Real-time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems

Kenji Oka † Yoichi Sato † Hideki Koike‡
† Institute of Industrial Science
The University of Tokyo
4-6-1 Komaba Meguro-ku
Tokyo 153-8505, Japan
{oka, ysato}@iis.u-tokyo.ac.jp

‡ Graduate School of Information Systems
The University of Electro-Communications
1-5-1 Chofugaoka Chofu
Tokyo 182-8585, Japan
koike@vogue.is.uec.ac.jp

Abstract

In this paper, we propose a fast and robust method for tracking a user's hand and multiple fingertips; we then demonstrate gesture recognition based on measured fingertip trajectories for augmented desk interface systems.

Our tracking method is capable of tracking multiple fingertips in a reliable manner even in a complex background under a dynamically changing lighting condition without any markers. First, based on its geometrical features, the location of each fingertip is located in each input infrared image frame. Then, correspondences of detected fingertips between successive image frames are determined based on a prediction technique.

Our gesture recognition system is particularly advantageous for Human-Computer Interaction (HCI) in that users can achieve interactions based on symbolic gestures at the same time that they perform direct manipulation with their own hands and fingers. The effectiveness of our proposed method has been successfully demonstrated via a number of experiments.

1 Introduction

Several augmented desk interface systems for seamless integration between real objects such as books and associated digital information, have been developed recently. One of the earliest attempts in this domain was presented in Wellner's DigitalDesk[12]. DigitalDesk is equipped with a CCD camera and a video projector, and a user can operate projected applications on a desk by using a fingertip.

Inspired by DigitalDesk, we have developed an augmented desk interface system[4]. The advantage of our augmented desk interface system is that it enables users to perform various kinds of tasks by manipulating both physical objects and electronically displayed objects simultaneously with their own hands and fingers.

(a) (b)

Figure 1. Interaction on our augmented desk

The key component of our augmented desk interface system was a vision-based method for tracking multiple fingertips in real time. Our method made use of an infrared camera, and reliably detected multiple fingertips in real time, without any markers, even in such challenging situations as a dynamically changing lighting condition or a complex background in each input image frame. [1] This was a distinct advantage of our method in comparison with other vision-based methods for tracking hands and fingers. In most of the previously proposed methods, image regions corresponding to human skin are typically extracted either by color segmentation or by background image subtraction[6]. Therefore, it is difficult to use those methods in our augmented desk interface system, where observed color of human skin and image backgrounds continuously changes due to projection by an LCD projector.

Despite the advantage of our proposed method over the other related methods, our method was limited due to lack of the capability of using a combination of direct manipulation and symbolic gestures.

The use of motions of hands and fingers for interactions

[1] Details of how to track users' hands and fingers in our system was reported in [8].

can be generally categorized into two types: direct manipulation and symbolic gestures. The position of a tracked hand or fingertip can be used directly as an input for direct manipulation. For instance, some researchers have used their tracking techniques for drawing or for 3D manipulation of computer graphics (CG) objects[3, 9, 11]. In the case of symbolic gestures, motions of hands and fingers are interpreted based on statistical analysis, e.g., the Hidden Markov Model (HMM)[7] and the Condensation algorithm[1]. HMM has been applied for recognizing the motions of a user's body, hand, and fingers[13, 10, 5].

For natural and intuitive interaction in augmented desk interface systems, it is important for us to be able to use a combination of these two types of interactions, i.e., direct manipulation and symbolic gestures. However, our previous method did not provide a mechanism for using symbolic gestures together with direct manipulation by hands and fingers. In addition, it was not capable of measuring the trajectory of each fingertip since the correspondences of fingertips between successive frames were not considered. This resulted in a limited variety of tasks that we could perform in our augmented desk interface system.

In this paper, we introduce a method for measuring trajectories of multiple fingertips by taking correspondences of fingertips detected in each image frame between successive image frames. First, locations of fingertips in the next frame are predicted based on a filtering technique. Then, the correspondences between the predicted locations and detected fingertips are examined. In this way, trajectories of multiple fingertips can be obtained in real time. In addition, the use of such correspondences contributes to improved performance for detecting fingertips in each image frame.

Moreover, we propose a mechanism for providing a combination of direct manipulation and symbolic gestures based on motions of multiple fingertips. In this work, we make use of the thumb in order to draw a good distinction between manipulative gestures and symbolic gestures; the gestures with the extended thumb are regarded as manipulative gestures, while those with the folded thumb are considered to be symbolic gestures. This is based on the observation that the fingers which users generally use in fine manipulation are only the thumb and the forefinger. After this, the symbolic gestures segmented by that method are recognized based on HMM to be applied interactive systems.

Figure 1 shows some applications using our proposed tracking and gesture recognition. In a drawing tool shown in Figure 1(a), users can make several figures on a desktop with our recognition system of symbolic gestures, and can manipulate those figures directly using their own hands and fingers[2]. Figure 1(b) shows how real objects can be registered and recognized based on users' gestures for creating links between real objects and virtual objects.

The remainder of this paper is organized as follows. In Section 2, we propose a new method for reliably tracking a user's hand and multiple fingertips in uncontrolled environments. Then, we introduce a gesture recognition system using measured trajectories of multiple fingertips in Section 3. Finally, we present our conclusions in Section 4.

2 Real-time tracking of fingertips

2.1 Detecting multiple fingertips in each frame

In this section, we briefly describe detection of multiple fingertips in each input image frame in real time, an achievement which was originally reported in [8].

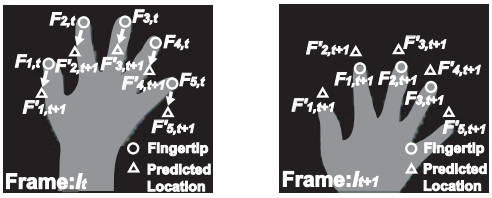
As we described in the previous section, extraction of a user's hand based on color image segmentation or background subtraction often fails when the scene contains complicated backgrounds with changing illumination. To avoid this difficulty, we make use of an infrared camera. In this way, image regions which correspond to human skin can be easily identified by binarization of the input image with a proper threshold value even in complex backgrounds and under different lighting conditions.

Then, for the purpose of fast search of multiple fingertips, a search window of a fixed size is set so that it includes a hand part of the arm region based on the orientation of the arm. The size of the search window should be determined by the approximate distance from the infrared camera to the user's hand. However, we found that a fixed size for the search window works reliably because the distance from the infrared camera to a user's hand on our augmented desk interface system remains relatively constant.

Once a search window has been determined for a hand region, fingertips are searched for within that window. The overall shape of a human finger can be approximated by a cylinder with a hemispherical cap. Thus, we use normalized correlation with a template of a circle with the proper size for detecting fingertips.

In our proposed method, the center of a user's hand is given as the point whose distance to the closest region boundary is the maximum. In this way, the center of the hand becomes insensitive to various changes such as opening and closing of the hand. Such a location for the hand's center is computed by morphological erosion operation of an extracted hand region. Therefore, a morphological erosion operator is applied to the obtained shape of the user's palm until the area of the region becomes small enough. Then, the center of the hand region is given as the center of mass of the resulting region.

Some parameters which are shown above, e.g., binarizing thresholds, the size of a search window and the size of circular template, are different for different persons to some extent. Therefore, those parameters are initialized based on a hand image at the start of our system, and every user can use our system reliably.



(a) (b)

Figure 2. Prediction and Correspondences

Figure 3. Fingertip trajectories

2.2 Measuring fingertip trajectories

In this section, we explain how to obtain trajectories of multiple fingertips detected as described in the previous section by taking into account correspondences of those detected fingertips between successive image frames.

2.2.1 Outline of measuring fingertip trajectories

The overall procedure of our proposed method is summarized as follows. Suppose that n_t fingertips are detected in the tth image frame I_t. The locations of these n_t fingertips are referred to as $F_{i,t}(i = 1, 2, \cdots, n_t)$ as shown in Figure 2(a). First, the locations $F'_{i,t+1}$ of n_t fingertips in the next frame I_{t+1} are predicted. Then, the locations of n_{t+1} fingertips $F_{j,t+1}(j = 1, 2, \cdots, n_{t+1})$ detected in the $t+1$th image frame I_{t+1} are compared with the predicted location $F'_{i,t+1}$ as shown in Figure 2(b). By finding the best combination among these two sets of fingertips, we can determine trajectories of multiple fingertips reliably in real time as shown in Figure 3.

2.2.2 Predicting locations of fingertips

In this section, we explain how to predict the locations of fingertips in one image frame based on their locations detected in the previous image frame by using a Kalman filter. It should be noted that the process described in this section is employed separately for each detected fingertip.

In our implementation, we measure the location and the velocity of each fingertip in each image frame. Hence, we define the state vector $\boldsymbol{x}_t$ as

$$\boldsymbol{x}_t = (x(t), y(t), v_x(t), v_y(t))^T \quad (1)$$

where $x(t), y(t), v_x(t), v_y(t)$ shows the location of fingertip $(x(t), y(t))$ and the velocity of fingertip $(v_x(t), v_y(t))$ in tth image frame. The observation vector $\boldsymbol{y}_t$ is defined to represent the location of the fingertip detected in the tth frame. The state vector $\boldsymbol{x}_t$ and the observation vector $\boldsymbol{y}_t$ are related as the following basic system equation:

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}\boldsymbol{x}_t + \boldsymbol{G}\boldsymbol{w}_t \quad (2)$$

$$\boldsymbol{y}_t = \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{v}_t \quad (3)$$

where $\boldsymbol{F}$ is the state transition matrix, $\boldsymbol{G}$ is the driving matrix, $\boldsymbol{H}$ is the observation matrix, $\boldsymbol{w}_t$ is system noise added to velocity, and $\boldsymbol{v}_t$ is the observation noise, i.e., the error between real location and detected location.

Here we assume approximately uniform straight motion for each fingertip between two successive image frames. Then, $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ are given as follows.

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$\boldsymbol{G} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (5)$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

Please note that (x, y) coordinates of the state vector $\boldsymbol{x}_t$ coincide with those of the observation vector $\boldsymbol{y}_t$ defined with respect to the image coordinate system. This is for simplicity of discussion without loss of generality; the observation matrix $\boldsymbol{H}$ should be in an appropriate form, depending on the transformation between the world coordinate system defined in the workspace, e.g., a desktop of our augmented desk interface system, and the image coordinate system.

Also, we assume that both the system noise $\boldsymbol{w}_t$ and the observation noise $\boldsymbol{v}_t$ are constant Gaussian noise with zero mean. Thus the covariance matrix for $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ becomes $\sigma_w^2 \boldsymbol{I}_{2\times 2}$ and $\sigma_v^2 \boldsymbol{I}_{2\times 2}$ respectively, where $\boldsymbol{I}_{2\times 2}$ represents a 2×2 identity matrix. This is a rather coarse approximation, and those two noise components should be estimated for each image frame based on some clue such as a matching score for normalized correlation for template matching. This part is left for further study of this work.

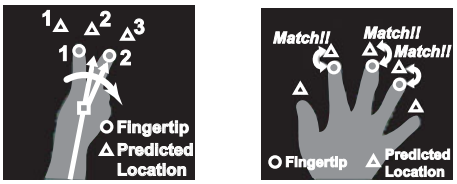
Finally, a Kalman filter is formulated as

$$\boldsymbol{K}_t = \tilde{\boldsymbol{P}}_t \boldsymbol{H}^T \left(\boldsymbol{I}_{2\times 2} + \boldsymbol{H}\tilde{\boldsymbol{P}}_t \boldsymbol{H}^T \right)^{-1} \quad (7)$$

$$\tilde{\boldsymbol{x}}_{t+1} = \boldsymbol{F}\left\{ \tilde{\boldsymbol{x}}_t + \boldsymbol{K}_t \left(\boldsymbol{y}_t - \boldsymbol{H}\tilde{\boldsymbol{x}}_t \right) \right\} \quad (8)$$

$$\tilde{\boldsymbol{P}}_{t+1} = \boldsymbol{F}\left(\tilde{\boldsymbol{P}}_t - \boldsymbol{K}_t \boldsymbol{H} \tilde{\boldsymbol{P}}_t \right) \boldsymbol{F}^T + \frac{\sigma_w^2}{\sigma_v^2} \Lambda \quad (9)$$

where $\tilde{\boldsymbol{x}}_t$ is equal to $\hat{\boldsymbol{x}}_{t|t-1}$ which is the estimated value of $\boldsymbol{x}_t$ from $\boldsymbol{y}_0, \cdots, \boldsymbol{y}_{t-1}$, $\tilde{\boldsymbol{P}}_t$ is equal to $\hat{\boldsymbol{\Sigma}}_{t|t-1}/\sigma_v^2$, $\hat{\boldsymbol{\Sigma}}_{t|t-1}$



(a) (b)

Figure 4. Prediction and matching of fingertips

represents the covariance matrix of estimation error of $\hat{\boldsymbol{x}}_{t|t-1}$, $\boldsymbol{K}_t$ is Kalman gain, and Λ is equal to $\boldsymbol{G}\boldsymbol{G}^T$.

Then the predicted location of the fingertip in the $t+1$th image frame is given as $(x(t+1), y(t+1))$ of $\tilde{\boldsymbol{x}}_{t+1}$. If we need a predicted location after more than one image frame, i.e., after m frames $(m > 1)$, the predicted location can be calculated as follows:

$$\hat{\boldsymbol{x}}_{t+m|t} = \boldsymbol{F}^m \{\tilde{\boldsymbol{x}}_t + \boldsymbol{K}_t (\boldsymbol{y}_t - \boldsymbol{H}\tilde{\boldsymbol{x}}_t)\} \quad (10)$$

$$\hat{\boldsymbol{P}}_{t+m|t} = \boldsymbol{F}^m \left(\tilde{\boldsymbol{P}}_t - \boldsymbol{K}_t \boldsymbol{H} \tilde{\boldsymbol{P}}_t\right) (\boldsymbol{F}^T)^m + \frac{\sigma_w^2}{\sigma_v^2} \sum_{l=0}^{m-1} \boldsymbol{F}^l \Lambda (\boldsymbol{F}^T)^l \quad (11)$$

2.2.3 Correspondences of fingertips between successive frames

For each image frame, fingertips are detected as described in Section 2.1. Then, correspondences between the locations of detected fingertips and the predicted locations of fingertips from Eq.(8) or (10) are examined.

More precisely, the sum of the square of distances between a detected fingertip and a predicted fingertip is computed for all of the possible combinations, and the combination with the least sum is considered to be the most reasonable combination.

If all possible combinations between the detected and the predicted fingertips are calculated, we must consider the maximum of ${}_5P_5$ combinations in the case for 5 detected fingertips and 5 predicted fingertips. To avoid high computational cost for examining all of those possible combinations, we reduce the number of combinations by considering the clockwise (or counter-clockwise) order of fingertips around the center of the hand (Figure 4(a)). In other words, we assume that the order of fingertips in input images do not change because of the crossing of fingers. For instance, in Figure 4(a), we consider only 3 combinations, i) ◯1-△1 & ◯2-△2, ii) ◯1-△1 & ◯2-△3, and iii) ◯1-△2 & ◯2-△3. In this way, we can reduce the maximum number of possible combinations from ${}_5P_5$ to ${}_5C_5$.

Occasionally, one or more fingertips may not be successfully detected in an input image frame. One example of such a situation is illustrated in Figure 4(b) where the thumb and the little finger are not detected due to some type of error. In order to improve the reliability of our method for tracking multiple fingertips, we use the predicted location of a missing fingertip to continue tracking of the fingertip as follows.

If no fingertip is found for a predicted fingertip, we examine the first element of the covariance matrix $\tilde{\boldsymbol{P}}_{t+1}$ in Eq.(9) for the predicted fingertip. This first element represents the ambiguity of the location of the predicted fingertip. Therefore, if the element is smaller than a pre-determined threshold of ambiguity, we consider that the fingertip happens to be undetected for some error for the image frame. Then we use the location of the predicted fingertip as the true location of the fingertip and continue tracking the fingertip. On the other hand, if the first element of the covariance matrix is larger than a pre-determined threshold, then we determine that the prediction of the fingertip is not reliable enough and terminate its tracking. In our current implementation, a threshold for the ambiguity is fixed and chosen experimentally.

If more fingertips are detected than were predicted, we start tracking of a fingertip which does not correspond to any of the predicted fingertips. Then the trajectory of the fingertip is treated as the trajectory of a new fingertip after the ambiguity of the predicted location of the fingertip becomes smaller than a pre-determined threshold.

2.3 Evaluation of the proposed tracking method

We have tested our proposed method for tracking trajectories of multiple fingertips. In particular, we have experimentally evaluated the performance improvement by taking into account correspondences of fingertips between successive image frames.

Seven test subjects participated in this experiment. The hardware configuration of our tracking system consists of a Linux-based PC with Intel Pentium III 500MHz with a Hitachi IP5000 image processing board, and a Nikon LAIRD-S270 infrared camera.

Test subjects were asked to move their hands freely on our augmented desk while the number of extended fingers was kept constant in each trial as shown in Figure 4(a). In the first trial, subjects moved their hands with one extended finger for 30 seconds, and then the subjects changed the number of extended fingers to two, three, four, and finally five. Each trial lasted for 30 seconds and therefore produced approximately 900 image frames. To ensure fair comparison, the output from the infrared camera was first recorded by using a video recorder, and then our proposed method was applied to the recorded video.

We compared the performance of our proposed method with and without correspondences between successive image frames. The result of this comparison is shown in Fig-

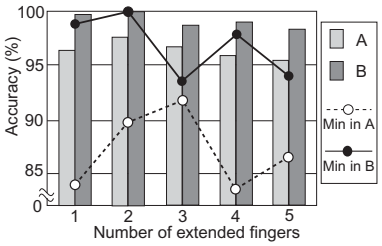


Figure 5. Evaluation of finger tracking

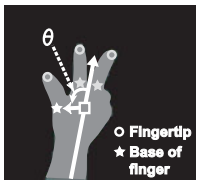


Figure 6. Definition of θ

Method	(a)	(b)	(c)
Average (%)	98.2	99.4	98.3
Standard deviation (%)	3.6	0.8	4.6

Table 1. Evaluation of distinguishing the thumb

ure 5. Method A represents tracking without correspondences, and method B represents tracking with correspondences. Bar charts in this figure show the average of the rate that the number of detected fingertips was correct, and line charts show the lowest rate among seven test subjects.

As Figure 5 shows, the accuracy of detecting fingertips is improved significantly by taking into account correspondences between image frames as proposed in our method. In particular, the accuracy of detection is almost 100% for the case of one or two fingers. In addition, we can see significant improvement in the lowest rate shown with the line chart. This demonstrates the effectiveness of our proposed method for tracking multiple fingertips for real-time applications for HCI.

3 Gesture recognition system based on multiple fingertip trajectories

In this section, we describe the application of our tracking method for gesture recognition for interactions. The advantage of our method is that it enables us to achieve interactions based on symbolic gestures at the same time that we perform direct manipulation with our hands and fingers. For distinguishing symbolic gestures from direct manipulation, our system finds the thumb in the measured trajectories (section 3.1). For recognizing the segmented symbolic gestures, HMM is employed in our system (section 3.2). Our gesture recognition system will be useful for various applications on augmented desk interface systems, especially a drawing tool shown in Figure 1(a).

3.1 Distinction of the thumb

Here we explain the method for distinguishing the thumb from the other detected fingertips. Our method makes use of the angle θ between the direction of a finger, i.e., the direction from the center of a hand to the base of a finger, and the orientation of an arm as shown in Figure 6. We use the base of a finger because it is more stable than the tip of a finger even if the finger moves.

In the initialization stage, we define the standard angle of the thumb θ_T and that of the forefinger θ_F ($\theta_T > \theta_F$). First, by applying the morphological process to a binarized hand image, regions of fingers are extracted. Then, the end of the extracted finger opposite the fingertip is regarded as the base of the finger, and θ is calculated.

Here θ_k is defined as θ in the kth frame from the origin of the trajectory of a finger, and the current frame is the Nth frame from the origin. Then, the score s_T, which represents the likelihood of the thumb, is given as follows:

$$s'_T(k) = \begin{cases} 1.0 & if\ \theta_k > \theta_T \\ \frac{\theta_k - \theta_F}{\theta_T - \theta_F} & if\ \theta_F \leq \theta_k \leq \theta_T \\ 0.0 & if\ \theta_k < \theta_F \end{cases} \quad (12)$$

$$s_T = \frac{\sum_{k=1}^{N} s'_T(k)}{N} \quad (13)$$

In this way, when s_T is above 0.5, the finger is regarded as the thumb.

We have evaluated the performance for distinction of the thumb. The condition of evaluation was the same as in Section 2.3. We have performed three kinds of tests on the assumption of actual work on a desktop, i.e., (a) drawing work with only the thumb, (b) clicking and dragging with the thumb and the forefinger, and (c) drawing work with only the forefinger. The results, depicted in Table 1, show good performance for the distinction of the thumb.

3.2 Recognition of symbolic gestures

Like other recognition techniques[13, 10, 5], our recognition system of symbolic gestures is based on HMM. The input to our recognition system consists of two components. One component is the number of detected fingertips. The other component is a discrete code from 1 to 16 which represents the direction of average motions of tracked fingertips. It is unlikely that we would move each of our fingers independently unless we consciously tried to do so. Thus, we decided to use the direction of the average motions of multiple fingertips instead of the direction of each fingertip. In addition, code 17 is also used as the code which represents approximate stillness.

We have tested our recognition system of symbolic gestures by using 12 different hand gestures which are shown in

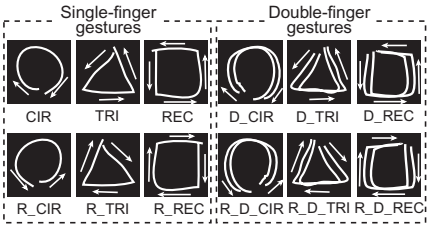


Figure 7. Examples of each gesture

Method	single-finger	double-finger
Average (%)	99.2	97.5
Standard deviation (%)	0.5	1.8

Table 2. Evaluation of gesture recognition

Figure 7. As a training set of data for each gesture, 80 hand gestures made by a single person were used for initializing HMM. Then 6 other individuals participated in this experiment for gesture recognition. For each trial, a test subject made one of the 12 gestures 20 times at arbitrary locations and with arbitrary sizes. The result of this experiment is shown in Table 2 where each value shows the average and the standard deviation of the accuracy for single-finger gestures and double-finger gestures.

As these results show, recognition of single-finger gestures was almost perfect and very reliable. The accuracy for double-finger gestures was also very high, and our gesture recognition system can offer a suitable tool for natural interactions with hand gestures.

4 Conclusion

In this paper, we have proposed a fast and reliable method for tracking a user's hand and fingertips for augmented desk interface systems. In our method, locations of multiple fingertips are predicted based on the Kalman filter; then, correspondences between the predicted locations and detected fingertips are examined for the purpose of obtaining trajectories of multiple fingertips simultaneously. In addition, we introduced hand gesture recognition system based on measured trajectories of multiple fingertips. Our gesture recognition system can recognize both direct manipulation and symbolic gestures at the same time. By detecting the thumb, those two types of gestures are clearly distinguished, and then segmented symbolic gestures are recognized based on HMM. Our tracking and gesture recognition system has performed very well in several experiments.

We are currently extending our method for determining whether each fingertip is touching the surface of a desk. We expect this extension to contribute to even further improvement in the performance of our tracking method. Another future direction of this work is extension for 3D tracking. Currently, our tracking method is limited to 2D motion on a desktop. While this is enough for our augmented desk interface system, interaction based on 3D motion of hands and fingers would be necessary for other types of applications. We are planning to investigate a practical technique for 3D tracking of hands and fingers using multiple cameras.

References

[1] M. Black and A. Jepson, "Recognizing temporal trajectories using the condensation algorithm," *Proc. IEEE FG '98*, pp. 16-21, 1998.

[2] X. Chen, et al., "A drawing system on augmented desk system with two-handed manipulation," *Proc. WISS 2001*, pp. 179-184, 2001 (in Japanese).

[3] J. Crowley, et al., "Finger tracking as an input device for augmented reality," *Proc. IEEE FG '95*, pp. 195-200, 1995.

[4] H. Koike, et al., "Interactive textbook and interactive Venn Diagram: natural and intuitive interface on augmented desk system," *Proc. ACM SIGCHI 2000*, pp. 121-128, 2000.

[5] J. Martin and J. Durand, "Automatic handwriting gestures recognition using hidden Markov models," *Proc. IEEE FG 2000*, pp. 403-409, 2000.

[6] V. Pavlovic, et al., "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Trans. PAMI*, Vol. 19, No. 7, pp. 677-695, 1997.

[7] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, pp. 4-16, 1886.

[8] Y. Sato, et al., "Fast tracking of hands and fingertips in infrared images for augmented desk interface," *Proc. IEEE FG 2000*, pp. 462-467, 2000.

[9] J. Segan and S. Kumar, "Shadow gestures: 3D hand pose estimation using a single camera," *Proc. IEEE CVPR '99*, pp. 479-485, 1999.

[10] T. Starner and A. Pentland, "Visual recognition of American sign language using hidden Markov models," *Proc. IEEE FG '95*, pp. 189-194, 1995.

[11] A. Utsumi and J. Ohya, "Multiple-hand-gesture tracking using multiple cameras," *Proc. IEEE CVPR '99*, pp. 473-478, 1999.

[12] P. Wellner, "Interacting with paper on the Digital Desk," *Communications of the ACM*, Vol. 36, No. 7, pp. 87-96, 1993.

[13] J. Yamato, et al., "Recognizing human action in time-sequential images using hidden Markov model," *Proc. IEEE CVPR '92*, pp. 379-385, 1992.