1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

HAND GESTURE COMMANDS FOR SLIDE VIEW CONTROL IN A PC BASED PRESENTATION

C. Y. Chen, Y. P. Fan, Z. Chen and H. L. Chou

Institute of Computer and Information Engineering

National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

E-mail: {zchen, hlchou}@csie.nctu.edu.tw

Abstract

In this paper hand gesture commands for slide view control in a PC based presentation system are addressed. First of all, a set of natural hand gestures is defined which will replace the mouse functions to control the slide view on the screen in a PC based presentation system. The advantages of using the natural hand gestures include: (1) no more disturbance during the slide presentation by abolishing the mouse, (2) no need of holding a laser pointer or a stick for pointing at the screen and (3) ease in using the hand gesture commands for slide view control.

Our system consists of four components: geometric configuration setup, hand segmentation, hand gesture recognition, and slide view control by hand gesture commands. Experimental results on using the hand gestures for slide presentation are reported.

Keyword: Hand Gesture, Geometric Configuration, Camera Calibration

I. Introduction

Nowadays the use of a PC based presentation system becomes common practice. The reasons for using such a presentation system include: (1) ease in updating or reorganizing the presentation slides at any time (2) ease in file management of the presentation slides. We assume that the presenter has created his presentation slides with a presentation editing software tool (Microsoft PowerPoint in our case). In the following we shall be concerned with the oral presentation delivered by the presenter using his or her slides. During the presentation the presenter uses a mouse or keyboard to step through the slides which are projected on the big white screen through a multimedia projector (a 3M multimedia projector in our case) or a LCD display panel sitting on an overhead projector. From time to time the presenter has to bent over to look down at the PC monitor display and commands the mouse in order to move the slide around or to change the slide scale. In addition, the presenter must hold a laser pointer or a pointing stick in hand in order to point at a specific slide word/line or a graph for presentation. Apparently, the presenter will interrupt his presentation when the presenter has to use the mouse or keyboard during presentation. Therefore, it seems very desirable if the presenter can simply use the natural hand gestures to command the slide show without using the mouse or a pointing device (a laser pointer or a stick).

A typical scenario in a hand-gesture based presentation system is as follows. A presenter stands in front of a presentation screen and points at the text/graphic data shown on the screen using his or her bare hand. To replace a mouse and a pointing device in controlling the slide presentation, the following operations must be provided:

(i) the function of a pointing device will be substituted by a vision-based system which projects the pointing direction of the index finger onto the presentation screen to locate a text/graphic symbol. This is referred to as the pointing function of the hand.

(ii) the movement of a cursor which locates a text/graphic symbol on the monitor screen will be accomplished in a similar way as stated above.

(iii) the drawing function of a cursor which creates a text/graphic symbol will be also accomplished through the pointing function of the hand.

(iv) the slide translation and size change are generally done by moving the mouse around to click the vertical/horizontal scroll bars or a combobox for character size specification. These mouse commands and cursor operations will be replaced through a set of hand gestures together with the simulated cursor movements.

One key component of our vision-based PC presentation system is the hand gesture recognition. There are some existing hand gesture recognition methods; some use a color glove to be fit with the hand in order to simplify the hand segmentation [1]-[2]; some constrain the freedom in hand movement [3]-[4]; some focus on segmentation of hand using an image sequence [5]; some are concerned with only the pointing gesture [6]; and there are also other approaches [7]-[10].

In this paper we will define three modes of natural hand gestures for slide view control: (1) slide movement or zooming, (2) slide AOI (area of interest) pointing, and (3) null operation. In practice, the pointing mode can be also used to implement the other commands for the slide view control by pointing to the proper icons on the

monitor screen, but this way of implementation is not considered natural or intelligent.

The system block diagram is shown in Fig.1. The system components are geometric configuration setup, hand segmentation, hand gesture recognition, and slide view control by hand gestures.

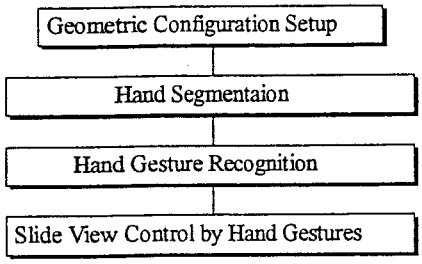


Fig.1. System block diagram

II. Geometric Configuration Setup

In order to free the presenter from holding a pointing device or operating a mouse, we not only recognize the hand gestures which control the slides but also estimate the pointing direction of the finger which locates the area of interest on the screen. To do so, we need to set up an appropriate geometric configuration of the system hardware components shown in Fig.2. In the system, cameras, cameras are fastened in the meeting room, and the presentation screen and the multimedia projector are fixed after being properly adjusted. It is important to note that the geometric relation between the presenter's hand and the presentation screen plays a key role in our hand gesture based presentation system. This is because the hand may serve as a pointing device. Here the vision-based technique using the TV cameras will be employed to determine the 3D locations of both of the hand and the presentation screen in the world coordinate system attached to the room. The following coordinate transformation matrices will be needed.

(a) $\mathbf{H}_{\text{World-to-Camera}}$ (abbreviated as **H**), the transformation from the world coordinates to the camera coordinates.

(b) $\mathbf{H}_{\text{PresentationScreen-to-ProjectorLCDScreen}}$ (abbreviated as **M**), the transformation from the presentation screen to the projector LCD screen. Here the projector LCD screen is the same as the monitor screen.

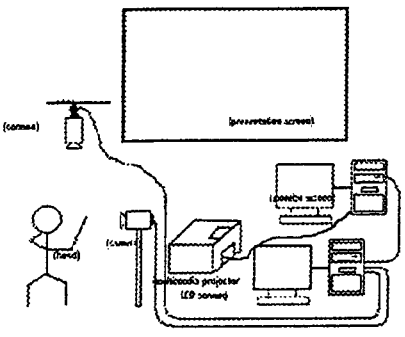


Fig.2. The geometric configuration of the system

All of the above transformation matrices must be obtained through some calibration processes in advance. During the presentation session, the two cameras (one fastened on the room ceiling and one fixed on a fixture) constantly track the location of the presenter's hand (only the presence of a raised hand indicates a meaningful operation). What we want to do with regard to the hand images is to find, from the top and side views, the hand gesture type or the hand's pointing vector in the world coordinates when it is in a pointing mode. These will be desirable in the next section. We shall first present the calibration procedures to obtain the transformation matrices **H** and **M**.

The relationship between the camera coordinate system and the world coordinate system is to be derived. Based on the homogeneous coordinates, the relationship can be expressed as

$$\begin{bmatrix} u \cdot w \\ v \cdot w \\ w \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (1)$$

where $(u, v)^T$ is the image point and $(X, Y, Z)^T$ is its corresponding 3D point. The 3-by-4 matrix, **H**, is called the projection matrix. By eliminating w in Eq.(1), we can obtain

$$\begin{aligned} (H_{11}-H_{31}u)X+(H_{12}-H_{32}u)Y+(H_{13}-H_{33}u)Z+(H_{14}-H_{34}u)=0 \\ (H_{21}-H_{31}v)X+(H_{22}-H_{32}v)Y+(H_{23}-H_{33}v)Z+(H_{24}-H_{34}v)=0 \end{aligned} \qquad (2)$$

At least 6 pairs of 2D image points and their corresponding 3D points are needed to solve the 12 unknown entries in the matrix **H**. Thus, if there are N corresponding point pairs, we will have the following overdetermined linear system

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -uX & -uY & -uZ & -u \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -vX & -vY & -vZ & -v \\ & & & & & & \vdots & & & & & \\ & & & & & & \vdots & & & & & \\ X & Y & Z & 1 & 0 & 0 & 0 & 0 & -uX & -uY & -uZ & -u \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -vX & -vY & -vZ & -v \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{14} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{24} \\ H_{31} \\ H_{32} \\ H_{33} \\ H_{34} \end{bmatrix} = 0$$

Also, the six point pairs must not be coplanar; otherwise, other calibration approach must be applied [11]. In our geometric configuration setup step, more than six correspondence pairs on two different 3D planes are used, a least squares solution to the overdetermined linear system can be found. There are two cameras. Each of them is calibrated using the above procedure.

Next, the geometric relationship between the PC monitor screen and the presentation screen will be derived. The relationship can be described as Eq.(1), assuming the projector satisfies a pinhole model. Since all 3D points on the big white presentation screen satisfy a plane equation

aX+bY+cZ=d, we can replace the Z component in Equation 1 with (d-aX-bY)/c. Thus we shall have

$$\begin{bmatrix} u\cdot w \\ v\cdot w \\ w \end{bmatrix} = \begin{bmatrix} H_{11} H_{12} H_{13} H_{14} \\ H_{21} H_{22} H_{23} H_{24} \\ H_{31} H_{32} H_{33} H_{34} \end{bmatrix} \bullet \begin{bmatrix} X \\ Y \\ (d-aX-bY)/c \\ 1 \end{bmatrix} \qquad (3)$$

After some matrix operations, we obtain

$$\begin{bmatrix} u\cdot w \\ v\cdot w \\ w \end{bmatrix} = \begin{bmatrix} (H_{11}-H_{13}\,a/c)X+(H_{12}-H_{13}\,b/c)Y+(H_{14}-H_{13}\,d/c) \\ (H_{21}-H_{23}\,a/c)X+(H_{22}-H_{23}\,b/c)Y+(H_{24}-H_{23}\,d/c) \\ (H_{31}-H_{33}\,a/c)X+(H_{32}-H_{33}\,b/c)Y+(H_{34}-H_{33}\,d/c) \end{bmatrix} \qquad (4)$$

In other words,

$$\begin{bmatrix} u\cdot w \\ v\cdot w \\ w \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \qquad (5)$$

where

$$\begin{cases} M_{11} = H_{11} - H_{13}a/c \\ M_{12} = H_{12} - H_{13}b/c \\ M_{13} = H_{13} + H_{13}d/c \\ M_{21} = H_{21} - H_{23}a/c \\ M_{22} = H_{22} - H_{23}b/c \\ M_{23} = H_{24} + H_{23}d/c \\ M_{31} = H_{31} - H_{33}a/c \\ M_{32} = H_{32} - H_{33}b/c \\ M_{33} = H_{34} + H_{33}d/c \end{cases}$$

Now the 3-by-4 projection matrix can be reduced to a 3-by-3 matrix, **M**. It relates the big white presentation screen to the PC monitor screen. Therefore, for any 3D point $(X, Y, Z)^T$ on the big white presentation screen, its corresponding 2D point on the PC monitor screen can be found through the matrix **M**.

Ⅲ. Image Segmentation of Hand

A primary step in our hand gesture controlled PC presentation is to extract the hand from the images. In a natural environment the background is rather complex, so the hand segmentation is a difficult task and time consuming. However, in a real time system such as the PC presentation system, a fast segmentation is needed. In the conference or meeting room the presenter usually stands in between the audience and the PC with the visual equipment. It is reasonable to assume that the background seen in the top or side view is virtually static. We shall take a picture of the static background scene as a reference frame first. After the presenter enters into the scene, new pictures of the scene are taken repeatedly. Each new picture will then subtract the background picture to produce an absolute difference picture.

The difference picture is basically bimodal, so it can be easily thresholded. The pixels with nonzero gray value mainly correspond to the presenter's body, although there may be some other regions (generally smaller) due to shadow or other lighting effects. Sometimes, the lighting condition is either too bright (especially, at the day time or with too much of lighting) or too dim (at the night time and with weak lighting) the segmented hand region may contain small gaps or holes. To smooth out the noise, a closing morphological operation is applied to the segmentation result.

There are two TV cameras for taking the pictures, one roughly from the top and one roughly from the side. During the presentation session, the presenter's hand is presumably raised and moves away from the torso, so it is easy to detect columnwise or rowwise a big width change. This is the place where the hand and the torso are connected. Therefore, at this place we can break the hand away from the human body. In the image a rectangular window (or box) covering the hand portion only can be drawn. This window will be the area of interest to be processed for hand gesture recognition.

Ⅳ. Hand Gesture Recognition

The first type of hand gestures is the slide movements in the up, down, left, right directions and slide zoom-in and zoom-out. The second type is hand gestures for pointing at the slide or clicking the mouse. The third type is a null operation. The images of the corresponding hand gestures are shown in Figs. 3(a)-3(h). The underlying theory for the hand gesture recognition is given below.

(a) The hand gestures for the move-up and move-down commands are virtually the same as the hand gestures for move-left and move-right ones (The first two gestures are fully visible from the side view, while the last two are from the top view). They differ by a rotation. The recognition of all the four hand gestures involves the detection of a hand formation consisting of a stretched thumb and four other clenched fingers.

(b) The hand gestures for zoom-out and zoom-in are fully visible to the side view TV camera. First, the upper and lower arms are detected to form an angle of about 90 degrees. Next, the image of the segmented lower arm is rotated clockwise by 90 degrees, then it is further analyzed as in the case of the move up or down hand gesture. If the thumb is stretched, it indicates a zoom-in command and if a finger is stretched away from the body, it indicates a zoom-out command.

(c) The hand gesture for location pointing is claimed if the arm and the index finger are stretched; if only the arm is stretched, but no fingers are stretched, then the hand is in the formation of a clenched fist. The clenched fist mode indicates " clicking the mouse ".

(d) If none of the above hand gestures are detected, a null operation is declared.

Next, we need to specify a way to detect if any finger of the hand sticks out. This is done based on the window previously found during the segmentation process in which the hand and the lower arm appear. Without loss of the generality, let us assume the arm and hand are more or less aligned in the horizontal direction and $XL \leqq X \leqq XR$. We scan the image column by column and measure the average column width over the entire window,

denoted as W. We then find the maximum column width Wmax located at X=Xmax in the area of the hand and the width of the wrist W. Also, we find the minimum column width Wmin located at X=Xmin (assume $X_L \leqq X \leqq X_R$-W/5). The decision rule for recognizing a stretched finger is as follows:

(a) if Wmax $\geqq$ 1.5W and X_R-Xpeak<1.5W, then the thumb is stretched,

(b) if Wmin $\leqq$ 0.5W and 2.5W>X_R-Xpeak >W, then the index finger (or some other finger) is stretched,

(c) if Wmax<1.5W and Wmin>0.5W, then no finger is stretched.

If the hand gesture has been recognized as a pointing mode, the following steps are executed to find the 3D pointing vector in the world coordinates:

(i) the image of the arm (or the index finger) is fitted as a line and its 3D line equation is to be calculated using a computer vision technique based on the top and side views. First of all, the fitted line, l: au+bv=c, is obtained and the projection matrix, **H**, is known. Then, we can derive the back-projection plane containing the fitted image as follows:

Substituting u=(c-bv)/a into Eq. (1) yields

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}^{-1} \left(\begin{bmatrix} [(c-b)/a)\cdot v \cdot w] \\ v \cdot w \\ w \end{bmatrix} - \begin{bmatrix} H_{14} \\ H_{24} \\ H_{34} \end{bmatrix} \right) \quad (6)$$

After eliminating v and w, we obtain the back-projection plane equation as

$$(aH_{11} - cH_{21} + bH_{21})X + (aH_{12} - cH_{22} + bH_{22})Y + (aH_{13} - cH_{23} + bH_{23})Z + (aH_{14} - cH_{24} + bH_{24}) = 0 \quad (7)$$

From the two cameras, two back-projection planes can be derived. The 3D line equation for the arm is the intersection of these two back-projection planes.

(ii) the intersection point of the 3D arm equation of the arm with the presentation screen is to be found. With the 3D plane equation, AX+BY+CZ = D, of the presentation screen obtained during the geometric configuration setup step and the 3D line equation, $(X-X_0)/\alpha = (Y-Y_0)/\beta = Z-Z_0$, of the arm estimated in step (i), the intersection point can be found as follows:

Substituting $X = \alpha(Z- Z_0)+ X_0$ and $Y = \beta(Z- Z_0)+ Y_0$ into AX+BY+CZ = D

Then, the intersection can be found from the following equation

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 + \alpha(D - AX_0 - BY_0 - CZ_0)/(A\alpha + B\beta + C) \\ Y_0 + \beta(D - AX_0 - BY_0 - CZ_0)/(A\alpha + B\beta + C) \\ (D + A\alpha Z_0 + B\beta Z_0 - AX_0 - BY_0)/(A\alpha + B\beta + C) \end{bmatrix} \quad (8)$$

(iii) the intersection point is mapped to a 2D point in the PC monitor screen coordinate system. This screen point is taken as the location of the mouse pointer. While knowing the 3D position the presenter want to points, the corresponding point location on the PC monitor screen must be calculated. The transformation **M** between the big white presentation screen and the PC monitor screen has been estimated during the geometrical configuration setup step, so the corresponding 2D screen point can be found by the following equation

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} (M_{11}X + M_{12}Y + M_{13})/(M_{31}X + M_{32}Y + M_{33}) \\ (M_{21}X + M_{22}Y + M_{23})/(M_{31}X + M_{32}Y + M_{33}) \end{bmatrix} \quad (9)$$

Next, we describe how the slide view can be controlled by the hand gestures. When a hand gesture is recognized, its corresponding mouse control command will be activated so that the intended slide movement or slide zooming will be accomplished. This is implemented in the slide view session of Microsoft PowerPoint. For a given slide control command, the mouse pointer will be moved from the current position to its corresponding icon position appearing on the monitor screen. In this way, the command is executed.

V. Experimental Results

The hardware facilities of our system include two Punix TV cameras, one Matrox frame grabber, two Pentium 100 PCs, and one 3M multimedia projector. Recall that Figs. 3(a)-3(h) are the raw images of hand gestures. Figs. 4(a)-4(h) are the processed images after the application of the difference operation to the corresponding hand gestures given in Figs. 3(a)-3(h). Figs. 5(a)-5(d) show the smoothing results after the application of the closing morphological operation. Figs. 6(a)-6(d) show some of the segmentation results. The typical computer processing time for executing each command is around 1 to 2 seconds. It is expected to come down after the program code is optimized.

After the hand gestures are properly recognized, these hand gesture results will invoke the corresponding mouse functions in the slide view session of the Microsoft PowerPoint. Figs. 7(a)-7(f) show some of the slide view changes corresponding to the hand gesture commands.

A number of people in our laboratory have been invited to test the system. Most of the time the system works properly, but occasionally the system malfunctions. The reasons for these malfunctions are mainly three folded: (1) the hand gestures are somewhat ambiguous, (2) the contrast between the hand and the background is not sharp, and (3) the presenter steps out of the permissible zone so that the hand is not properly located in the image.

VI. Conclusions

We have presented hand gesture commands for slide view control in a PC based presentation. The system implementation techniques are given. The on line test of our system indicated our system worked most of the time. In the future we shall improve the processing speed of the system. On the other hand, we shall also modify our segmentation and recognition methods to allow additional and less constrained hand gestures commands.

References

[1] Yoshio Iwai, Ken Watanabe, Yasushi Yagi and Masahiko, "Gesture Recognition Using Colored Gloves", Proc. International Pattern Recognition, pp. 662-666, 1996.

[2] Kirsti Grobel and Hermann Hienz, "Video-Based Handshape Recognition Using a Handshape Structure Model in Real Time", Proc. International Conference on Pattern Recognition, pp. 446-450, 1996.

[3] Toshikazn Wada and Takashi Matsuyama, "Appearance Sphere: Background Model for Pan-Tilt-Zoom Camera", Proc. International Conference on Pattern Recognition, pp. 718-722, 1996.

[4] Shinchi Tamura, "Recognition of Sign Language Motion Images", Proc. International Pattern Recognition, pp. 343-353, 1988.

[5] Yuntao Cui and John J. Weng, "View-Based Hand Segmentation and Hand-sequence Recognition with Complex Backgrounds", Proc. International Conference Pattern Recognition, pp. 617-621, 1996.

[6] Yao-Strong Yang, Yi-Ping Hung, Chiou-Shann Fuh and Ing-Bor Hsieh, "Finger Tracking and Its Application to Free-Hand Pointer", Proc. International Conference on Computer Systems Technology for Industrial Applications '96, Hsinchu, Taiwan, pp.267-273, 1997.

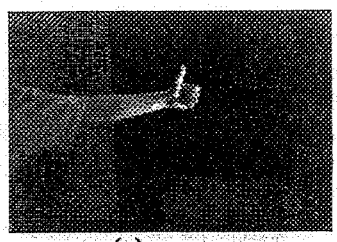
[7] James J. Kuch and Thomas S. Huang, "Vision Based Hand Modeling and Tracking for Virtual Teleconferencing and Telecollaboration", Proc. International Conference on Computer Vision, pp. 265-270, 1995.

[8] P.Nesi and A. del Bimbo, "A vision-based 3D mouse", Proc. International Conference on Computer Vision, pp. 229-234, 1995.

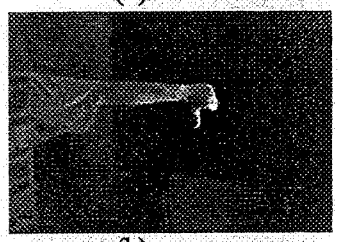
[9] Roberto Cipolle and Nicholas J. Hollinghurst, "Human-robot interface by pointing with uncalibrated stereo vision", Image and Vision Computing, 1996.

[10] Akria Utsumi, Tsutomu Miyasato, Fumio Kishino and Ryohei Nakatsa, "Hand Gesture Recognition System Using Multiple Cameras", Proc. International Pattern Recognition, pp. 667-671, 1996.

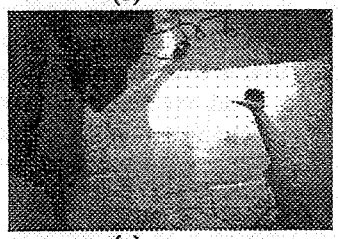
[11] S. W. Shih, Kinematic and Camera calibration of Reconfigurable Binocular Vision Systems, Ph.D. dissertation, Institute of Electrical Engineering, National Taiwan University, 1996.



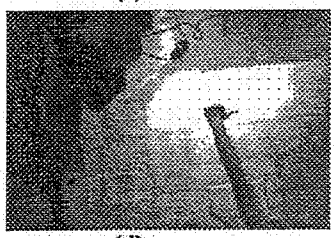
(a)



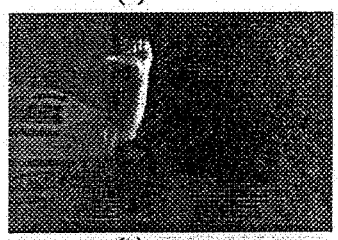
(b)



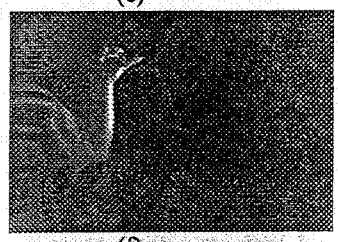
(c)



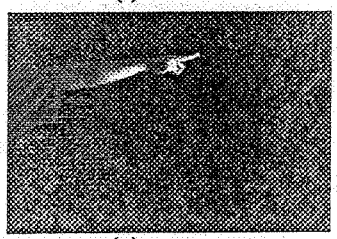
(d)



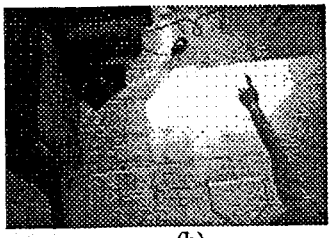
(e)



(f)



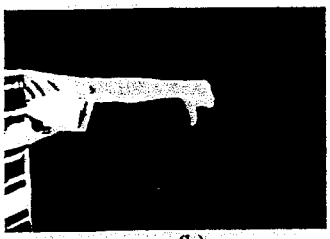
(g)



(h)

Fig.3 Hand gesture images: (a) move up, (b) move down, (c) move left,(d) move right, (e) zoom in, (f) zoom out, (g) pointing (the side view),pointing (the top view).

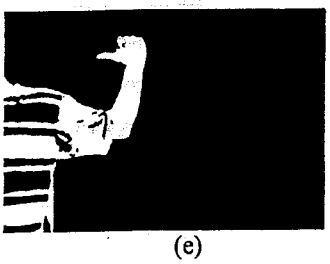
(a)



(b)

(c)

(d)



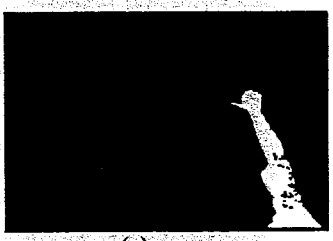
(e)

(f)

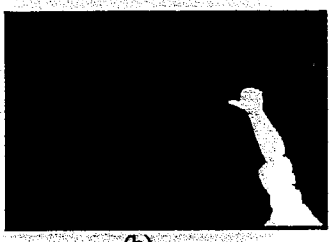
(g)

(h)

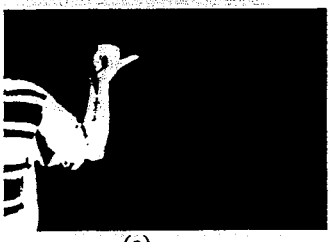
Fig.4 The processed images after the application of difference operation to images in Figs. 3(a)-3(h).



(a)



(b)



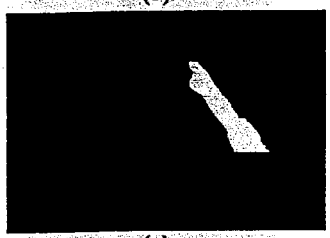
(c)

(d)

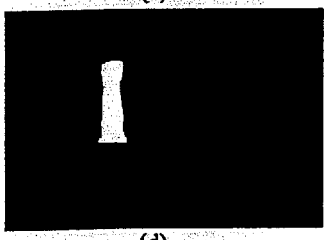
Fig.5 The smoothing effect of a morphological operation. (a),(c) the initial images, (b),(d) the processed images.

(a)

(b)

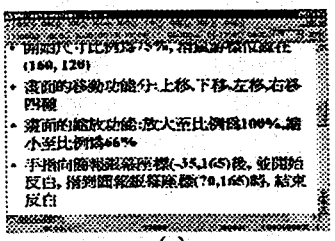


(c)

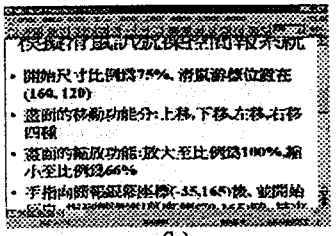


(d)

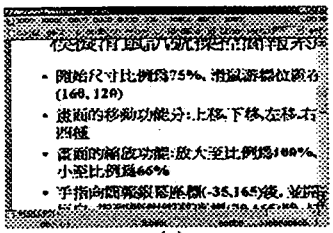
Fig.6 The segmentation results: (a) two arms, (b) lower arm, (c) pointing hand, (d) clenched fist.



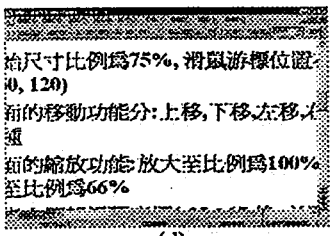
(a)



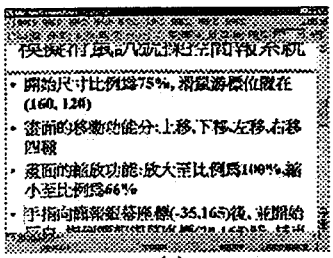
(b)



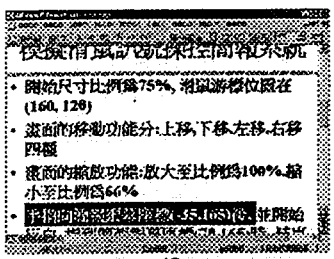
(c)



(d)



(e)



(f)

Fig.7 The slide control functions: (a) the original slide, (b) move-up, (c) move-right, (d) pointing (indicated by a vertical bar), (e) block marking.