PAPER

Anomaly Detection of Folding Operations for Origami Instruction with Single Camera

Hiroshi SHIMANUKI††, Nonmember, Toyohide WATANABE†, Fellow, Koichi ASAKURA††, Senior Member, Hideki SATO††, Member, and Taketoshi USHIAMA††, Senior Member

SUMMARY When people learn a handicraft with instructional contents such as books, videos, and web pages, many of them often give up halfway because the contents do not always assure how to make it. This study aims to provide origami learners, especially beginners, with feedbacks on their folding operations. An approach for recognizing the state of the learner by using a single top-view camera, and pointing out the mistakes made during the origami folding operation is proposed. First, an instruction model that stores easy-to-follow folding operations is defined. Second, a method for recognizing the state of the learner’s origami paper sheet is proposed. Third, a method for detecting mistakes made by the learner by means of anomaly detection using a one-class support vector machine (one-class SVM) classifier (using the folding progress and the difference between the learner’s origami shape and the correct shape) is proposed. Because noises exist in the camera images due to shadows and occlusions caused by the learner’s hands, the shapes of the origami sheet are not always extracted accurately. To train the one-class SVM classifier with high accuracy, a data cleansing method that automatically sifts out video frames with noises is proposed. Moreover, using the statistics of features extracted from the frames in a sliding window makes it possible to reduce the influence by the noises. The proposed method was experimentally demonstrated to be sufficiently accurate and robust against noises, and its false alarm rate (false positive rate) can be reduced to zero. Requiring only a single camera and common origami paper, the proposed method makes it possible to monitor mistakes made by origami learners and support their self-learning.

key words: handicraft education, origami instruction, anomaly detection, one-class SVM, augmented reality (AR)

1. Introduction

Learning techniques for a handicraft such as paper-craft and knitting are generally difficult. The best way to learn the technique is to watch an instructor demonstrate operations step by step. The instructor can immediately give feedback to the learner, interrupt if the learner performs an operation incorrectly, and thereby help the learners perform each step correctly. These days, through media such as books and the Internet for self-learning, the learner can find step-by-step illustrations of a certain handicraft process and watch the operations in instructional videos. Since such media provide no feedback, the learner might perform an operation incorrectly without noticing, and their learning pace will decrease.

Origami, the art of paper folding, is one example of a handicraft. The definition and expression of the origami process—which is a sequence of transformations of a sheet of paper by folding the paper into a complete origami model—has been systematized and standardized by creators and researchers [1], [2]. For example, the Yoshizawa-Randlett system [1] is a diagramming system used to describe the folds of origami models. Still in general use today, it is accepted as the default throughout the international origami community. An origami learner usually folds a sheet of paper while watching an instructional diagram or video showing the origami process based on the diagramming system. An example of an instructional diagram is shown in Fig. 1. Even if beginners follow such a diagram, however, they do not always easily understand the meanings of the instructions sufficiently and fold the paper correctly by themselves. In particular, the beginners tend to continue working despite making mistakes, which accumulate and cause the final folded origami model to have the wrong shape.

Folding mistakes in origami are caused by having multiple candidates for a certain fold that is superficially represented in instructional diagrams or recorded in a video from a fixed viewpoint. The candidate folds are roughly divided into three categories as shown in Fig. 2. As shown by these examples, the silhouette (“outer shape” hereafter) of a correctly folded paper often differs from the incorrect outer shape. Therefore, we studied the possibility of detecting folding mistakes by using the outer shape only. Folding mistakes that hardly change the outer shape are thus excluded from this study, for example, in case of folding only one layer from layered sheets like Fig. 2(b), and in case of resulting in the same silhouette by different operations, and so on.

As for the proposal in this study, which aims to detect folding mistakes, the actions of an origami learner folding paper are monitored by using only a single top-view camera, which can be prepared by an ordinary learner easily. The silhouette of the origami paper is recognized frame by frame from the images captured by the camera. An incorrect operation is detected by comparing the shape of the silhouette with an instruction model that represents origami processes constituted from only correct folds (i.e., without incorrect ones). However, the features extracted from the camera images are insufficient to recognize the learner’s state be-
cause the learner’s hands cause shadows and occlusions and the shape of the origami paper is constantly changed during folding. If a correct fold by the learner is judged as a wrong one as a result of such noises, the learner will become confused. Therefore, the false alarm rate (“false positive rate”) should be as low as possible.

In this study, a method for detecting folding mistakes on the basis of a one-class support vector machine (one-class SVM), which is one type of the unsupervised anomaly detection technique, was developed. Two kinds of features extracted from camera images are devised to be classified by the SVM. One is the change ratio of the origami paper, which expresses progress of folds. The other is folding mistake ratio, which is defined on the basis of the degree of difference from the correct shape. Moreover, the SVM classifier is improved by data cleansing that automatically selects accurate features only.

The proposed method is used to interactively point out folding mistakes of origami learners, especially beginners, and to encourage them to continue trying without giving up in the manner of self-education.

The rest of this paper is organized as follows. Section 2 discusses related works. In Sect. 3, an instruction model is defined. In Sect. 4, methods recognizing the state of the learner are described. In Sect. 5, a method for detecting the learner’s mistakes is explained. In Sect. 6, the result of experiments are presented and analyzed. In Sect. 7, conclusions drawn from this study are presented.

2. Related Work

Augmented reality (AR) techniques are useful for intuitively supporting works and tasks. It has been demonstrated that a folding process can be understood more easily by watching overlaid videos representing the process of transforming a sheet of paper than by reading instructional diagrams [3]. However, when a learner performs operations incorrectly, they might not notice the error only with prerecorded videos and animations. To the best of our knowledge, providing appropriate origami instructions according to the learner’s state has not been researched.

Recognition of the shape (state) of origami paper has been studied to a certain extent. A system that can recognize a folding process from an instructional diagram in an origami book and represent the process with CG animations was proposed by the authors [4]. Because the illustrations in the diagram are sometimes expressed with sufficient distortions for the reader of the book to understand, it is difficult to recognize the shape of the origami paper correctly. However, the learner’s hands frequently cause occlusions on camera images, which make recognition of the origami paper harder. An interactive environment for origami learning in which the state of origami is understood with electronic tagged paper was developed by Ju et al. [5]. However, it is difficult for general learners to prepare the tag-reading hardware. An approach—based on the SURF descriptors—for identifying the states of origami paper in camera images was proposed by Gayathri et al. [6]; however, that approach did not consider occlusions by hands or other objects.

Several methods for recognizing characteristic patterns...
printed on a sheet of paper have been proposed. One method recognizes a folded sheet of paper on which 2D bar codes are printed [7]. Although this method uses only digital photographic images, in which stepwise states of the origami paper are presented, it takes long time to recognize how the sheet is folded. To monitor the continuously changing shape of origami paper when it is being folded, it is necessary to reduce that processing time. Another method recognizes a deformable sheet such as a piece of cloth or paper [8]; however, characteristic patterns or figures must be printed on the sheet first. In addition, a system that tracks textured paper on basis of SURF descriptors was developed by Zhu et al. [9]. And a method for recognizing a paper map and projecting virtual contents onto a foldable map was proposed by Martedi et al. [10].

In this study, we use only solid-color sheets of origami paper which are commercially available for anyone around the world to try origami. A system developed by Kinoshita et al. [11] recognizes the shapes of commercial origami paper in single top-view camera images and displays how to fold the paper onto the images interactively. Based on the recognition method used by their system, a method for monitoring actions of an origami learner is proposed. Moreover, detecting folding mistakes statistically is attempted by using another reported method [12]; however, it was found that the method is sensitive to noise, which drastically decreases its detection rate.

Several previous works focused on 3D modeling to manipulate a virtual sheet of paper. In one work, a system deforms a virtual paper by picking and moving a corner vertex of a sheet by clicking a mouse [13]. In another work, an interface enables more complex ways of folding a sheet by simulating the sheet with a spring-mass model [14]. In another work, 3D modeling was used to design a virtual paper sheet on a tablet [15]. By using the interface, a virtual sheet of origami paper can be constructed interactively and intuitively; however, the implemented folding operations were limited by using devices such as a mouse and a tablet. In our work, an interface for designing and editing the instructional contents is needed to reduce the time and effort required for constructing the folding processes.

3. Origami Instruction

The framework for supporting an origami learner and the data structure for origami instructions are described in the following. As shown in Fig. 3, the framework uses two internal models: an instruction model and a silhouette model. The instruction model records the folding processes necessary to complete the origami work. It is manually constructed in advance on the basis of the instructional diagrams in drill books. The silhouette model represents the outer shapes of the instruction model, and it is used for comparing a sheet of paper in a camera image with the state of the instruction model. The origami support procedure consists of the following four steps. First, the silhouette of the learner’s origami paper sheet is extracted. Second, the position and direction of the learner’s paper are estimated by comparing with the state of the silhouette model. Third, the state of the learner’s paper is specified, and the performed fold is analytically validated. Fourth, instructional contents are overlaid onto the learner’s paper so that the learner can fold the sheet accurately by referring to the overlaid contents [16].

3.1 Instruction Model

To constitute a sequence of origami folds, the data structure defined by Miyazaki et al. [13] is used. The structure describes history information of origami models as geometric objects (such as vertices, edges, and faces) and records the overlap order of coplanar faces as “face lists.” An example of face lists is shown in Fig. 4. In this example, faces $F_2$ and $F_3$, which are coplanar, are overlapped and $F_2$ is located over $F_3$.

Based on the data structure, all instruction contents needed to complete the origami model are constituted as an instruction model. The instruction model for the folding process described in Fig. 1 is shown in Fig. 5. Each node represents each state of the origami paper, and the arrows indicate the state transition by each folding operation (OP).

In the instructional diagram, due to space limitation, several operations are often described as one illustration. Although origami experts can work the operation out even if
some steps are skipped, for beginners one illustration should be depicted finely and briefly. Therefore, the model composer (instructor) should normalize the compound operation including several steps into one operation in advance. The available operations are defined in Sect. 3.1.1, and the normalization method is explained in Sect. 3.1.2.

3.1.1 Folding Operations

Seven types of folding operation commonly used in instructional diagrams are used in this study: mountain fold, valley fold, inside reverse fold, outside reverse fold, spread fold, turn, and unfold.

The mountain fold and valley fold are the simplest and most frequently used operations in origami. Obviously, as shown in Fig. 6, the mountain fold is the reverse of the valley fold in the sense that their faces move in the opposite directions. Namely, the mountain or valley fold divides faces along a crease and turns one side of the divided face by \( \pi \) or \(-\pi\) (±180 degrees).

As shown in Fig. 7, in a similar manner to the relation between mountain and valley folds, the inside reverse fold is the opposite operation to the outside reverse fold in the sense that the faces move in different directions.

As the most complex operation, the spread fold is a generic term for a squash fold (Fig. 8(a)), a petal fold (Fig. 8(b)), a rabbit ear, and a swivel fold which are commonly used in instructional diagrams. All these spread-fold operations can be mathematically defined in a similar way [17].

The turn operation, which is frequently used in origami, just rotates the whole sheet of paper without folding the paper (Fig. 9(a)). Additionally, the unfold operation cancels (“opens”) the existing creases folded by the previous operations (Fig. 9(b)). There are further operations, such as a “sink fold”, depending on the instructional diagram being followed. However, such operations are not used in this work because they are too complex for origami beginners to perform.

3.1.2 Normalization of Folding Operations

Several equations for converting a compound operation to the operations defined by the instruction model are outlined below in Extended Backus Naur Form (EBNF) notation [18].

- **Mountain fold** \( \triangleleft \) turn, valley fold, turn;

  A mountain fold is an unstable operation for origami beginners because it must be performed by holding the origami paper up so that it is not contacting the work surface while folding it. Using mountain folds is there-
fore avoided if possible. To apply the functional equivalent of the mountain fold while keeping the origami paper on the work surface, it is necessary to perform a valley fold between turn operations.

- **Pre-creasing** ≜ valley fold, unfold;
  
  Pre-creasing is an operation that makes crease lines used in a later operation. It is usually performed before complex operations. The combination of a valley fold and an unfold achieves the pre-creasing operation.

- **Pleat fold** ≜ valley fold, valley fold; 
  
  A pleat fold is equivalent to repeating valley folds according to the number of crease lines required. However, the valley folds must be performed in the correct order. In the example in Fig. 10 (a), a valley fold is performed at crease \( a \) and \( b \). By performing the valley fold at \( b \) after the valley fold at \( a \), the pleat fold is completed without a mountain fold.

- **Roller fold** ≜ \( a, b, c \); 
  
  Here, \( a, b, \) and \( c \) represent different valley folds. A roller fold is equivalent to repeating valley folds in order from one direction. In Fig. 10 (b), three valley folds are performed in order from the right.

- **Gate fold** ≜ \( (a, b) \mid (b, a) \); 
  
  Here, \( a \) and \( b \) represent different valley folds, as shown in Fig. 11 (a). A gate fold is achieved by performing two valley folds independently.

- **Blintz fold** ≜ \( (a, b, c, d) \mid (b, a, c, d) \mid \ldots \mid (d, c, b, a) \); 
  
  Here, \( a, b, c \) and \( d \) represent different valley folds, as shown in Fig. 11 (b). A blintz fold is defined as the combination of four valley folds for all corners.

### 3.2 Silhouette Model

To estimate the state of the learner’s origami paper sufficiently quickly, the shape of the paper and the silhouette of each origami paper (state) in the instruction model are matched. However, if some outer shapes of the sheet in certain states are congruent, the state is underspecified. Therefore, successive states having the same shapes are integrated as shown in Fig. 12. Moreover, the silhouette model provides the turning back operation to allow the learner to confirm the correctness of the fold oneself. Coefficients \( \alpha \) and \( \beta \) are used in the state transition and are normally set to satisfy \( \alpha > \beta \).

A silhouette—consisting of vertices and edges in 2D space—is constituted by a projection transform of one state in the instruction model. The silhouette at state \( m \) is represented by \( \sigma_m = (V_m, E_m) \), where \( V_m \) and \( E_m \) denote sets of vertices and edges, respectively. One vertex is defined by 2D coordinates and connected with two edges only. Two types of vertices exist: one is based on the vertex in the instruction model, and the other is an intersection point of edges in the instruction model. The edges of a silhouette do not intersect, and the graph is expressed as a simple cycle.

### 4. Tracking of Origami Paper

As preprocessing for estimating the position of the learner’s origami paper, segmentation processing is applied to camera images. Each camera image is automatically segmented into three regions: background, the learner’s hands, and the sheet of origami paper. The segmentation procedure consists of the following steps:

1. The color space of the source image \( I_t \) is converted from RGB into CIE \( L^*a^*b^* \);
2. The pixels are clustered by k-means, which is a general clustering method, in the color space;
3. The learner’s hand regions (clusters) \( H_t \in I_t \) are extracted by common skin color thresholding;
4. The origami paper regions \( P_t \in I_t \) are determined by using the prepared color information of the paper.

Examples of the preprocessing are shown in Fig. 13. The extracted regions \( H_t \) and \( P_t \) are constituted as a collection of pixels or a collection of polygons. For simplicity of implementation, the former manner of expression is used.
4.1 Estimating Position of Origami Paper

By matching origami paper region $P_t$ extracted from a camera image and silhouette $s_m$ at the current state in the silhouette model, the position and the direction of the learner’s origami paper are estimated. Therefore, silhouette $s_m = (V_m, E_m)$ is mapped on to the image space by similarity transformation (Helmert transformation). The 2D coordinates of vertices $v \in V_m$ in silhouette $s_m$ are transformed into the coordinate space of the image by using the following equation:

$$
\begin{pmatrix}
    v'^T \\
    1
\end{pmatrix} =
\begin{pmatrix}
    k \cos \theta & -k \sin \theta & a \\
    k \sin \theta & k \cos \theta & b \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    v^T \\
    1
\end{pmatrix},
$$

where $k$ denotes a scale factor, $\theta$ denotes a rotation angle and $(a, b)$ denotes a translation vector. The image region of transformed silhouette $s'_m$ is expressed as $S_m$, and transformation $T: s_m \rightarrow S_m$ is defined as

$$
S_m = T_{k, \theta, a, b}(s_m).
$$

While the learner is doing origami, the shape of the origami paper is changed rapidly, and part of the sheet is always occluded by the learner’s hands. Therefore, to evaluate the distance between the changing paper regions and the silhouette, the overlap degree function $R(I_t, s_m)$ proposed by Kinoshita et al. [11] is utilized. $R(I_t, s_m)$ is defined as

$$
R(I_t, s_m) = \frac{|S_m \cap P_t|}{|S_m \cup P_t|} + \sigma |S_m \cap H_t|,
$$

where $|X|$ is the area or the pixel size of region $X$, and $\sigma$ ($0 \leq \sigma \leq 1$) is the weight value of the noise caused by the hands. The position and direction of silhouette $s_m$ for which $R(I_t, s_m)$ attains the maximum value, $R^*(I_t, s_m)$, are calculated for each camera image:

$$
R^*(I_t, s_m) = \max_{k, \theta, a, b} R(I_t, s_m).
$$

It takes quite a while to compute the best overlap degree $R^*(I_t, s_m)$. Therefore, the position of the origami sheet is estimated in real-time by using a simple and effective matching algorithm [11], [19], [20]. The algorithm can recognize the position of the origami sheet in about one millisecond per video frame. Furthermore, even if no region of the origami sheet could be detected temporarily because of occlusion by the learner’s hands or noise, the algorithm enables to go back to detecting the correct position after that.

4.2 Transition of Learner’s State

The state of the silhouette model corresponding to the learner’s sheet is always grasped. Therefore, the learner’s state at time $t$ is defined as $l_t$, which is transitioned from one state of the silhouette model to the other state depending on the shape of the learner’s paper $P_t$. Initial state $l_0$ generally corresponds to initial state $s_0$ of the silhouette model: $l_0 \leftarrow s_0$. Furthermore, current state $l_t$ is determined on the basis of state $l_{t-1}$ at the previous video frame. If the learner’s state, $l_{t-1}$, corresponds to $s_m$, the shape of paper $P_t$ is compared with each of three successive silhouettes ($s_{m-1}$, $s_m$, $s_{m+1}$), which denote the previous state, current state and next state of the silhouette model, respectively. However, note that $s_{m-1}$ is not used when $s_m$ is the initial state and that $s_{m+1}$ is not used in the case of the final state.

When $l_{t-1} = s_m$, learner’s state $l_t$ is recognized by the following procedure.

1. Three overlap degrees $R(I_t, s_{m-1})$, $R(I_t, s_m)$, and $R(I_t, s_{m+1})$ in a video frame $I_t$ are optimized.

2. State transition probabilities are determined by multiplying the optimized degrees by coefficients $\alpha$ and $\beta$, which are defined in Sect. 3.2, as follows:

$$
p_{m-1} \leftarrow \alpha R^*(I_t, s_{m-1}),
p_m \leftarrow (1 - \alpha - \beta) R^*(I_t, s_m),
p_{m+1} \leftarrow \beta R^*(I_t, s_{m+1}).
$$

3. When the maximum value of the state transition probabilities is defined as $p_{max} = \max(p_{m-1}, p_m, p_{m+1})$, learner’s state $l_t$ at the frame is determined as

$$
l_t \left\{ \begin{array}{ll}
{s_{m-1}} & \text{if } p_{m-1} = p_{max}, \\
{s_m} & \text{if } p_m = p_{max}, \\
{s_{m+1}} & \text{if } p_{m+1} = p_{max}.
\end{array} \right.
$$

An example of estimating the position of a learner’s origami paper being folded several times is shown in Fig. 14.
In this example, the learner’s state transitions from $s_5$ to $s_4$ and then changes to $s_6$ via $s_5$. The chart represents variations of maximum values of overlap degrees for respective silhouettes. In some sections in the chart, here the overlap degrees are low because most of the origami paper is covered by the learner’s hands. If one of three overlap degrees is high enough, the origami paper can be tracked accurately.

5. Mistake Detection

By monitoring the change of the learner’s origami sheet during one folding operation, a mistake by a learner can be detected. Therefore, the difference among regions changed by folding is pre-defined as

$$\Delta P_t = (P_t \cup S_m) - (P_t \cap S_m),$$

$$\Delta S_m = (S_m \cup S_{m+1}) - (S_m \cap S_{m+1}).$$

$\Delta P_t$ represents the amount of change of the paper region from the beginning of the folding operation, and $\Delta S_m$ represents the difference between the current and next silhouettes. Note that shape of the paper $P_t$ and silhouette $S_m$ becomes progressively small and distorted through several times of folding operations and $\Delta P$ becomes sensitive to the noise. Therefore, measures without dependence on the size of the silhouette are required for detecting mistakes.

As one of features used for detecting the learner’s mistake, the “change ratio” of the origami paper $C_t$ expresses the progress of the folding operation and is defined as

$$C_t = |\Delta P_t \cap H_t|.$$  

In this equation, hand region $H_t$ is excluded as unnecessary for mistake detection; however, overlap degree $R$ is computed by using $H_t$ (see Eq. (3)). To reduce false detection due to influence by the hands during the folding operation, feature $C_t$ is calculated by using only the change of the origami sheet appearing on the camera images. Furthermore, “folding mistake ratio” $M_t$ is defined as

$$M_t = |\Delta P_t \cap \Delta S_m \cap H_t|.$$  

Two features, $C_t$ and $M_t$, are observed every video frame, and whether the observed values are normal or not is judged. Therefore, an anomaly detection problem for time series data is solved by using a one-class support vector machine (one-class SVM) [21]. The values of $C_t$ and $M_t$ always have a large number of digits because they represent the areas in the image space. Therefore, the logarithms of the values are used as a feature vector for a one-class SVM:

$$x_t = (c_t, m_t) = (\log C_t, \log M_t) \in \mathcal{X}.$$  

5.1 One-Class SVM

Mistakes made by the learner during the folding operations are detected by using a one-class SVM commonly used for anomaly detection. The one-class classification problem is formulated to find a hyperplane that separates a desired fraction of the training data from the origin of the feature space $\mathcal{X}$. This hyperplane cannot always be found in the original feature space; thus, mapping function $\Phi: \mathcal{X} \to F$ from $\mathcal{X}$ to kernel space $F$ is used. Particularly, the following Gaussian kernel (radial basis function) is always used:

$$k(x, y) = (\Phi(x), \Phi(y)) = e^{-\gamma \|x-y\|^2}. $$

To find the hyperplane, the following separation problem is solved:

$$\text{minimize} \quad \frac{1}{2} \|\omega\|^2 + \frac{1}{l} \sum_i \xi_i - \rho,$$

subject to $$(\omega \cdot \Phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, l,$$

where $\omega$ denotes a vector orthogonal to the hyperplane, and $\nu$ denotes the fraction of training data that are allowed to be rejected. Moreover, $l$ is the total number of training data, $\xi = [\xi_1, \ldots, \xi_l]$ is a vector of nonzero slack variables used to penalize the rejected data, and $\rho$ denotes the margin of the hyperplane from the origin.

The one-class SVM algorithm depends on two free parameters $\nu$ and $\gamma$. The correct choice of these parameters significantly influences the quality of the classifier.

5.2 Data Cleansing

A one-class SVM is an unsupervised learning algorithm that uses normal samples only. To detect folding mistakes, by using videos of correct folding operations, the origami sheet is tracked, and the features in the video frames are extracted. However, the extracted features include noises due to shadows of the learner’s hands and occlusions caused by the hands. The SVM classifier is therefore trained only with features extracted from the frames that do not include such noises.

In preparing training samples, manual selection of accurate features from all video frames requires too much effort. Therefore, by introducing criterion $n$ for the fraction of training data, accurate features are automatically extracted from the frames that fulfill the following condition:

$$R^*(I_t, s_{m+1}) \geq (1 - n) \max_i R^*(I_t, s_{m+1}),$$

for all the states in the silhouette model.

5.3 Anomaly Detection for Time Series

The trained SVM classifier is used to detect an anomaly from a video that may contain mistakes. Since the extracted features include noises, as described above, a false positive occurs frequently in the case of detection at every frame. Therefore, statistics of the time series data in the sliding window are utilized. The median of change ratio $c_t$ and the minimum value of mistake ratio $m_t$ in the window are
adopted as the statistics. Using the sliding window can reduce the influence of noise.

6. Experiments

To evaluate the effectiveness of the proposed methods, we conducted the experiments based on 5-fold cross-validation. First, an instruction model and an origami video dataset were prepared. The proposed anomaly detection method based on a one-class SVM was then verified experimentally.

6.1 Construction of Instruction Model

An instruction model is constituted by an origami instructor, or by a learner (if possible). To reduce the burden on the model composers, a support tool that enables information about folding operations to be input by intuitively manipulating a virtual sheet of origami paper with a mouse has been developed [16]. The interface of the tool is shown in Fig. 15.

On the upper left of the screen, the process tree that maintains the history of folding operations from the initial state to the current state is shown in list form. Clicking each item of the list displays the state of origami paper associated with the item. Moreover, unnecessary states and process can be removed by clicking the mouse.

On the right of the screen, information about a folding operation is shown. To input the information, first, the model composer selects the type of operation, such as “Valley” fold and “Mountain” fold. Next, when the composer drags a vertex of the virtual origami model to the lower-left of the screen, the operational information (which consists of the moved vertex, the coordinates of the vertex, and the generated crease) is updated. However, the shape of the model is not permitted to change if the “Turn” operation is selected. Finally, when the “Apply” button is clicked to confirm the operation, a new state of the process tree is generated. If the operation is mistaken, the “Undo” button is clicked to cancel it. This tool also has functions for saving and loading the constructed folding process as the instruction model in files.

6.2 Dataset

To prepare experimental data, five Japanese college students were tasked with folding sheets of origami paper into three kinds of origami works while videos were captured of their folding operations three times per work. The origami works attempted were simple ones aimed at beginners, and listed in Table 1. Further, the videos were manually divided into scenes of each folding. However, the scenes in which the silhouette does not change such as turn operation are excluded from this experiment. About 98 video files were produced in the following file format:

- Frame size: 320 × 240
- Frame rate: 10 fps

Normal samples were obtained by tracking the origami sheet with the correct states in the silhouette model: current state $s_m$ and next state $s_{m+1}$. On the other hand, anomalous samples were prepared by tracking the sheet with the wrong states: current state $s_m$, and the other state $s'_{m+1}$ that can be transitioned from $s_m$. Accordingly, one normal sample and one anomalous sample are made for each video file. An example of a single dataset is represented in Fig. 16. Each sample is generated by tracking with two adjacent states only, and the likelihood of transitions between the states are set up to be equal, i.e. $\alpha = \beta = 0.5$ which are the coefficients mentioned in Sect. 3.2.

6.3 Evaluation

To validate the proposed method for detecting mistakes made by an origami learner, the following evaluation indices were utilized:

\[
\text{(detection rate)} = \frac{\text{(number of normal samples judged as normal)}}{\text{(number of normal samples)}},
\]

Table 1 Origami works used in the experiment

| Work name | Number of steps | Used type of fold            |
|-----------|-----------------|------------------------------|
| Crow      | 6               | valley, inside reverse, unfold|
| Cicada    | 10              | valley, turn                 |
| Whale     | 7               | valley, outside reverse, unfold|

(a) A step in a normal silhouette model
(b) An anomalous silhouette $s'_6$

Fig. 15 Interface of the support tool for constructing instruction models

Fig. 16 Experimental data generated by tracking the origami sheet with a normal silhouette and an anomalous one
The proposed method was experimentally evaluated by 5 × 5-fold cross-validation, and owing to variation in the free parameters, two experiments were conducted. In one experiment, “E1” hereafter, an anomaly was detected by using the classifiers trained while changing hyperparameters $\nu$ and $\gamma$ of the original one-class SVM. In the other experiment, “E2” hereafter, the proposed criterion $n$, namely, the fraction of training data described in Sect. 5.2, is changed instead of $\nu$. The change ranges of the free parameters are listed in Table 2. Parameter $w$ presents the width of the sliding window for detecting an anomaly from the time series data described in Sect. 5.3. Since the frame rate of the video files is 10 fps, $w = 10$ represents a one-second interval.

LIBSVM [22] was used in the experiments with a one-class SVM.

### Table 2  Change ranges of free parameters in experiments E1 and E2

| Parameters | E1                  | E2                  |
|------------|---------------------|---------------------|
| $\nu$      | [0, 0.5] spaced at 0.01 | $\approx 0$ (fixed) |
| $\gamma$   | (0, 5] spaced at 0.1 |                     |
| $n$        | 1 (fixed)           | (0, 5] spaced at 0.01 |
| $w$        | 1, 2, …, 10         |                     |

(coverage) = 

\[
\frac{\text{number of anomalous samples judged as anomalous}}{\text{number of anomalous samples}}
\]

(F score) = \[
2 \cdot \frac{\text{detection rate} \cdot \text{coverage}}{\text{detection rate} + \text{coverage}}
\]

(false alarm rate) = 1 − (detection rate).

The best mean F score obtained by the experiments and the optimal values of the corresponding parameters are listed in Table 3. It is clear from the table that the mean for E2 is higher than that for E1. It is therefore concluded that a more accurate classifier could be constructed by reducing noise from the raw training data.

### Table 3  Best mean F score in experiments E1 and E2

| Experiments | E1 ($\nu = 0.03$, $\gamma = 0.2$, $w = 2$) | E2 ($n = 0.05$, $\gamma = 1.3$, $w = 5$) |
|-------------|------------------------------------------|------------------------------------------|
| Mean detection rate | 0.755 | 0.818 |
| Mean coverage | 0.751 | 0.792 |
| Mean F score | 0.753 | 0.805 |

6.3.2 Influence of Sliding Window Width

To validate the utility of the sliding window explained in Sect. 5.3, the mean detection rate and the mean coverage were evaluated while the window width $w$ was varied. The other parameters at which the best F score was obtained in experiment E2 were used; $\nu = 0$, $\gamma = 1.3$, and $n = 0.05$. Figure 18 represents the results of the mean detection
rate and the mean coverage during $w = 1, \ldots, 10$. The detection rates were extremely low at $w = 1$. Therefore, the anomaly detection is sensitive to noise if the statistics in the window are not used. The detection rate and F score were high for $3 \leq w \leq 5$ and were gradually reduced for $6 \leq w$.

6.3.3 False Alarm Rate

If a fold by an origami learner is judged to be incorrect despite it being correct, the learner will become confused. Therefore, the false alarm rate must be as low as possible. Maximum values of coverage for each false alarm rate in the results of E2 are shown in Fig. 19. When the false alarm rate is zero, up to 52% of the anomalies were detected. Unfortunately, we consider that this result is too low for an anomaly detector; accordingly, it is necessary to investigate at a practical level instead of using video files. The other maximum coverages at some false alarm rates are represented in Table 4.

### Table 4 False alarm rates and maximum coverages

| False alarm rate | Maximum coverages | $\gamma$ | $n$ | $w$ |
|------------------|-------------------|---------|-----|-----|
| 0.00             | 0.520             | 0.1     | 0.09–0.10 | 9 |
| 0.01             | 0.622             | 0.2     | 0.07–0.08 | 4 |
| 0.05             | 0.663             | 1.1     | 0.11  | 4 |
| 0.10             | 0.714             | 1.6     | 0.09–0.10 | 4 |

Fig. 19 Maximum coverage at each false alarm rate.

7. Conclusion

A framework for instructing learners of origami with a single top-view camera is presented. The proposed method for detecting anomalies by using a one-class SVM can detect mistakes made by an origami learner from the images that contain occlusions by the learner’s hands and other noises. Experimental evaluation of the proposed method demonstrated that the false alarm rate is zero. However, the coverage ratio of anomaly must be improved.

To detect anomalies more accurately, distortion of the sheet of paper must be recognized in 3D space. As for future work, we will investigate the possibility of recognizing both the origami paper and the learner’s hands with multiple cameras or other sensors that learners can easily setup. Furthermore, we must devise a method for automatically recognizing and separating the beginning and the end of each folding operation and consider an effective way to point out the learner’s mistakes.

References

[1] S. Randlett, The Art of Origami; Paper Folding, Traditional and Modern, E. P. Dutton, 1961.
[2] R.J. Lang, “Origami diagramming conventions.” http://langorigami.com/article/origami-diagramming-conventions, 2011.
[3] M. Goto, Y. Uematsu, H. Saito, S. Senda, and A. Ketani, “AR-Based Supporting System by Overlay Display of Instruction Video,” The Journal of the Institute of Image Electronics Engineers of Japan, vol.39, no.5, pp.631–643, 2010 (in Japanese).
[4] H. Shimanuki, T. Watanabe, et al., “A Recognition System for Folding Process of Origami Drill Books,” Proc. 5th IAPR International Workshop on Graphics Recognition, GREC 2003, pp.308–317, 2003.
[5] W. Ju, L. Bonanni, R. Fletcher, R. Hurwitz, T. Judd, R. Post, and J. Yoon, “Origami Desk: Integrating Technological Innovation and Human-centric Design,” Proc. 4th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS ’02), pp.399–405, 2002.
[6] L. Gayathri and P. Kumar, “Origami Foldaway Support for Beginners Using Image Processing,” International Journal of Engineering & Technology, vol.7, no.2, pp.217–221, 2018.
[7] J. Mitani, “Recognition and Modeling of Paper Folding Configuration Using 2D Bar Code,” Information Processing Society of Japan, vol.48, no.8, pp.2859–2867, 2007 (in Japanese).
[8] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, “Surface Deformation Models for Nonrigid 3D Shape Recovery,” IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.8, pp.1481–1487, 2007.
[9] K. Zhu, O. Fernando, A. Cheok, et al., “A SURF-based Natural Feature Tracking System for Origami Recognition,” Proc. 20th International Conference on Artificial Reality and Telexistence (ICAT 2010), pp.153–159, 2010.
[10] S. Martedi, H. Uchiyama, G. Enriquez, H. Saito, T. Miyashita, and T. Hara, “Foldable Augmented Maps,” IEICE Trans. Inf. & Syst., vol.E95-D, no.1, pp.256–266, 2012.
[11] Y. Kinoshita and T. Watanabe, “Estimation of Folding Operation Using Silhouette of Origami,” IEAENG International Journal of Computer Science, vol.37, no.2, pp.177–184, 2010.
[12] H. Shimanuki, T. Watanabe, K. Asakura, and H. Satow, “Detection of Mistaken Foldings Based on Region Change of Origami Paper,” Proc. Intelligent Interactive Multimedia Systems and Services (KES-HMSS-18), Smart Innovation, Systems and Technologies, vol.98, pp.84–92, 2018.
[13] S. Miyazaki, T. Yasuda, S. Yokoi, and J. Toriwaiki, “An Origami Playing Simulator in the Virtual Space,” The Journal of Visualization and Computer Animation, vol.7, no.1, pp.25–42, 1996.
[14] Y. Furuta, H. Kimoto, J. Mitani, and Y. Fukui, “Computer Model and Mouse Interface for Interactive Virtual Origami Operation,” Information Processing Society of Japan, vol.48, no.12, pp.3658–3669, 2007 (in Japanese).
[15] P. Paczkowski, J. Dorsey, H. Rushmeier, and H.K. Kim, “PaperCraft3D: Paper-Based 3D Modeling and Scene Fabrication,” IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.4, pp.1717–1731, 2019.
[16] H. Shimanuki, T. Watanabe, K. Asakura, and H. Satow, “Construction and Analysis of Easily Fold-able Processes for Computer-aided Origami,” Proc. 11th International Conference on Ubiquitous Information Management and Communication, IMCOM ’17, pp.961–968, ACM, 2017.
[17] H. Shimanuki, J. Kato, and T. Watanabe, “Recognition of Folding Process from Origami Drill Books,” Proc. 7th International
[18] N. Wirth, “What Can We Do about the Unnecessary Diversity of Notation for Syntactic Definitions?,” Communications of the ACM, vol.20, no.11, pp.822–823, 1977.

[19] T. Watanabe and Y. Kinoshita, “Folding Support for Beginners Based on State Estimation of Origami,” Proc. TENCON 2012 - 2012 IEEE Region 10 Conference, 2012.

[20] H. Shimanuki, T. Watanabe, K. Asakura, and H. Sato, “Improvement of Sheet Tracking for Computer-aided Origami,” IEICE Trans. Inf. & Syst., vol.J101-D, no.6, pp.958–965, 2018 (in Japanese).

[21] B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” Neural Computation, vol.13, no.7, pp.1443–1471, 2001.

[22] C. Chang and C. Lin, “LIBSVM: A Library for Support Vector Machines,” ACM Transactions on Intelligent Systems and Technology, vol.2, no.3, Article No. 27, 2011.

Hiroshi Shimanuki received an M.E. degree in information engineering from Nagoya University. His research interests include issues related to computational origami.

Toyohide Watanabe received a Dr.Eng. degree from Kyoto University. He was a Professor in the Graduate School of Information Science, Nagoya University. Currently, he is a researcher at the Nagoya Industrial Science Research Institute. His research interests include the support mechanism and environment in the educational/learning field.

Koichi Asakura received a Dr.Eng. degree in information engineering from Nagoya University. He is a Professor in the Department of Information Systems, School of Informatics at Daido University. His research interests include distributed/parallel processing, ad-hoc network, disaster information sharing systems, programming learning support systems, and so on.

Hideki Sato received a B.E. degree in electronic engineering from Nagoya University and a Dr.Eng. degree in computer science and communication engineering from Kyushu University. He had been a Professor in the School of Informatics at Daido University since 2002. He is currently a Specially Appointed Professor there. His research interests include database systems and geographical information retrieval.

Taketoshi Ushiana received a Dr.Eng. degree in information engineering from Nagoya University. He is an Associate Professor in the Faculty of Design, Kyushu University. His research interests include content environment design, recommender systems and intelligent web content.