19th IEEE International Symposium on Robot and
Human Interactive Communication
Principe di Piemonte - Viareggio, Italy, Sept. 12-15, 2010

# Visual Interface System by Character Handwriting Gestures in the Air

Toshio Asano and Sachio Honda
Hiroshima Institute of Technology, Japan
tasano@cc.it-hiroshima.ac.jp

**Abstract—A visual interface system that recognizes handwriting of Japanese *katakana* characters in the air has been developed. Characters written in a single stroke have both on-strokes and off-strokes. Thus, the shapes of the hand-gesture characters are different from the shapes of characters written on paper. It is difficult to trace the shapes of characters in air because the writer cannot see the trajectories. In this study, a light emission diode (LED) pen and a TV camera are used to capture the LED light trajectory, and the movements of the light are converted into direction codes correcting the slant of the handwriting character. The codes are normalized to 100 data items to eliminate the effect of writing speed. The 100 direction codes are compared with model data in which the direction codes of 46 Japanese characters are defined. Next, the system has expanded to a multi-camera system. Two of four cameras are selected and the 3-D positions of the gesture trajectories are calculated by the stereo method, and the position data are converted into front view data. In the experiments, we attained a recognition rate of 92.9% for the single-camera system. The multi-camera system has the advantage that it can recognize gestures regardless of the origin directions of the gestures. The system also has the ability to recognize the directions of the gesture commands with an accuracy of 9º.**

Key words: Gesture recognition, Hand gesture, Human interface, Image processing, Character recognition

## I. Introduction

Computers are used in a variety of situations, but human interface methods are currently limited to input devices such as keyboards and mice. If computers with TV cameras can understand human gestures, then people will be able to easily interface with computers and robots [1-3].

Gesture and shape recognition is very popular in the field of image recognition [4-6]. Gestures are done by moving, for example, the head, the body, or the hands. Among these, hands can express information most easily. Shaking and rotating are usually used in hand gestures [7,8], but if a system can recognize a word, then many kinds of information can be sent by gestures to computers and robots. Research in which alphabets are used has been reported [9]. Shapes of some alphabets were deformed for recognition. These alphabets are not useful in Japan. Rather, *katakana* or *hiragana* would be more useful for Japanese.

Japanese language uses three kinds of characters: *hiragana*, *katakana* and *kanji*. The *katakana* characters are based on Japanese pronunciation and can express all Japanese words and sentences. There are 46 basic *katakana* characters and some consonants for certain characters.

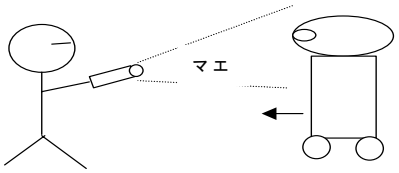


Fig. 1 Character hand gesture recognition system for robot control.

The difficult points of hand gesture recognition of letters are as follows. 1) Gestures used to write in the air using a single stroke consist of on-strokes and off-strokes of the characters. 2) The trajectories are invisible to the writer, so the letter loci are very bad.

The gesture trajectory contains on-strokes and off-strokes. The on-strokes contained in the gesture trajectory are the strokes that would be written on paper, and the off-strokes are the strokes that do not represent writing on paper but rather the return lines of pen movement.

In this paper, two visual interface systems, a single-camera system and a multi-camera system, that enable recognition of characters written in a single stroke using a light emission diode (LED) pen are presented. A computer recognizes the handwritten *katakana* letters using their direction codes. This system can be used as an interface tool for computers and an instruction tool for robots. Fig. 1 shows an example as a robot control command. The character command means “Go forward”.

## II. Detection Method

The developed system uses a TV camera to detect movements of a LED pen. A button on the pen is pressed when writing a character. Fig. 2 shows the positions of the LED light for each frame.

The $x$, $y$ position data of points of light are converted into direction codes representing direction vectors (see Fig. 3). The movement distance $\rho_i$ and angle $\theta_i$ are calculated for each light position by the following equations:

$$\rho_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}\ , \tag{1}$$

and

$$\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right), \tag{2}$$

978-1-4244-7990-0/10/$26.00 ©2010 IEEE

where $(x_i,y_i)$ is the coordinate pair of the $i$-th light position $P_i$, $\rho_i$ is the distance between $P_i(x_i,y_i)$ and $P_{i-1}(x_{i-1},y_{i-1})$ and $\theta_i$ is the angle of the vector from $(x_{i-1},y_{i-1})$ to $(x_i,y_i)$.

The angles are each converted into one of the eight direction codes that are shown in Fig. 4. The angle ranges of the direction codes have different widths. Angle ranges for vertical and horizontal directions are narrow (30°) and those for slant directions are wide (60°). This is because the angles used for slant strokes are more varied than those for vertical and horizontal strokes. Fig. 5 shows a *katakana* and its direction codes. On-strokes (4,2,2,1) and off-strokes (7,5) are mixed.

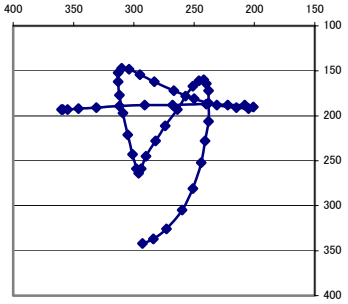


Fig. 2 Detection of LED pen tip

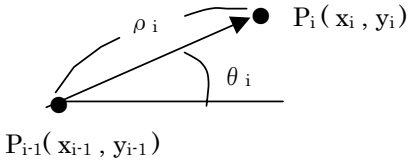


Fig. 3 Direction vector.

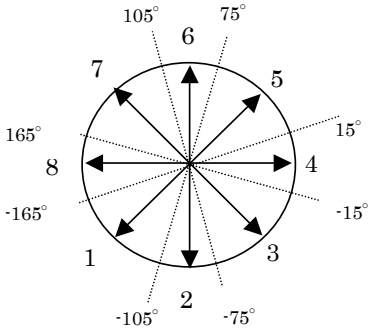


Fig. 4 Eight direction codes.

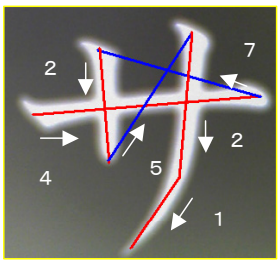


Fig. 5 Example: the direction codes

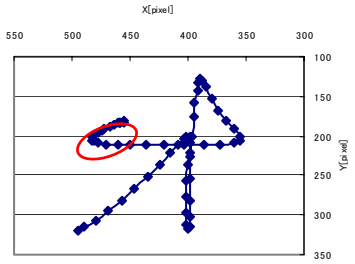


Fig. 6 Rejection of trembling points.

## III. Recognition Algorithm

### *A. Rejection of Trembling Points*

Sometimes, at the beginning of writing, light points are crowded and the movement distance ρ is very small. An example is shown in Fig. 6. We call these points "trembling points", and it is reasonable to reject these points from the locus of the character. If the movement distance ρ is smaller than 2 pixels, the direction code data for the light point is rejected.

### *B. Code Normalization by Speed*

The writing speeds are different for each person. These speeds also change during the strokes of a character. However, the image capture speed is constant (30 frames/second). Therefore, to remove the influence of writing speed, the direction codes are normalized to 100 data according to the displacement $\rho_i$. The original direction code series $h_j$ ( $j$= 1 to $max$) is normalized based on the displacements to the code series $H_k$ ($k$ = 1 to 100) having 100 direction codes. The $H_k$ is determined by the following equation.

$$H_k = h_j \quad , \tag{3}$$

$$1 \le k \le \frac{\rho_1}{\rho_{sum}} \times 100 \qquad \text{for} \quad j=1,$$

$$\frac{\sum_{i=1}^{j-1} \rho_i}{\rho_{sum}} \times 100 < k \le \frac{\sum_{i=1}^{j} \rho_i}{\rho_{sum}} \times 100 \qquad \text{for} \quad 2 \le j \le \max ,$$

where $\rho_{sum}$ is the total length of strokes, given by

$$\rho_{sum} = \sum_{i=1}^{max} \rho_i$$

Fig. 7 shows $x$, $y$ positions of light points of a character for each frame. The original 41 direction codes and the speeds for each frame are shown in Fig. 8. It is clear that the speed of the LED pen changes drastically, from 0 to 26 pixels/frame, over the course of drawing the character. Fig. 9 shows the result of the speed normalization. The normalization is especially effective at the beginning of a character.

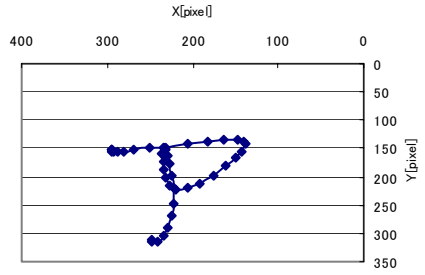


Fig. 7 $x, y$ positions of light points for each frame of a character.

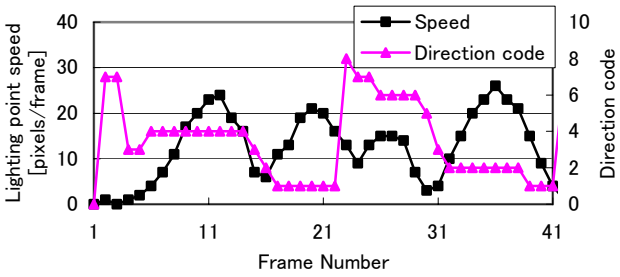


Fig. 8 Speed of the LED pen and the original direction codes.

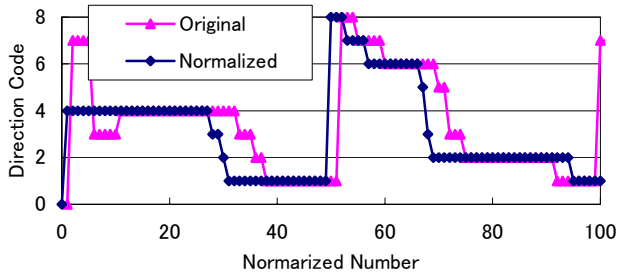


Fig. 9 Conversion to speed-normalized direction codes.

## C. Partial Matching

The normalized direction codes of a hand gesture are compared with direction codes of model data. Fig. 10 shows the matching of a gesture with model data. Some mismatching occurs at direction code changing points. This is because the percentages of each stroke differ slightly according to the writer. Direction code data deviating by +/- 2 directions codes from the model, indicated by gray in Fig. 10, are eliminated from the matching-test data.

The differences of direction codes ($\delta d$) between test dataand model data are calculated by equation (4). Because the direction codes correspond to a circle, as shown in Fig. 4, the maximum possible difference is 4. That is, if the difference of the direction code numbers is 7, the real difference is 1.

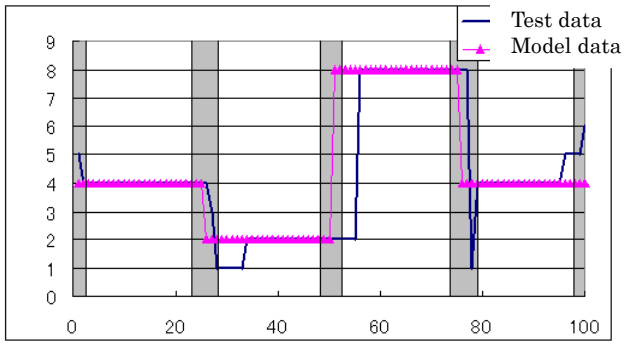


Fig. 10 Partial Matching of Katakana 'コ' (ko).

Equation (5) defines the mean of the errors. The final decision is done by finding the smallest $E$ from among all characters.

$$\delta d = |d_{\text{test}} - d_{\text{model}}| \; ; \quad \text{if } \delta d > 4 \quad \text{then } \delta E = 8 - \delta d \; ; \tag{4}$$

$$E = \frac{\sqrt{\sum_{i=1}^{100-Ne} (\delta E_i)^2}}{100 - Ne}, \tag{5}$$

where $Ne$ is the number of eliminated data.

## D. Oblique Characters

It is difficult for a person lying in bed to write a character so that the character is upright with respect to a camera. In this case, the person writes a straight line that is horizontal from his or her perspective. The line is written right to left, as shown in Fig. 11. The system calculates the deviation σ of angle θ using equation (6), and if the deviation σ is smaller than threshold $\sigma_t$, the system changes the definition of the horizontal line angle to $\theta_{ave}$ defined by equation (7). The range for each direction code shown in Fig. 4 is rotated by angle $\theta_{ave}$. Fig. 12 shows the new horizontal line $X$ for the oblique character.

$$\sigma = \frac{\sqrt{\sum_{i=1}^{N} (\theta_i - \theta_{ave})^2}}{N-1}, \tag{6}$$

where

$$\theta_{ave} = \frac{\sum_{i=1}^{N} \theta_i}{N}. \tag{7}$$

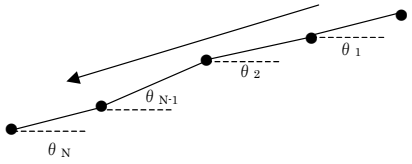


Fig. 11 Detection of horizontal angle $\theta_{ave}$.

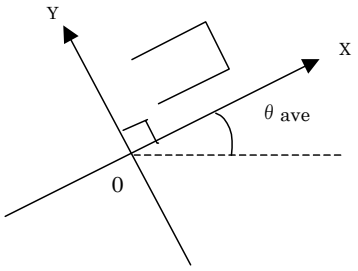


Fig. 12 Oblique character and the new horizontal line *X*.

## IV. Multi-camera System

In the multi-camera system, a person performs a character handwriting gesture in the air, and four TV cameras are used to capture the data of the LED light points. Because the single-camera system used only one camera, the operator was required to write the letters in view of the camera. However, it is sometimes difficult to find a camera for a character handwriting gesture. Therefore, we developed a system that does not depend on the gesture direction. Fig. 13 shows an example of the proposed multi-camera system.

### *A. Selection of Two Cameras*

Four TV cameras detect the movements of a LED pen. The surface of the LED is modified to increase light diffusion. A button on the LED pen is pressed when writing a character.

Two cameras are selected for use in calculating the 3-D positions of the light points. This selection is performed by comparing the areas of the light-point trajectories obtained from the four cameras. The trajectory area is defined as the area of the smallest rectangle that encloses the trajectory. First, the camera having the largest trajectory area is selected. Then, the trajectory areas of the two adjacent cameras are compared, and the camera having the larger area is selected. Fig. 14 shows the images obtained by the four cameras. In Fig. 14, cameras 1 and 2 are selected for the calculation of the 3-D positions of the light points.

### *B. Detection of 3-D Position*

The 3-D positions of light points are calculated using two cameras. The coordinates of the light points for the left-hand camera are represented as $(x_c,y_c)$ and those for the right-hand camera are represented as $(x_p,y_p)$.

The 3-D coordinates $(X,Y,Z)$ of the light points are calculated as follows:

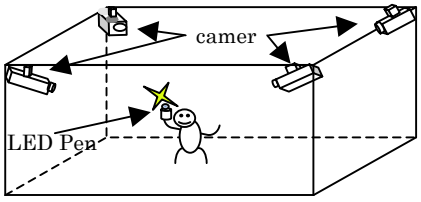


Fig. 13 Gesture recognition system with multi-camera.

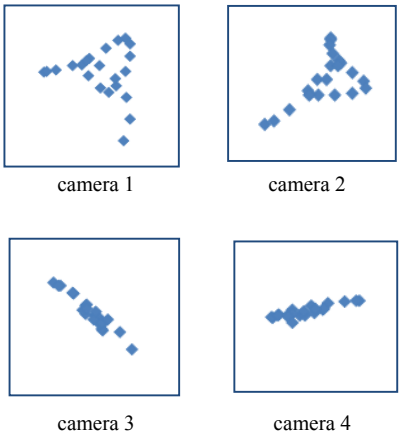


Fig. 14 Trajectories of light points for four cameras.

$$\begin{pmatrix} C_{11}-C_{31}x_c & C_{12}-C_{32}x_c & C_{13}-C_{33}x_c \\ C_{21}-C_{31}y_c & C_{22}-C_{32}y_c & C_{23}-C_{33}y_c \\ P_{11}-P_{31}x_p & P_{12}-P_{32}x_p & P_{13}-P_{33}x_p \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} x_c-C_{14} \\ y_c-C_{24} \\ x_p-P_{14} \end{pmatrix}, \quad (8)$$

where $C_{ij}$ and $P_{ij}$ are camera parameter values of right-hand and left-hand cameras, respectively, which are obtained from camera calibration procedure [10].

### *C. Conversion to the Front View*

The character drawing plane is calculated from the 3-D coordinates (X,Y,Z) of the light points. The plane is expressed by the following equation:

$$aX + bY + cZ + d = 0 \quad (9)$$

where a, b, and c are the components of a vector normal to the plane. The values of a, b, and c are obtained by solving the following equation:

$$\begin{pmatrix} \sum_{i=1}^{n} X_i^2 & \sum_{i=1}^{n} X_i Y_i & \sum_{i=1}^{n} X_i Z_i \\ \sum_{i=1}^{n} X_i Y_i & \sum_{i=1}^{n} Y_i^2 & \sum_{i=1}^{n} Y_i Z_i \\ \sum_{i=1}^{n} X_i Z_i & \sum_{i=1}^{n} Y_i Z_i & \sum_{i=1}^{n} Z_i^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = - \begin{pmatrix} \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} Z_i \end{pmatrix} \quad (10)$$

where n is the total number of light points.

To obtain a front view of the character, the drawing plane is rotated so that the normal vector is matched to the Z-axis. The rotation angles $(\theta,\phi)$ are calculated by the following equations:

$$\text{X-axis rotation angle: } \theta = \sin^{-1} \frac{b}{\sqrt{b^2+c^2}}, \quad (11)$$

and

$$\text{Y-axis rotation angle: } \phi = \sin^{-1} \frac{a}{\sqrt{a^2+b^2}}. \quad (12)$$

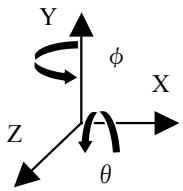

Fig. 15 Rotation angles.

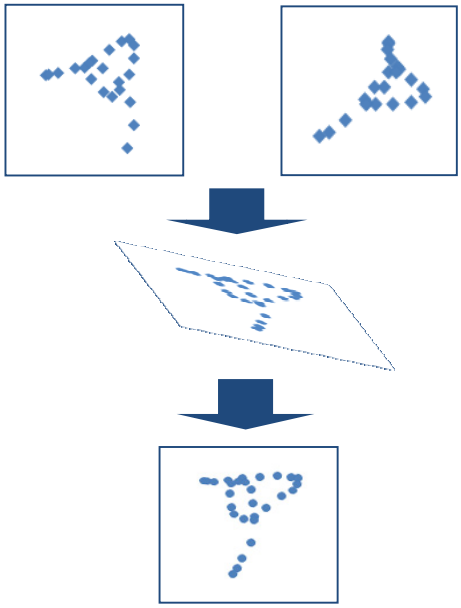

Fig. 16 Conversion to the front view.

Fig. 15 shows the rotation angles. Fig. 16 shows the procedure for the conversion to the front view.

## V. Experiments

### A. Experimental Set-Up

*1) Single Camera System*—A CCD TV camera (1/2 inch CCD, f = 6 mm) picks-up the LED movements. An image processor IP7000 (Hitachi) is used for image capture and recognition. The image size is 512 x 440 pixels.

*2) Multi-Camera System*—Fig. 17 shows the experimental set-up. The distance between the cameras is 3 m. The camera calibration was performed using eight corner points of a 90 cm cubic angle iron jig.

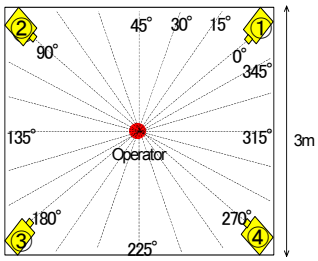

Fig. 17 Experimental set-up for the multi-camera system.

### B. Single Camera System Results and Discussions

*1) Test 1 (1 Person)*—Hand gestures were repeated 10 times for 46 characters and 2 consonant diacritics. The model data are prepared for the specific person. The average recognition success rate was 92.9%. The success rate was 79.4% without the partial matching algorithm. This is the improvement of 13.5%.

*2) Test 2 (15 People)*—The 15 test subjects (all male students) tried *katakana* hand gestures. They drew each character 10 times. The direction codes of the model data were previously determined from standard *katakana* fonts. The recognition success rate depended very much on the text subject. The highest success rate among the 15 subjects was 94.6%, the lowest rate was 74.4%, and the mean rate was 84.7%. Table 1 shows the success rates for each character. The diacritic gestures '゛' and '゜' were recognized at high rates.

It was difficult to distinguish between some characters due to similarities in their direction code patterns.

TABLE 1
Recognition Rates for the 15 Subjects (%)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ア | 55.3 | カ | 96.7 | サ | 88 | タ | 96 | ナ | 98 |
| イ | 74.7 | キ | 88 | シ | 89.3 | チ | 92 | ニ | 82 |
| ウ | 68 | ク | 93.3 | ス | 86.7 | ツ | 96.7 | ヌ | 72 |
| エ | 82.7 | ケ | 66 | セ | 60 | テ | 83.3 | ネ | 98.7 |
| オ | 86.7 | コ | 96 | ソ | 63.3 | ト | 87.3 | ノ | 100 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ハ | 96.7 | マ | 60.7 | ヤ | 64.7 | ラ | 76.7 | ワ | 75.3 |
| ヒ | 92 | ミ | 87.3 | - | - | リ | 84 | ヲ | 86 |
| フ | 97.3 | ム | 68 | ユ | 89.3 | ル | 96.7 | ン | 78 |
| ヘ | 100 | メ | 88 | - | - | レ | 100 | ゛ | 94.7 |
| ホ | 76 | モ | 78 | ヨ | 94.7 | ロ | 89.3 | ゜ | 92.7 |

### C. Multi-Camera System Results and Discussions

*1) Recognition Rate*—The recognition rates of the single-camera system and the multi-camera system were compared. The writing angles were varied in increments of 15º, and each character recognition rate was measured.

Gestures for each character were performed 20 times. Figs. 18 and 19 show the success rates for each character. In the single-camera system, a high recognition rate was obtained only in front of the camera. On the other hand, high recognition rates were obtained for any directions using the multi-camera system. The processing time of the image processor was 0.4 sec/character. Fig. 14 shows the overall success rate for four successive characters.

*2) Direction Distinction Accuracy*—The directions in which the characters were written can be detected by the Y-axis rotation angle ϕ given in Equation (12). The standard deviation of the angle ϕ was tested for each direction and was found to be 4.5°. Therefore, this system has the ability to distinguish direction in increments of 9° (2σ). This means that 20 directions can be distinguished.

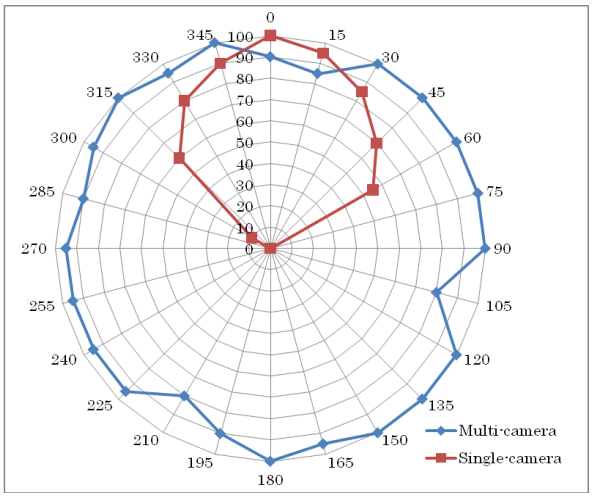


Fig. 18 Recognition rates for character 'ヒ' (hi).

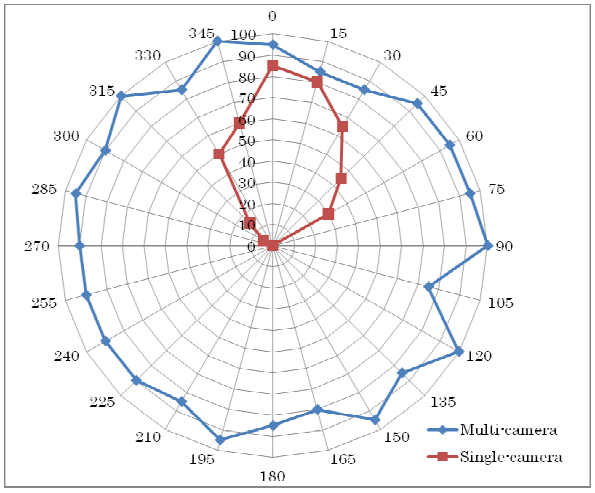


Fig. 19 Recognition rates for character 'ロ' (ro).

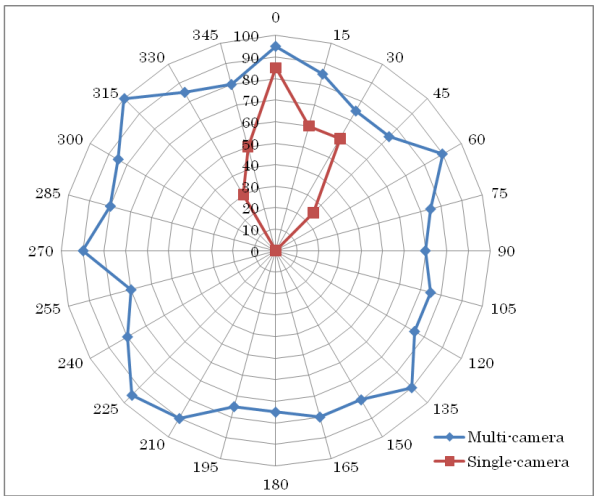


Fig. 20 Recognition rates for characters 'ヒロシマ' (hiroshima).

One application of this technique could be for remote control for home appliances. There are many infrared remote controllers in use, for example, for TV sets, air conditioners, hi-fi sets, and video-recorders. The number of controllers could be reduced by adopting gesture recognition systems.

## V. Conclusion

We have developed two visual interface systems, a single-camera system and a multi-camera system, that enable recognition of *katakana* characters written in a single stroke using a LED pen. TV cameras are used to capture the light trajectories and these trajectories are converted into direction codes. The direction codes are normalized to remove variations in writing speed. The code series are compared with model data in which the *katakana* direction codes are defined. In the experiment, we have achieved a mean recognition rate of 92.9% for one subject and 84.7% overall for the 15 test subjects. It is possible to extend these systems to include *hiragana* characters, simple *kanji* characters, and English alphabets.

The multi-camera system can recognize handwriting character gestures from any angles. The 3-D positions of the light points are calculated, and the data are transformed into data representing the front view position. The success rate for recognition was 90%. The multi-camera system also has the ability to distinguish direction in increments of 9º.

Future work will be to improve the reliability of the systems and to develop a robot system that can recognize hand character gestures and can talk with a person.

## References

[1] Tomonari Sonoda and Yoichi Muraoka, "A Letter Input System of Handwriting Gesture," *IEICE Trans.* Vol.J86-D-Ⅱ, No.7, pp.1015-1025, 2003 (in Japanese).

[2] Masaji Katagiri and Toshiaki Sugimura, "Personal Authentication by Signatures in the Air with a Video Camera," *Technical Report of IEICE*, PRMU2001-34, pp.9-16, 2001 (in Japanese).

[3] Hee-Deok Yang, A-Yeon Park, Seong-Whan Lee, "Gesture Spotting and Recognition for Human-Robot Interaction," *IEEE Trans .ROBOTICS*, Vol.23, No.2, pp.256-270, 2007

[4] Thierry Artieres, Sanparith Marukatat, Patrick Gallinari,"Online Handwritten Shape Recognition Using Segmental Hidden Markov Models," *IEEE Trans. PAMI*, Vol.29, No.2, pp.205-217, 2007

[5] Jae-Wan Park, Jong-Gu Kim, Dong-Min Kim, Min-Yeong Chong, Chil-Woo Lee, "Learning Touch Gestures using HMM on Tabletop Display, " *Proc. 16th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, pp.427-430, 2010

[6] Hyeon-Kyu Lee and Jin H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Trans. PAMI*, Vol.21, No.10, pp.961-973, 1999.

[7] Kota IRIE, Kazunori UMEDA,"Detection of Waving Hands from Images - Application of FFT to time series of intensity values -",*Proc. 3rd China-Japan Symposium on Mechatronics,* pp.79-83, 2002.

[8] Kota Irie, Naohiro Wakamura, Kazunori Umeda, "Construction of an Intelligent Room Based on Gesture Recognition ," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.193-198, 2004.

[9] Ho-Sub Yoon, Jung Soh, Byung-Woo Min, and Hyun Seung Yang, " Recognition of Alphabetical Hand Gestures Using Hidden Markov Model," *IEICE Trans*. Vol.E82-A, No.7, pp.1358-1366 ,1999.

[10] Tim Morris, *Computer vision and image processing*, Palgrave macmillan, p.226, 2004.