Hand Shape Estimation Using Image Transition Network

Yasushi Hamada, Nobutaka Shimada and Yoshiaki Shirai
Dept.of Computer-Controlled Mechanical Systems, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail:hamada@cv.mech.eng.osaka-u.ac.jp

Abstract

This paper presents a method of hand posture estimation from silhouette images taken by two cameras. First, we extract the silhouette contour for a pair of images. We construct an eigenspace from images of hands with various postures. For effective matching, we define a shape complexity for each image to see how well the shape feature is represented. For a pair of input images, the total matching error is computed by combining the two matching errors according to the shape complexity. Thus the best-matched image is obtained for a pair of images. For rapid processing, we limit the matching candidate by using the constraint on the shape change. The possible shape transition is represented by a transition network. Because the network is hard to build, we apply offline learning, where nodes and links are automatically created by showing examples of hand shape sequences. We show experiments of building the transition networks and the performance of matching using the network.

1 Introduction

Recently image-based human interfaces and understanding the hand gestural languages have attracted increasing attentions as an alternative to traditional input devices like mouses or keyboards. Such attempts previously proposed are approximately divided into two categories.

The first category is the 3-D model-based approach including the model fitting methods[1] and "Estimation by Synthesis(ES)" methods [2][3] which match possible postures generated from a given 3-D shape model and search for the postures best-matched to the input image. While these methods are effective for recognition of arbitrary hand postures, they often require much computation.

The second category directly matches the image features to those of models. The methods of this category[4][5][6] [7][8] register the image appearances or the image features in the learning sequences, and then the input sequence is classified into one of the registered sequence. For recognition of a limited set of hand postures, only useful models are registered. Moreover, computation is usually less because 3-D shapes are not estimated.

For recognition of hand shapes in a gesture sequence, however, the first category is more effective because it is able to limit the search space by the constraint of the joint angles or by that of the velocity. The second category, on the other hand, has to try to match every models.

This paper proposes a method of matching a given hand posture just like the second category, while limiting the candidates by a transition network built during a learning phase. While the Hidden Markov Model (HMM) approach has to build sequence models for all gesture sequences, this transition network alone represents the transition of all possible gestures.

First, in this paper, a basic matching method is described. We determine the features for a pair of images to estimate the hand posture. We collect various hand images to make the model of the postures. A silhouette is extracted from each image and the feature vector is computed as a sequence of the distances from the center of the silhouette to the contour points. The eigenvectors are determined from all feature vectors.

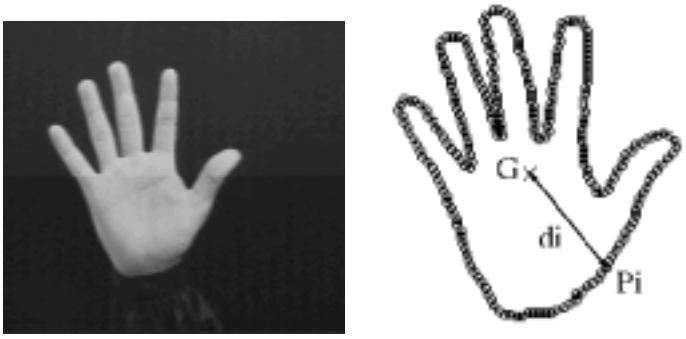
For effective matching, we define the shape complexity for each image to see how well the shape feature is represented. For a pair of input images, the total matching error is computed by combining the two matching errors according to the shape complexity. Thus the best-matched image is obtained for a pair of images.

Next, an effective hand posture matching of a gesture sequence is described. For a given application, we may be able to limit the matching candidate by the constraint on the shape change. That is, the next shape is confined to a set of possible models. The possible transition is represented by a transition network. Because the transition network is hard to build, we apply offline learning, where nodes and links are automatically created by showing examples of hand shape sequences. It is important to merge similar nodes in different image sequences so that the transition obtained in a sequence can be used at the similar node in other sequences.

2 Feature Extraction

2.1 Contour Feature

For simplicity, the hand region is assumed to be brighter than the background and the clothes so that the hand region



(a) A silhouette (b) Extracted contour feature

Figure 1. Feature extraction

(a) Camera layout (b) Captured images

Figure 2. Stereo images

is easily obtained (Figure 1(a)).

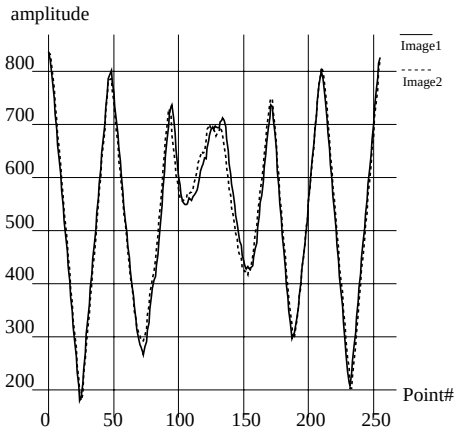
A pair of hand images are obtained by two cameras fixed laterally in front of the user (Figures 2(a) and 2(b)). For each image of the pair, hand region is extracted and then its area S and center of gravity G are computed. Then 256 points $P_i (i = 1 \cdots 256)$ are sampled on the contour of the region so that they are placed at a constant interval (Figure 1(b)). Scale-normalized distance r_i is obtained by

$$r_i = \frac{d_i}{\sqrt{S}} \tag{1}$$

where d_i donotes the distance between G and P_i. This is the shape feature independent of the translation and scale change.

Because the sequence of features depends on rotation, the realignment of the elements is necessary. We select the most significant peak or valley as the start point of r_i. Then realigned $\{r_i\}$, $\boldsymbol{x} = \{r_{a_1}, \cdots, r_{a_{256}}\}^T$, is obtained as the feature vector. Figure 3 shows extracted feature vectors for two similar hand images.

(a) Hand Image 1 (b) Hand Image 2



(c) Extracted Feature Vectors: They are well-matched together.

Figure 3. Feature Vectors of Similar Images

2.2 Building of Eigenspace

In the offline learning phase, possible hand shapes are registered as the model images. For efficient registration, the eigenspace of the feature vector is constructed. The bases of the eigenspace are computed by selecting k principal eigenvectors $\boldsymbol{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_k]$ obtained by Principal Component Analysis. The compressed feature vectors $\boldsymbol{g}_n = \boldsymbol{E}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}})$ $(n = 1, \cdots, M)$ are stored in the database.

In the online shape estimation, scale-normalized distances $\{r_i\}$ are similarly obtained. For normalization of the rotation, we select start point candidates as the significant peaks and valleys. For robust normalization, we select L candidates and evaluate each of them. For the jth candidate $(1 \leq j \leq L)$, feature vector $\boldsymbol{y}_j = \{r_{b_{j1}}, \cdots, r_{b_{j256}}\}^T$ is generated as the jth realigned $\{r_i\}$.

Each $\boldsymbol{y}_j$ is projected into the eigenspace and then the compressed feature vector of the input is computed as

$$\boldsymbol{h}_j = \boldsymbol{E}^T(\boldsymbol{y}_j - \bar{\boldsymbol{x}}).$$

All candidates are matched to the model features to determine the best-matched model.

3 Appearance Matching Using Stereo Images

The basic matching criterion for L feature vectors and the model image n is

$$d_n = \min_{j=1,\cdots,L}(\|\boldsymbol{h}_j - \boldsymbol{g}_n\|) \tag{2}$$

The best-matched model is determined as

$$d = \min_{n=1,\cdots,M}(d_n) \tag{3}$$

Because matching is often ambiguous, we use a pair of stereo images. The matching scheme for stereo images is described in this section.

3.1 Matching based on Shape Complexity

Since two input feature vectors are obtained from stereo images, two matching criteria are computed by equation (3). Note that some hand shapes are difficult to discriminate from a single silhouette.

A problem is how to integrate them to determine the best model.

A simple method is to use the average of the two criteria. This method, however, may be influenced by a bad silhouette (which is not suitable to determine a unique shape).

We conjecture that the more complex are the shapes, the more effectively they represent the 3-D hand shape. The complexity of the shape feature is defined as

$$c = \sum_{i=1}^{256} \frac{|r_{i+k} - r_i|}{k} \tag{4}$$

where k is an experimentally determined constant. $k = 10$ was used in the experiments.

If the complexity of one image is much more than the other, only the former may be used for matching. In general, each of the stereo images is assigned to a weight (w_l, w_r) according to the complexity, and the best model is determined by the weighted average of the two criteria.

Let the complexity of the left and right image be c_l, c_r. The computation of the weight is experimentally determined in the following way

If $\sqrt{c_l^2 + c_r^2} \leq t_1$,
  then $w_l = c_l,\ w_r = c_r$.
If $1/t_2 \leq c_l/c_r \leq t_2$,
  then $w_l = c_l,\ w_r = c_r$.
Otherwise,
  $w_1 = 1, w_2 = 0$ for $c_1 > c_2$.

where t_1 and $t_2 (\geq 1)$ are determined by experiments. In following experiments, $t_1 = 7.84$ and $t_2 = 1.23$ are adopted.

3.2 Result of Experiment

First the eigenspace is built by typical hand images. By experiments, the performance saturates with 12 eigenvectors. Therefore, 12 dimensional eigenspace is used in the following sections.

By a recognition experiment with 260 model images of different hand postures and 74 input images, recognition rate 95.9% was obtained. Examples of input images and the recognition results are shown in Figure 4.

Figure 4. Matching results (from left side, input images(left, right) and matched model images(left and right)

4 Transition Network

For recognition of gestures or a hand sign language, a sequence of hand shapes should be recognized. For a given set of gestures, a limited set of shape changes is allowed.

For efficient matching, possible shape changes are stored in a transition network, where nodes represent typical shapes and links represent possible transitions. Generally such a transition network is difficult to build because it takes much efforts to teach all possible transitions.

This section describes a method to build a transition network by showing a limited number of gesture sequences and effective recognition of a shape sequence using the network.

4.1 Building of Transition Network

The transition network is represented in the eigenspace which is built for recognition of hand shapes described in the previous sections.

For learning the network, sample sequences are taken and the network is incrementally built. For a given sample sequence, a sequence of feature vectors is first created in the eigenspace. Each vector is then matched to model nodes using the criterion d described in section 3.

If d is less than a threshold d_{thres}, it is matched to the model node ($d_{thres} = 1.6$ in this paper). If the matched node is the same as the previous node, the hand shape is regarded as the same as the previous one. In this case, no transition takes place.

If the merged node is different from the previous one, the feature vector is merged to it and the link associated to this feature vector is also attached to the model node. By this operation, a new possible transition is automatically created without actual samples. Figure 5 depicts this case.

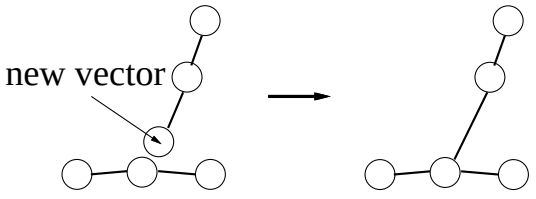


Figure 5. Transition network

If d is greater than d_{thres}, it is regarded as a new shape. Then the new shape becomes a model node.

By repeating this operation for sample sequences, typical hand shapes and possible transitions are represented in the transition network. Note that each node corresponds to the stereo pair of images and the shape feature (in eigenspace).

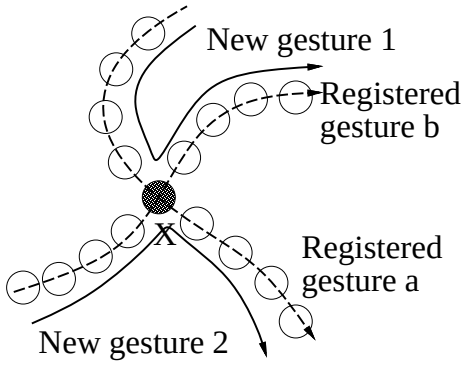


Figure 6. Junction node of transition network

4.2 Shape sequence recognition using Transition Network

In recognition of a shape sequence, the transition network is utilized to find next shape candidates. Given the previous recognition result, the shape candidates are determined as the neighbor of the previous node. Thus the computation cost is much reduced.

Because the junction node (with more than two links) is automatically generated during the learning phase, a new sequence can be tracked. In figure 6, for example, gesture **a** and **b** have been shown and junction X has been generated. If a part of gesture **a** followed by a part of gesture **b** (gesture 1 in the figure) is shown in the recognition phase, it is successfully tracked in the network. Also gesture 2 can be tracked similarly .

5 Experimental Results

We made an experiment of building a transition network and recognizing gesture sequences.

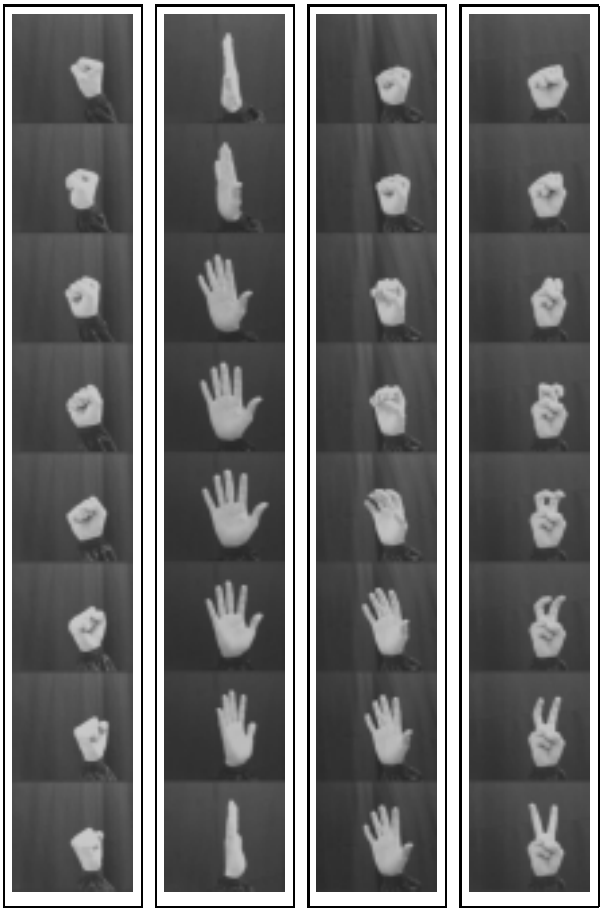
In learning, we prepared 8 kinds of gestures each of which consists of 5 to 35 sample sequences. Each sequence consists of approximately 200 frames. Figure 7 shows 4 kinds of gesture (only 8 typical frames are shown).

In total, 141 sequences and 28200 frames are shown and a transition network with 258 nodes is generated. The number of the node is not very large because many frames correspond to one node.

Figure 8 shows the generated network where the node and the link is represented by a point and a line. Only two principal components in the 12 dimensional eigenspace are shown.

Table 1 shows how many nodes are generated by showing the gestures in the figure.

Next, an experiment with new gesture sequences are performed. Figure 9(a) shows a sequence which consists of multiple kinds of learned gestures. The first part the se-



(a) Gesture A (b) Gesture B (c) Gesture C (d) Gesture D

Figure 7. Gesture sequences used for building transition network (left images only. Top of each column is the start of a sequence and the bottom is the end.)

quence is similar to a part of a learned gesture B. Then the sequence shifts to another gesture which is similar to learned gesture C, and shifts to the third gesture D. The result of recognition is shown in Figure 9(b).

Although this sequence is not learned explicitly, the system traced the transition network successfully. Figure 10 shows the trace of gesture A - D shown in Figure 7 in the transition network (thick gray lines), and the recognition result of new gesture sequence (Figure 9(b)) projected on the same network (thick black lines). We can see that the sequence shifts from gesture B through C to D.

We compare the number of matching trials to estimate the efficiency of the proposed method. In the above example, the number of matching for recognition of an image is 258, which means 51600 matching trials are necessary for recognition of 200 frames.

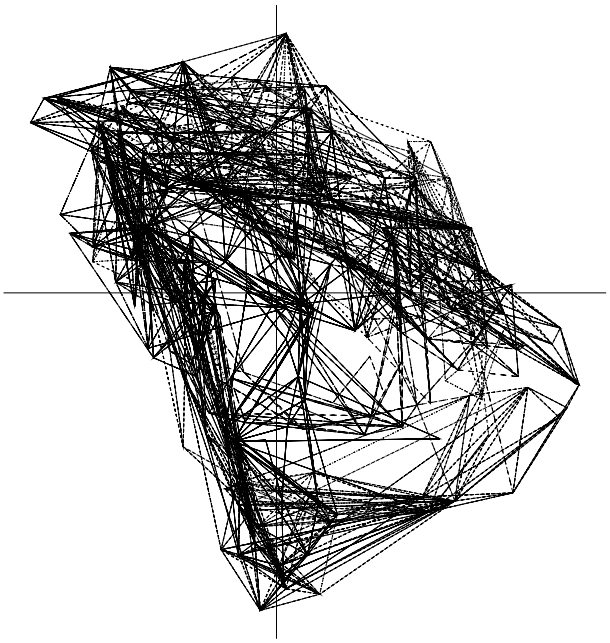


Figure 8. Generated transition network

Table 1. The number of newly generated nodes

Gesture A	3 nodes
Gesture B	49 nodes
Gesture C	16 nodes
Gesture D	2 nodes

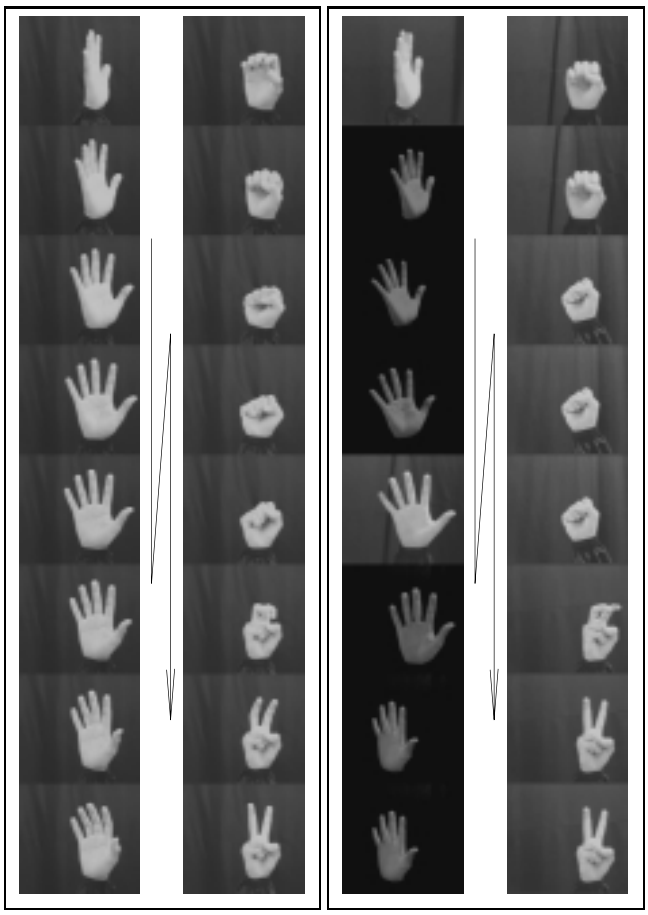
By using the transition network, the proposed method tries only linked nodes. Because the number of linked nodes is not constant, the number of trials depends on the input sequence. For the above sequence, the number of trials is reduced to 5789 which is 11% of the conventional method.

Now that the average number of links for a node is 8.9 in the generated network. This means that for a random sequence, average number of trials is about 3% of the conventional method.

6 Conclusion

This paper presented a method of the hand posture estimation of gesture sequences from silhouette images taken by two cameras. In the offline learning phase, we construct an eigenspace of image features. In the online recognition phase, the complexity of the left and right image are first evaluated, and the best-matched model is determined by integrating the both matching results on the basis of the complexities.

For efficient recognition of gesture sequences, the shape transition network is proposed. In the learning phase, the



(a) Input test sequence (b) Matched model

Figure 9. Posture estimation result by shape tracking with transition network (left images only)

network is automatically generated from the gesture sequences. In the recognition phase, the shape candidates are determined as the neighbor of the previous node in the network. An experiment proved that the computation is reduced to 11% of the conventional method.

The proposed method just recognizes a hand shape. A future work is to recognize a sequence of hand shapes as a meaningful unit such as a word in a sign language.

References

[1] J. M. Rehg and T. Kanade. "Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking". *ECCV'94*, pp. 35–46, 1994.

[2] J. J. Kuch and T. S. Huang. "Virtual Gun: A Vision Based Human Computer Interface Using the Human Hand". In *MVA'94*, pp. 196–199, 1994.

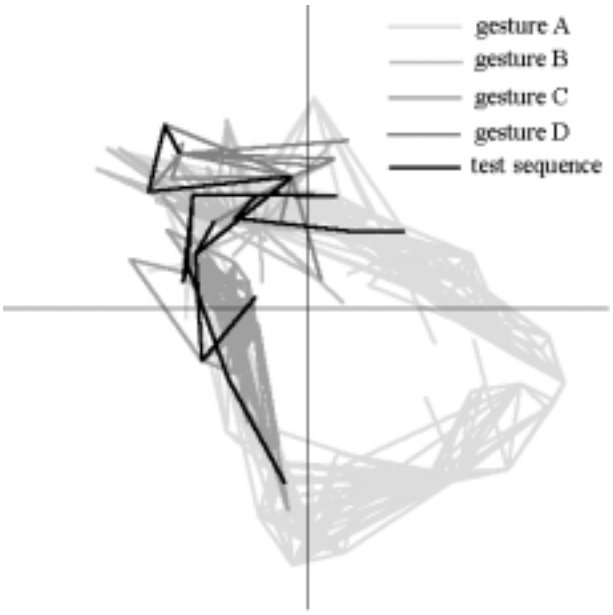


Figure 10. Trace of the test sequence on the transition network

[3] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura. "Hand Gesture Estimation and Model Refinement using Monocular Camera". In *Proc. of 3rd Int. Conf. on Automatic Face and Gesture Recognition*, pp. 268–273, 1998.

[4] Y. Cui and J. Weng. "Learning-based Hand Sign Recognition". *Proc.of Int.Workshop on Automatic Face and Gesture Recognition*, pp. 201–206, 1995.

[5] A. D. Wilson and A. F. Bobick. "Configuration States for the Representation and Recognition of Gesture". *Proc.of Int.Workshop on Automatic Face and Gesture Recognition*, pp. 129–136, 1995.

[6] B. Moghaddam and A. Pentland. "Maximum Likelihood Detection of Faces and Hands". *Proc.of Int.Workshop on Automatic Face and Gesture Recognition*, pp. 122–128, 1995.

[7] T. Nishimura, T. Mukai, and R. Oka. "Spotting Recognition of Human Gestures performed by People from a Single Time-Varying Image". In *Proc. of IROS'97 vol. 2*, pp. 967–972, 1997.

[8] M. J Black and A. D. Jepson. "EigenTracking: Robust Matching and Tracking of Articlated Objects Using a View-Based Representation". *Int.J.of Computer Vision 26(1)*, pp. 63–84, 1998.