami exists in the near location from the origami in the previous frame. In addition, a right corresponding often makes smaller the gap than wrong corresponding. Therefore, the matching degree M is defined in (3) as the production of d_1 and d_2.

$$d_1 = \frac{1}{n}\sum_{i=1}^{n} min(D(v_i, c)) \qquad c \in C \tag{1}$$

$$d_2 = \frac{1}{n}\sum_{i=1}^{n} D(v_i, v_i') \tag{2}$$

$$M = d_1 d_2 \tag{3}$$

4.2 Estimation of Location of Folding Line

We explain the method to estimate the location of a folding line in Fig.6. After the shape of an origami is extracted from a camera image as illustrated in Fig.6(a)[7], straight lines are detected by Hough transform[8]. The re-

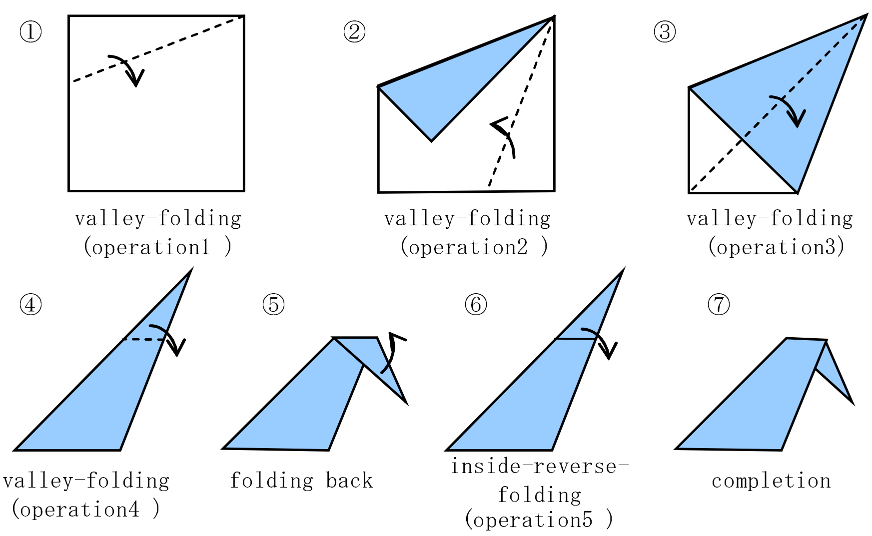

Figure 8: Folding process of "gonbei-garasu"

sult is shown in Fig.6(b). Detected lines are not the edges of camera-origami because they are hidden with hand of user and only the part of edge is detected. Therefore, if there is a segment of the straight line between two feature points, an edge can be detected as shown in Fig.6(c), assuming there is an edge between the feature points. At this time, as shown in Fig.6(c), not only the edge constructing the origami silhouette but also border with the color of two sides of an origami might be extracted. As shown in Fig.6(d), our system distinguishes whether the extracted straight line exists inside the pre-silhouette. Fig.6(d) paints out the inside of the pre-silhouette, and edges which satisfy the condition of folding line are displayed in dashed lines. In this case, from the reason as above, there is possibility that two or more folding lines are detected as shown in Fig.6(d), but there is no problem.

4.3 Decision of Folding Operation

It is necessary to find the most suitable one of the detected folding lines, and to decide folding direction and folding type. Therefore, our system compares the camera-origami with the silhouette model shape that is obtained from each folding line. The most suitable shape silhouette model is made into new one, and a folding line is determined.

The new shape silhouette model is obtained from the pre-silhouette and estimated folding operation. Dividing the pre-silhouette model shape with the folding line, and moving the same face that user moved centering on folding line, and integrating shapes two silhouette model. The moved face can be judged from a camera image.

After the folding line is decided, our system estimates folding type. In order to estimate folding type, our system uses the limitation that "ginside-reverse-folding" and "outside-reverse-folding" are applied after folding mark has been left. The operation of leaving folding mark is input by the user. As a result, folding type can be judged from the history of the shape silhouette model in Fig.7. The shape silhouette model obtained to n-th is expressed as $\text{Silhouette}_{[n]}$. When "inside-reverse-folding" or "outside-reverse-folding" is applied, $\text{Silhouette}_{[n]}$ and $\text{Silhouette}_{[n-2]}$ become similar form by the limitation.

5 Evaluation Experiments

In order to show the effectiveness of our proposed method, we evaluated our prototype system. To show the implementability of the folding operation support system, accuracy and computing time of estimation of folding operation are evaluated in the experiment.

5.1 Description of Experiments

The origami work called "gonbei-garasu" is experimental object. The folding operations that can be estimated are applied five times. The folding process of "gonbei-garasu" is shown in Fig.8. Folding operations other than folding back are called *operation*5 from *operation*1. *Operation*4 from *operation*1 apply "valley-folding", and after folding back, "inside-reverse-folding" is applied in *operation*5. We made "gonbei-garasu" 20 times, and examined the operations 100 times in total.

Experiment environment is as follows.

- CPU: AMD Athlon(tm) 64 Processor 3000+ 2GHz
- Main Memory: 1GB

Table 1: Average of computation time and accuracy of estimation

	$operation1$	$operation2$	$operation3$	$operation4$	$operation5$
computation time (ms)	315	274	223	329	368
accuracy	0.75	0.85	0.90	1.00	0.95

- Development Environment: Microsoft VisualC++, OpenCV[9]
- WEB Camera: ELECOM UCAM-E1D30MBK

In order to designate the background as black, we prepared a black board, and operated an origami on the board. And we experimented adjusting a web camera attached up so that the shoot range might not exceed the board.

5.2 Experimental Results

Table 1 shows the average of computation time and accuracy for estimation of each folding operation. Accuracy is calculated by dividing the number of times correctly estimated in all number of times by each operation.

From Table 1, it turns out that the average of computation time for estimating folding operation is about 294ms. We can see that sufficient high-speed processing is realized to perform supporting folding operation from this result. In addition, accuracy of estimation of folding operation is as high as 84.7% on the whole, and our system can start over even if estimation fails. Therefore, this shows our system is practical.

6 Conclusion

In this paper, we proposed the method which can estimate a folding operation with a camera image. As a result of the evaluation experiment, it showed that our method can estimate folding operation at high speed. Also estimation accuracy becomes enough values. Our future work is to introduce the model which express the state of origami, and to construct of the folding operation support system.

References

[1] Mitani, J.: Recognition and modeling of paper folding configuration using 2d bar code. Transactions of Information Processing Society of Japan **48**(8) (2007) 2859–2867

[2] Ju, W., Bonanni, L., Fletcher, R., Hurwitz, R., Judd, T., Post, R., Reynolds, M., Yoon, J.: Origami desk: Integrating technological innovationand human-centric design. Proceedings of the 4th Conference on Designing Interactive Systems (2002) 399–405

[3] 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

[4] Kameda, Y., Minoh, M., Ikeda, K.: Three dimensional motion estimation of a human body using a difference image sequence. Proceedings of the 2nd Asian Conference on Computer Vision '95 (1995) 181–185

[5] Shi, J., Tomasi, C.: Good features to track. Proceedeings of the 94th Computer Vision and Pattern Recognition (1994) 593–600

[6] Harris, C., Stephens, M.: A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference (1988) 147–151

[7] Canny, J.: A computational approach to edge detection. IEEE Transaction on Pattern Analysis and Machine Intelligence **8**(6) (1986) 679–698

[8] Ballard, D.H.: Generalizing the hough transform to detect arbitrary shapes. Pattern Recognition **13**(2) (1981) 111–122

[9] Intel: Intel open source computer vision library. http://www.intel.com/research/mrl/research/opencv/