Three-Dimensional Hand Pointing Recognition using Two Cameras by Interpolation and Integration of Classification Scores

Dai Fujita and Takashi Komuro

Graduate School of Science and Engineering, Saitama University, Japan

Abstract. In this paper, we propose a novel method of hand recognition for remote mid-air pointing operation. In the proposed method, classification scores are calculated in a sliding window for hand postures with different pointing directions. Detection of a pointing hand and estimation of the pointing direction is performed by interpolating the classification scores. Moreover, we introduce two cameras and improve the recognition accuracy by integrating the classification scores obtained from two camera images. In the experiment, the recognition rate was 73% at around 1 FPPI when ±10° error was allowed. Though this result was still insufficient for practical applications, we confirmed that integration of two camera information greatly improved the recognition performance.

Keywords: Hand detection, hand pose estimation, multi-class classification

1 Introduction

Along with the progress of hardware performance and information processing technology, user interface (UI) applications using hand gestures recognized from camera images are being commercialized. Hand gesture UIs enable intuitive interaction with a computer.

However, existing hand gesture UIs have a problem that only limited kinds of input operations are possible compared to other conventional input interfaces. In addition, users have to move their hand largely and the users often get tired during operation. One of the reasons is that existing hand gesture systems cannot recognize subtle finger movement. Hand pointing is a gesture that uses subtle finger movement and that allows users to perform various types of input operations.

There have been studies that have investigated the effectiveness of the mid-air pointing operation [1, 2]. These studies use a hand-worn device to recognize precise hand pointing. On the other hand, a hand gesture control device named Leap Motion that enables precise mid-air pointing operation without wearing a device has been commercialized [3]. Leap Motion has been paid attention as the next generation UIs, but it recognizes hands with a small sensor put on a desk, and the operation space is limited within about $50\,\mathrm{cm}^3$ above the sensor. Therefore, remote operation of a computer from a distance cannot be realized.

Some systems that realize remote hand pointing operation by putting several cameras on the environment have been developed [4–6], but the operation space is limited within a narrow area.

To realize mid-air remote pointing operation of a computer over a wide area, we have to put front-facing cameras for hand recognition on the display. However, it is difficult to recognize a hand that is pointing in the direction of the camera. A pointing hand towards the camera easily changes its appearance and lacks visual salient features in the images. Though much research which recognizes a hand by using color and/or shape information has been conducted [7, 5], such methods either lack robustness or require heuristics which use much prior knowledge to construct a recognition system. Meanwhile, research of recognizing a hand robustly based on machine learning using edge feature of images [8–11], and estimating direction of a target object by using multi-class classification [12, 13] has been conducted.

The objective of this study is to realize robust hand pointing recognition in the three-dimensional (3-D) space with sufficient precision to enable remote pointing operation in a general indoor environment. To achieve this objective, we propose a method to detect a pointing hand and to estimate its direction using multi-class classification based on machine learning from images captured by two cameras.

In the proposed method, classification scores are calculated in a sliding window for hand postures with different pointing directions. Detection of a pointing hand and estimation of the pointing direction are performed by interpolating the classification scores. Then, the interpolated classification scores obtained from two camera images are integrated. By using two cameras, our method not only can obtain 3-D hand positions but also can improve its accuracy by utilizing the difference of hand appearances between two camera images. Since hand pointing has largely different appearances according to viewpoints and difference of camera positions reduces the effect of complex background, integration of two camera information can be more effective than just using doubled size of information.

2 Classification of Hands with Different Pointing Directions

The basic principle of the proposed method is to divide the angle space of hand pointing into some classes, and to recognize a pointing hand and its direction by interpolating classification scores for different classes. Interpolation of classification scores enables hand pointing direction recognition with higher resolution than just classification into one class. In the proposed method, pointing hands in an image are detected and their yaw and pitch angles are estimated. Figure 1 shows the overview of the proposed method.

The proposed method consists of two phases: classification and integration. In this section, we describe classification of hands with different pointing directions and interpolation of classification scores. The integration phase will be described in Section 3.

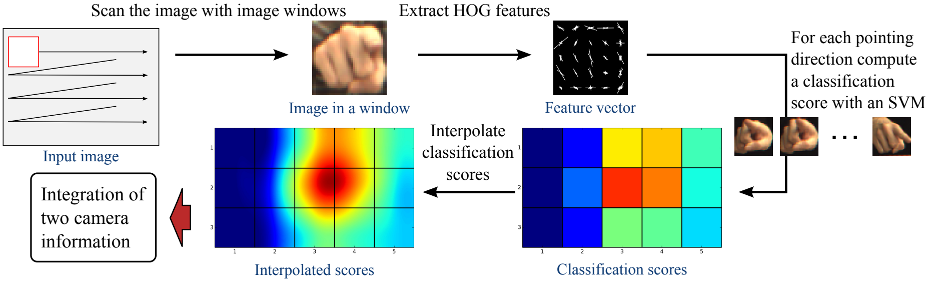

Scan the image with image windows
Extract HOG features
Input image
Image in a window
Feature vector
For each pointing direction compute a classification score with an SVM
Interpolate classification scores
Integration of two camera information
Interpolated scores
Classification scores

Fig. 1: Flow of the proposed method

2.1 Overview of Classification Score Computation

Classification score computation consists of the following four steps.

1. Scanning an image with sliding windows and correcting image distortion
2. Feature extraction
3. Computation of classification scores
4. Interpolation of classification scores and estimation of hand pointing direction

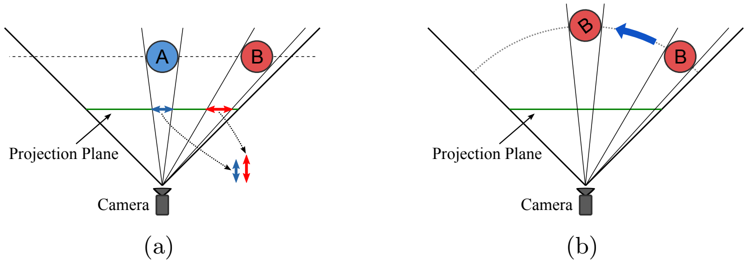
First, an input image is scanned with sliding image windows $W(x, y, s)$. The subsequent procedure will be performed per image in a window. Image distortion caused by perspective projection is corrected. Second, an input vector $\boldsymbol{x}$ is obtained by extracting Histograms of Oriented Gradient (HOG) features [14] from the image in the window. Third, a classification score $S(i, j)$ is computed for each pointing direction by using a classifier trained in advance. Fourth, classification scores are interpolated to improve the estimation resolution. Finally, if the maximum interpolated score is equal to or greater than a threshold λ, a pointing hand is detected and its direction (θ, ϕ) is estimated. We describe details of these steps in the rest of this section.

2.2 Scanning an Image with Sliding Windows and Correction of Image Distortion

To detect pointing hands over the input image, the whole input image is scanned by shifting a sliding window by a regular distance. To detect pointing hands regardless of individual variation of the size of a hand and distance to the hand, N different sizes of windows are used in scanning. The sizes of larger windows are determined by multiplying the size of the smallest window $N-1$ times by a scale ratio a.

Since the shape of a window can vary due to image distortion on the projection plane, our method first determines the center position (x, y) and the scale s of the window while scanning. After the position and scale are obtained, image distortion is corrected around the position, and the exact shape of the window is determined.

Figure 2 illustrates an overview of how image distortion is corrected. Assuming the shape of a pointing hand to be a sphere, it looks larger when projected on a side of the image plane than when projected on the center of the plane as shown in Figure 2(a). Therefore, the distortion is corrected by projecting and rotating the window to the image center as shown in the Figure 2(b).



(a) (b)

Fig. 2: Image distortion correction

2.3 Feature Extraction

A feature vector $\boldsymbol{x}$ is extracted from the image in the window $W(x, y, s)$. Here, HOG is used as a feature.

To equalize feature dimension, an image in a window is resized to a fixed size $w_{\mathrm{p}} \times h_{\mathrm{p}}$ pixels. Then a feature vector $\boldsymbol{x}$ is computed from the resized image.

2.4 Computation of Classification Scores

When a feature vector $\boldsymbol{x}$ in the window is obtained, classification of whether the vector belongs to each class with a different pointing direction is performed, and a classification score is computed for each direction.

First, we define classes for different hand pointing directions. The frontal direction is used as the origin in the angle space and the angle space is divided into $w_{\mathrm{c}} \times h_{\mathrm{c}}$ equal-sized rectangular regions. Columns of the classes are numbered $1, 2, \ldots, w_{\mathrm{c}}$ from left to right. Likewise, rows of the classes are numbered $1, 2, \ldots, h_{\mathrm{c}}$ from top to bottom. A class at i-th row and j-th column is denoted by $C_{i,j}$.

Let θ_{size} and ϕ_{size} be vertical and horizontal angle ranges included in a class. A class $C_{i,j}$ includes pointing hands with directions (θ, ϕ) satisfying the following constraints

$$\left(j - \frac{w_{\mathrm{c}}}{2} - 1\right) \theta_{\mathrm{size}} \leq \theta < \left(j - \frac{w_{\mathrm{c}}}{2}\right) \theta_{\mathrm{size}}, \tag{1}$$

$$\left(i - \frac{h_{\mathrm{c}}}{2} - 1\right) \phi_{\mathrm{size}} \leq \phi < \left(i - \frac{h_{\mathrm{c}}}{2}\right) \phi_{\mathrm{size}}, \tag{2}$$

where $\theta = 0°$ and $\phi = 0°$ mean a finger is pointing exactly to the center of a camera. θ increases when the hand rotates to the right, and ϕ increases when the hand rotates to the bottom.

We mention the parameters used in the experiment described later. In the integration phase, the angle difference caused by binocular disparity narrows an overlapping area of two cameras' fields of view. Therefore, $w_\mathrm{c} > h_\mathrm{c}$ is desirable to widen the view range. Moreover, $w_\mathrm{c} h_\mathrm{c}$ should be kept as small as possible to reduce the computation cost. From now on, we use $w_\mathrm{c} = 5$, $h_\mathrm{c} = 3$ in the example figures. Figure 3 shows an example of a class definition.

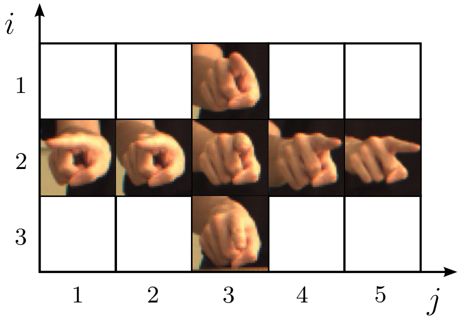


Fig. 3: An example of class definition. Example images are shown in several classes

Using the above-mentioned class definition, $w_\mathrm{c} h_\mathrm{c}$ classifiers, each of which classifies a feature vector into a class $C_{i,j}$ and the non-hand object class, are trained in advance. Linear support vector machines (SVMs) are used for training. The feature vector is classified by all those classifiers, and classification scores for all the classes are computed.

A classification score $S(i,j)$ for a class $C_{i,j}$ is defined as

$$S(i,j) = \frac{\boldsymbol{w}_{i,j}^\top \boldsymbol{x} + b_{i,j}}{||\boldsymbol{w}_{i,j}||}, \tag{3}$$

where $\boldsymbol{w}_{i,j}$ and $b_{i,j}$ are a weight vector and a bias parameter for $C_{i,j}$ respectively. $\boldsymbol{w}_{i,j}$ and $b_{i,j}$ are selected to satisfy $S(i,j) > 0$ if the feature vector $\boldsymbol{x}$ is in $C_{i,j}$. Thus the larger $S(i,j)$, the more the classifier is confident that $\boldsymbol{x}$ is of a pointing hand with the corresponding direction.

2.5 Interpolation of Classification Scores

If we classify the feature vector into the class with a maximum score, the resolution of hand pointing direction is limited to coarse and discrete values. Therefore, we improve the resolution by interpolating the classification scores.

Assuming the classification scores change smoothly, we use bicubic interpolation. Let k ($k > 1$) be a scale factor for interpolation, and $P(i', j')$ be an interpolated classification score using bicubic interpolation ($i' = 1, \ldots, kh_{\mathrm{c}}, j' = 1, \ldots, kw_{\mathrm{c}}$). A maximum score $\hat{S}$ of interpolated classification scores and its position $(\hat{i}, \hat{j})$ are obtained as

$$\hat{S} = \max \left\{ P(i', j'), i' = 1, \ldots, kh_{\mathrm{c}}, j' = 1, \ldots, kw_{\mathrm{c}} \right\}, \tag{4}$$

$$\hat{i} = \frac{\arg\max_{i'} \left\{ P(i', j'), i' = 1, \ldots, kh_{\mathrm{c}}, j' = 1, \ldots, kw_{\mathrm{c}} \right\}}{k}, \tag{5}$$

$$\hat{j} = \frac{\arg\max_{j'} \left\{ P(i', j'), i' = 1, \ldots, kh_{\mathrm{c}}, j' = 1, \ldots, kw_{\mathrm{c}} \right\}}{k}. \tag{6}$$

A pointing hand is detected in the window if the maximum interpolated score $\hat{S}$ is equal to or greater than a threshold λ. Otherwise the window is considered as a non-hand region. When the window is a hand pointing region, its horizontal and vertical directions (θ, ϕ) are estimated by

$$\theta = \left(\hat{j} - \frac{w_{\mathrm{c}}}{2} \right) \theta_{\mathrm{size}}, \tag{7}$$

$$\phi = \left(\hat{i} - \frac{h_{\mathrm{c}}}{2} \right) \phi_{\mathrm{size}}. \tag{8}$$

Figure 4 shows a typical example of classification score computation and interpolation.

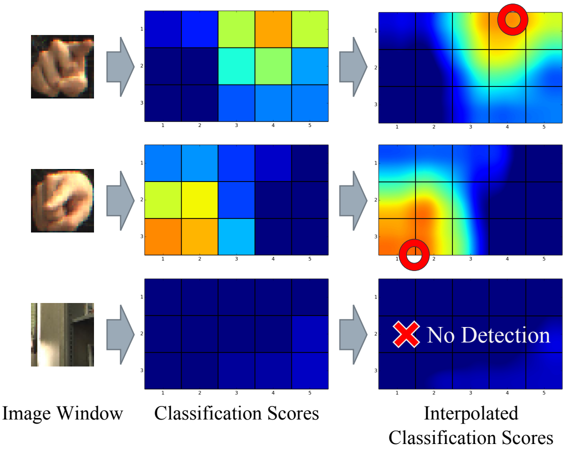


Fig. 4: A typical example of classification score computation and interpolation. Hand pointing direction is estimated using the position of the peak score drawn with a red circle

3 Three-Dimensional Hand Pointing Recognition using Two Cameras

In the previous section, we described the algorithm for a single camera. In this section, we describe a method to improve the performance by using two cameras and integrating the classification scores.

It is obvious that the use of two cameras improves the accuracy because it doubles available sensor information. However, the accuracy may be further improved by using two cameras, for example, (i) when a hand is overlapping with a face and (ii) when a finger is pointing at around a camera.

When a hand is overlapping with a skin-colored region, gradients of a hand are weakened and unreliable as cues. In this situation hand recognition becomes difficult. However, even when a hand is overlapping with a skin-colored region from a camera, a non-overlapping hand can sometimes be seen from the other.

When a hand is pointing toward a camera, recognition is difficult because contours of an index finger (which has the most discriminative feature in hand pointing images) cannot be obtained. When using two camera information, such a problem can be alleviated by utilizing the angle difference from two cameras.

3.1 Layout of Two Cameras

Two cameras are horizontally placed as shown in Figure 5. Let b [mm] be a baseline length. b is set to be large enough to utilize the difference of hand appearances, and at the same time small enough to keep a wide field of view. Operation space of a user is where a hand can be captured with a sufficient size by both the cameras.

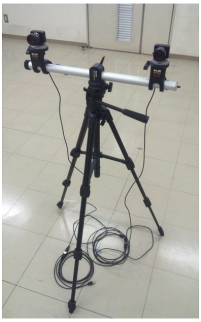


Fig. 5: Layout of two cameras

3.2 Classification and Interpolation For Each Camera

Before integration of two camera information, classification scores and their interpolation for both left and right camera images are calculated in advance.

The same algorithm described in Section 2 is applied to the two camera images independently.

3.3 Window Pair Selection Based on the Epipolar Constraint

To integrate two camera information, pairs of windows of left and right camera images which satisfy the epipolar constraint are selected. In this section, we describe a method to select pairs of windows considering the discreteness of window scale, image distortion correction and individual variation of the size of a hand. The procedure to select pairs of windows to be integrated is as follows:

1. Scan a left camera image with windows of all scales.
2. For each position and scale of the window for the left camera image, scan a right camera image with windows of appropriate scales
3. Select a pair of windows for left and right camera images

First, a left camera image is scanned with windows of all scales. Let $(x_\mathrm{L}, y_\mathrm{L})$ and s_L be the center position and scale of the window respectively. The scale s_L is a ratio of the size of the window to that of the smallest window.

Second, scales of windows for the right image are determined. Multiple scales for the right image are selected from $\{a^{-1}s_\mathrm{L}, s_\mathrm{L}, as_\mathrm{L}\}$, where a is the scale ratio of window sizes, based on the difference between the distances from the left and right cameras to the position of the target point (the distances are transformed to depths in image distortion correction).

Third, the right camera image is scanned with windows of the selected scales. Let s_R be the current scale. The right camera image is scanned with windows which have the Y center equal to y_L. The scan range is calculated as follows. Let z_{s_R} [mm] be the optimal depth from a camera to a hand for the scale s_R. The depth for the smallest window z_1 is determined from the standard sizes of hands, and $z_{s_\mathrm{R}} = z_1/s_\mathrm{R}$ holds under the pinhole camera model. Here the scan range on the right image $[x_\mathrm{begin}$ [pixel], x_end [pixel]$)$ is written as

$$x_\mathrm{begin} = x_\mathrm{L} - \left\lfloor \frac{abf}{z_{s_\mathrm{R}}} \right\rfloor, \tag{9}$$

$$x_\mathrm{end} = x_\mathrm{L} - \left\lfloor \frac{bf}{az_{s_\mathrm{R}}} \right\rfloor, \tag{10}$$

where f [pixel] is the focal length and the origins of the images are at the center of the images.

Finally, a pair of windows for left and right camera images is selected. Let $(x_\mathrm{R}, y_\mathrm{R})$ be the center position of the window for the right image.

3.4 Integration of Two Camera Information

Integration of classification scores is performed on the selected pair of windows for the left and right images $W(x_\mathrm{L}, y_\mathrm{L}, s_\mathrm{L})$, $W(x_\mathrm{R}, y_\mathrm{R}, s_\mathrm{R})$. As a result of integration, the maximum classification score $\hat{S}$ and estimated direction (θ, ϕ) are obtained.

Let $P_{\mathrm{L}}(i', j')$ and $P_{\mathrm{R}}(i', j')$ be the interpolated classification scores calculated in the left and right windows respectively. Based on the centers of the windows, yaw angle difference $\varDelta\theta$ between the windows is written as

$$\varDelta\theta = \tan^{-1} \frac{(x_{\mathrm{R}} - x_{\mathrm{L}})f}{x_{\mathrm{L}} x_{\mathrm{R}} + f^2}. \tag{11}$$

The sum of the interpolated scores is calculated considering the angle difference as

$$P(i', j') = \begin{cases} P_{\mathrm{L}}(i', j' + \varDelta j) + P_{\mathrm{R}}(i', j') & (j' \leq k w_{\mathrm{c}} - 2\varDelta j) \\ 0 & (\text{otherwise}) \end{cases}, \tag{12}$$

where k is a scale factor used in the interpolation, $i' = 1, 2, \ldots, k h_{\mathrm{c}}$, $j' = 1, 2, \ldots, k w_{\mathrm{c}}$ and

$$\varDelta j = \mathrm{round}\left(\frac{k \varDelta\theta}{\theta_{\mathrm{size}}} \right). \tag{13}$$

The maximum classification score $\hat{S}$ and estimated direction (θ, ϕ) are calculated from Equation (4), (5) and (6) in the same manner as that for a single camera. If the maximum integrated classification score is equal to or greater than a threshold λ, a pointing hand is detected in the pair of windows. When a pointing hand is detected, its direction is estimated as (θ, ϕ). Also, the 3-D position of the pointing hand is obtained as

$$X = \frac{b(x_{\mathrm{L}} + x_{\mathrm{R}})}{x_{\mathrm{L}} - x_{\mathrm{R}}},\ Y = \frac{b y_{\mathrm{L}}}{x_{\mathrm{L}} - x_{\mathrm{R}}},\ Z = \frac{bf}{(x_{\mathrm{L}} - x_{\mathrm{R}})}. \tag{14}$$

Figure 6 shows an overview of the integration process.

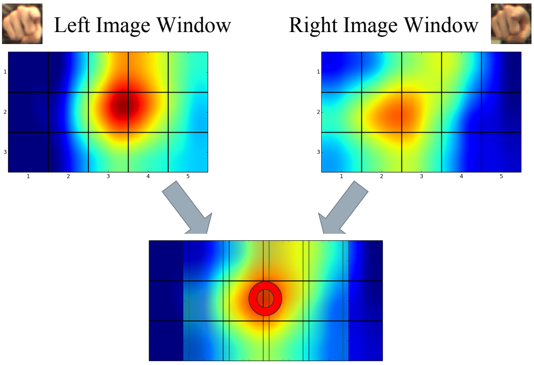


Fig. 6: An overview of integration of two camera information. Hand pointing direction is estimated using the peak classification score

4 Evaluation Experiment

We conducted an experiment to evaluate the performance of the proposed method. We evaluated recognition rates and false positives per image (FPPI) under various conditions for comparison. The experiment was conducted with two types of the numbers of cameras (single camera versus two cameras), two types of numbers of classes $((w_c, h_c) = (5, 3), (6, 4))$ and various peak threshold values λ.

4.1 Experimental Setup

The hand pointing training set consisted of pointing right hand images that were collected from four subjects of 21–22 years old students. Since our method does not support roll angles of hand pointing, subjects were asked to maintain pointing posture with their back of the hand facing upward. The test set consisted of 2,645 stereo images that were collected from the same four subjects as in the training set. The training set consisted of 66,825 positive hand images in total and 1,835 negative non-hand images. The number of the hand images were increased by translating 7,425 source images by one pixel in eight directions. When evaluating the performance for the images of a subject, the hand images of the subject were excluded from the training set (i.e. the training set for a subject did not contain any examples for the subject). An example of hand images in the training set is shown in Figure 7. The numbers of the hand images in the training set varied according to the subjects and the angle range. The number of the source images for each subject and number of classes is summarized in Table 1.

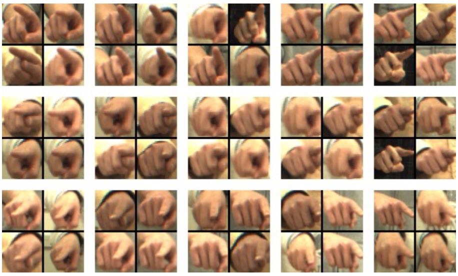


Fig. 7: An example of hand images

The hand images used in the training and test set were captured using a system specially made to obtain ground truth for hand pointing directions. The system consists of a rack and a Kinect sensor which is downwardly installed on top of the rack and can obtain direction of hand pointing.

This experiment was performed under the fluorescent lighting condition as in usual indoor environment. The baseline b was set to 400 mm. The horizontal

Table 1: The numbers of the source images in the training set. The subjects are identified as A, B, C and D

$w_p \times h_p$	A	B	C	D
5×3	4827	4961	5026	4515
6×4	5586	5682	5760	5247

angle of view of cameras was 32.4°. The distance between the cameras and a user was around 2 m.

The size of the test set images were 640×480 pixels. The number of the sizes of sliding windows N was set to five. The scale ratio a was 1.2. Scaling of sliding windows is realized in a relative sense by resizing input images. The shift amount for window scanning was four pixels. Windows in which a pointing hand was detected were aggregated if they were largely overlapping and estimated directions were similar to each other. The image size for HOG feature extraction was 40×40 pixels (which was the same as the absolute fixed size of a sliding window). HOGs were computed with the cell size of 8×8 pixels and block size of 2×2 cells. The size of the feature dimension was 576. The angle range included in a class was $\theta_{\mathrm{size}} = 10°$, $\phi_{\mathrm{size}} = 13°$ when $(w_{\mathrm{c}}, h_{\mathrm{c}}) = (5, 3)$, $\theta_{\mathrm{size}} = 9°$, and $\phi_{\mathrm{size}} = 10°$ when $(w_{\mathrm{c}}, h_{\mathrm{c}}) = (6, 4)$.

We defined the evaluation criteria as follows. Recognition rates were defined as the ratio of a number of successful recognition to a number of hand pointing regions in the test set. Recognition was considered as successful if it satisfied the following conditions: (i) the relative error of the detected position was small and (ii) the error of the estimated direction was within α. The condition (i) can be written as follows:

$$|x - X| \leq \delta \wedge |y - Y| \leq \delta \wedge |x + w - X - W| \leq \delta \wedge |y + h - Y - H| \leq \delta \quad (15)$$

where (x, y), w and h denote the position, width and height of a detected window respectively, (X, Y), W and H denote those of the ground truth and $\delta = 0.3(\min(w, W) + \min(h, H))/2$. α was set to 10° unless otherwise specified. Unsuccessful recognition was considered as false detection.

Recognition was performed with several thresholds λ which were common to the data from all the subjects. Mean values of recognition rates and FPPIs for the data from each subject with the same threshold value were used as the overall recognition rates and FPPIs respectively.

Let λ^* be a threshold whose FPPI was nearest to 1. By using λ^* as a candidate threshold, we measured the recognition rates and FPPIs per subject when $\lambda = \lambda^*$. We also measured the recognition rates with varying angle error tolerances α when $\lambda = \lambda^*$.

4.2 Results and Discussion

Figure 9 shows some examples of recognition results. A rectangle represents the position of a detected pointing hand. An arrow drawn from a center of the rectangle represents the estimated pointing direction. From the results, we can see that the hand pointing was detected and its directions were estimated correctly.

Figure 9(a) shows Receiver Operating Characteristic (ROC) curves. Figure 9(b) shows the results with varying angle error tolerances α and the threshold of λ^*. Table 2 shows the result per subject with the threshold of λ^* and mean FPPIs was around 1.

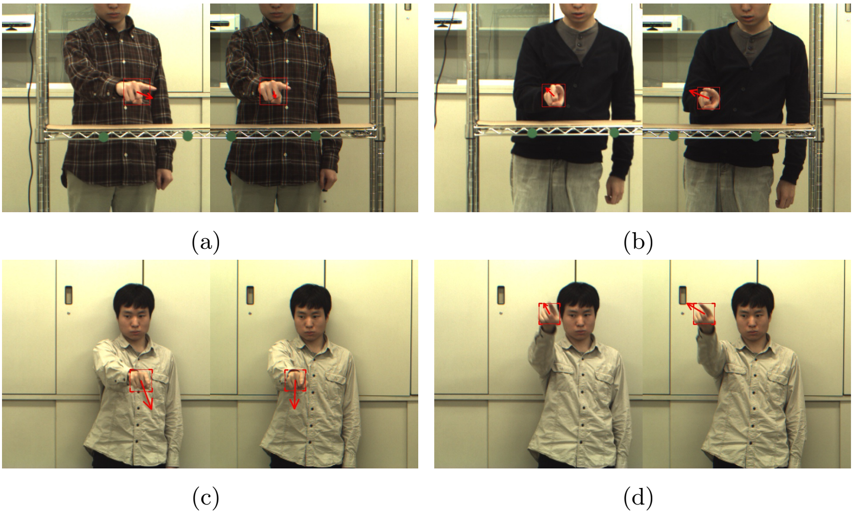
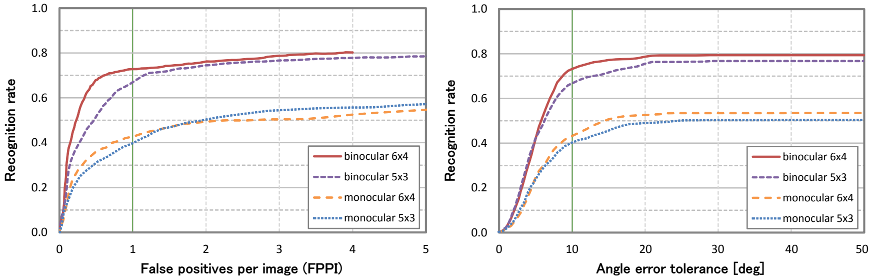

(a) (b) (c) (d)

Fig. 8: Recognition result examples

The larger number of classes ($(w_c, h_c) = (6, 4)$) gave better performance than the smaller one (5×3). Recognition rates were 42.83% when using a single camera and 72.80% when using two cameras, at around FPPI 1 with the number of classes 6×4. The accuracy was greatly improved by integrating two camera information.

Integration of two camera information improved the accuracy, but the recognition rate was still around 70%, which may be insufficient for the use in real applications. The accuracy will be improved by using information of multiple video frames. The detection rate, which is equal to the recognition rate regardless of the angle error tolerance α, was 79.03% when using two cameras and the number of classes 6×4. From Table 2, we can see that there were large gaps between the subjects. The reason may be that the number of subjects was small and thus sufficient variation could not be obtained. For reference, when the training set included the same data as in the test set, the recognition rate was 94.8% at around 1 FPPI when using two cameras and the number of classes 5×3. This result showed that the upper limit performance of the proposed



(a)

(b)

Fig. 9: (a) ROC curves with the angle error tolerance α of 10°. (b) Recognition rates and angle error tolerances α with the threshold of λ^* (a threshold when FPPI was around 1). Numbers in legends denote the number of classes ($w_c \times h_c$)

Table 2: Recognition rates and FPPIs with the threshold of λ^*

# of cameras	# of classes	Subject	Rec. rate [%]	FPPI
Single	5×3	A	27.51	2.00
		B	19.47	0.28
		C	38.12	0.42
		D	74.55	1.33
		Ave.	**39.91**	**1.01**
	6×4	A	36.11	1.72
		B	13.88	0.58
		C	49.39	0.75
		D	71.96	1.00
		Ave.	**42.83**	**1.01**
Two	5×3	A	86.60	1.02
		B	27.43	0.56
		C	62.53	0.68
		D	88.92	1.63
		Ave.	**66.37**	**0.97**
	6×4	A	94.09	1.23
		B	28.47	1.02
		C	84.24	0.75
		D	84.39	0.97
		Ave.	**72.80**	**0.99**

method was much higher and therefore the accuracy may improve by increasing the variation of training data.

The mean execution time for one frame was 206 ms when $(w_c, h_c) = (5, 3)$ and 316 ms when $(w_c, h_c) = (6, 4)$. The computation was executed on a desktop PC with Intel Core-i5 2400 CPU with multiple threads. Though this execution time was not sufficient for real-time applications, it can be improved by using hand tracking as well as the accuracy.

5 Conclusions and Future Works

In this paper, we proposed a method for hand pointing recognition using two cameras. The result of the experiment showed that correct hand detection and direction estimation were realized, and that the accuracy was greatly improved by integrating two camera information.

Future works include accuracy improvement. The accuracy was still not sufficient to realize gesture UI applications using hand pointing operation. To improve it, we will increase the number of the training samples and use information of multiple consecutive frames by hand tracking. Furthermore, performance investigation in various environment is also needed. By training classifiers using a dataset including environmental variation, we should investigate whether or not our method can be used in various environment. In addition, to realize comfortable UI applications, operation space should be enlarged by using, for example, fish-eye lenses. Future works also include creation of a gesture UI application using hand pointing.

References

1. Vogel, D., Balakrishnan, R.: Distant freehand pointing and clicking on very large, high resolution displays. In: UIST, New York, NY, USA (2005) 33–42
2. Nakamura, T., Takahashi, S., Tanaka, J.: One-finger interaction for ubiquitous environment. In: ICIS. (2010) 267–272
3. Leap Motion. https://www.leapmotion.com/
4. Schick, A., van de Camp, F., Ijsselmuiden, J., Stiefelhagen, R.: Extending touch: Towards interaction with large-scale surfaces. In: ITS. (2009) 117–124
5. Sato, Y., Saito, M., Koike, H.: Real-time input of 3d pose and gestures of a user's hand and its applications for hci. In: IEEE VR. (2001) 79–86
6. Hu, K., Canavan, S., Yin, L.: Hand pointing estimation for human computer interaction based on two orthogonal-views. In: ICPR. (2010) 3760–3763
7. Dominguez, S., Keaton, T., Sayed, A.H.: A robust finger tracking method for multimodal wearable computer interfacing. IEEE Trans. on Multimedia **8**(5) (2006) 956–972
8. Yuan, Q., Thangali, A., Ablavsky, V., Sclaroff, S.: Parameter sensitive detectors. In: CVPR. (2007) 1–6
9. Huang, D.Y., Hu, W.C., Chang, S.H.: Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. Expert Systems with Applications **38**(5) (2011) 6031–6042

10. Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: CVPR. Volume 2. (2003) 432–439
11. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. IEEE Trans. on PAMI **28**(9) (2006) 1372–1384
12. Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., Torr, P.H.S.: Randomized trees for human pose detection. In: CVPR. (2008) 1–8
13. Huang, C., Ding, X., Fang, C.: Head pose estimation based on random forests for multiclass classification. In: ICPR. (2010) 934–937
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005) 886–893