S & M 1545

Real-time Hand Gesture Recognition System and Application

Hsiang-Yueh Lai,* Hao-Yuan Ke, and Yu-Chun Hsu

Department of Electrical Engineering, National Chin-Yi University of Technology,
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan (R.O.C.)

(Received October 30, 2017; accepted January 22, 2018)

Keywords: DSP, finger skin pixel, human–computer interaction, hand gesture recognition

Human–computer interaction (HCI) is a trend of technology evolution toward making our life more convenient and easier. In some circumstances where hands cannot conveniently touch equipment, the hand gesture recognition system is a good solution for HCI. A real-time hand gesture sensing and recognition system is proposed and its application to TV channel and volume control is examined in this study. The digital signal processor (DSP) DM6437 of Texas Instruments is used in our portable hand gesture recognition system. For the real-time recognition of hand gestures, we propose a novel finger skin pixel algorithm to quickly and easily distinguish the hand in a complex image. The region of interest is used to reduce the amount of computation. Hand features are detected in two steps. First, the thumb and pinkie are recognized on the basis of significant shape features. Second, a circle is defined from the hand center for threshold decomposition to count the number of fingers. The hand gesture recognition rate is 94.3%. The hand trace direction can be found easily by using a hand gesture center point in our system. Finally, our system is applied to TV channel and volume control using hand gestures and hand tracing.

1. Introduction

In some circumstances where hands cannot conveniently touch equipment, such as in medical environments and in the kitchen, the hand gesture recognition system is a good solution for human–computer interaction (HCI). Hand gesture recognition is also popularly applied in, for example, virtual simulation, sign language recognition, and computer games.(1)

Data gloves and vision-based recognition are popular and frequently used to capture images for hand gesture recognition. Although data gloves have higher accuracy, those equipped with many sensors are expensive. Data gloves(2) are also inconvenient for users who must wear them for gesture recognition. Therefore, the vision-based gesture recognition system for a user's bare hand is adopted in our research. Recently, a popular peripheral device called Microsoft Kinect(3) for Xbox 360 was developed for hand gesture recognition. The Kinect depth map improves the robustness of hand capture; it is, however, expensive. Thus, a low-cost CCD camera is used in our system.

*Corresponding author: e-mail: anne@ncut.edu.tw
http://dx.doi.org/10.18494/SAM.2018.1790

References 3–7 describe studies on image or gesture recognition mostly developed on PCs. In Ref. 8, a digital signal processor (DSP)-based gesture recognition system is described. It only traces the hand but does not identify hand gestures. Other DSP-based systems, which only recognized four hand gestures, are proposed in Refs. 9 and 10. TV channel control using the numbers 0 to 9 is difficult to accomplish remotely using only four hand gestures.

A DSP, DM6437 of Texas Instruments, is used for our portable hand gesture recognition system. For the real-time hand gesture sensing and recognition system, we propose a finger skin pixel algorithm to quickly and easily distinguish the hand in a complex image and use the region of interest to reduce the amount of computation. The hand trace direction can be found easily using a hand gesture center point in our system. Finally, our system is applied to TV channel and volume control using hand gestures and hand tracing.

2. System Hardware Platform

The system hardware framework is shown in Fig. 1. The DSP evaluation module (EVM) DM6437 EVM is developed by Texas Instruments.

First, the camera captures phase alternating line (PAL) images and transfers them to the DM6437 EVM for computation and recognization. The computation and recognition results are displayed on a monitor. The monitor is optional and can be removed after the system is fine tuned. The computation and recognition processes will be described in later sections. The 89SC51 and SC51P0304 circuits are designed as an infrared remote controller, which is connected to the DM6437 EVM with an RS-232 serial interface. The infrared remote controller achieves TV channel and volume control using the DM6437 EVM recognition results.

The single-core DSP-board DM6437 EVM DaVinci™ platform was developed by Texas Instruments (TI) in 2006. It is belongs to the TMS320C64x+™ series with corresponding function libraries for image processing. The clock rate of DM6437 can be up to 700 MHz, and it can realize 5600 million instructions per second (MIPS).[11] The integrated development environment of the DSP-board DM6437 is Code Composer Studio.

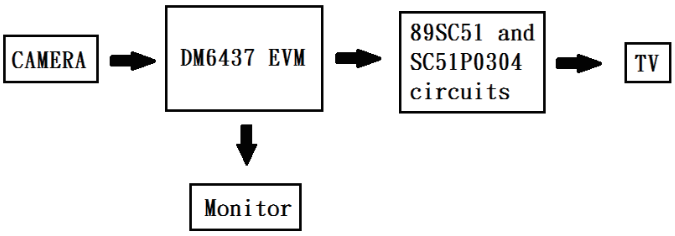


Fig. 1. Hardware framework.

3. Hand Region Detection

For a hand gesture recognition system, it is important to distinguish a hand from a complex background. The flowchart of hand region detection is shown in Fig. 2. First, the CCD camera captures dynamic images. Second, the image preprocessing step makes the images stabler and clearer. Third, the hand detection step identifies the hand region from the dynamic images with a complex background image.

3.1 Image preprocessing

The image preprocessing steps are as follows. First, a more stable color image is obtained by an automatic white balance (AWB) process. Second, the skin color regions in the image are detected, and then the image is changed into a binary image. Finally, the noise of the binary image is removed using morphological erosion and dilation.

3.1.1 AWB

The color space YUV is used for DM6437. Y means luminance, and U and V are chrominance. For reducing the amount of computation, the subsample method YUV422 is used

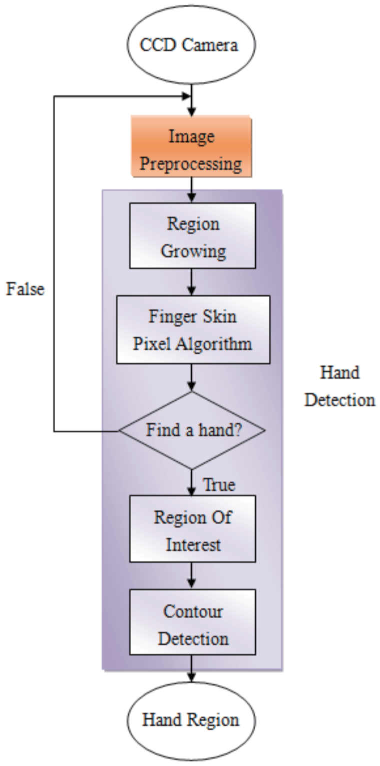


Fig. 2. (Color online) Flowchart of hand region detection.

for DM6437. A group of UY or VY is one pixel. A macropixel combines a UY pixel and a VY pixel, as shown in Fig. 3.

The captured image colors might differ owing to various light sources. Hence the perfect reflector assumption algorithm(12) is used to correct the white balance of the captured images. The brightest pixel in an image is considered as a white pixel. If the brightest pixel is not a white pixel, then correct its brightness. If the Y value of the brightest pixel is not 255, we correct it to 255, and the U and V of the brightest pixel are corrected to 128 in the macropixel. The corrected error values are applied to each pixel in the image to finish AWB.

An AWB example is shown in Fig. 4. The baseball is light yellow before AWB, as shown in Fig. 4(a). After AWB, as shown in Fig. 4(b), the color temperature of the baseball is changed. The captured images with AWB have stable luminance characteristics and improve the recognition results.

3.1.2 Skin color detection

The skin color model is used to determine whether a macropixel of a color image is a skin color pixel or nonskin color pixel. According to Ref. 13 and our experiments, the ranges of skin color are as follows.

$$\begin{cases} 65 < Y < 170 \\ 85 < U < 140 \\ 85 < V < 160 \end{cases} \tag{1}$$

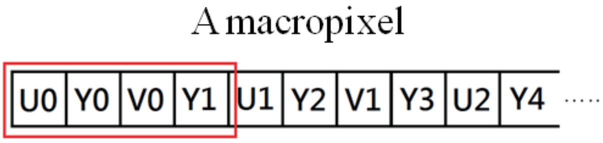
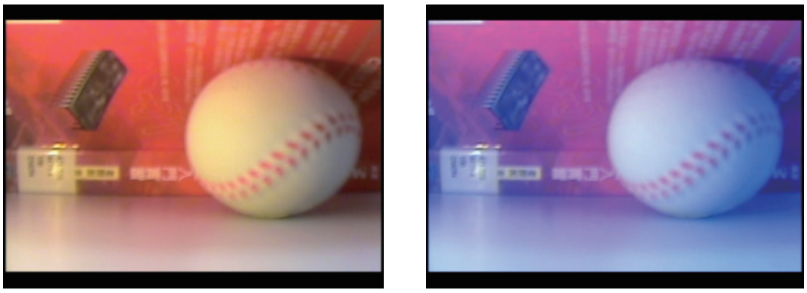


Fig. 3. (Color online) Order for YUV422.



(a) (b)

Fig. 4. (Color online) (a) Before AWB. (b) After AWB.

When the Y, U, or V in a macropixel satisfies Eq. (1), the macropixel is a skin color pixel. The skin color pixels are converted to white pixels and the others are converted to black pixels.

3.1.3 Noise cancellation

Noise is a troublesome problem in image processing. To obtain a cleaner image with a smoother contour, we adopt morphological erosion and dilation. The source image is shown in Fig. 5(a). Erosion can eliminate noise and small convex points in an image. The image will be shrunk after erosion, as shown in Fig. 5(b). Dilation can fill cavities in an image and then the image contours will be smoother. The image will be expanded after dilation, as shown in Fig. 5(c).

3.2 Hand detection

After image preprocessing, we obtain a clean skin color binary image and the hand region can be identified by hand detection described in Fig. 2. First, the continuous skin color regions from the skin color binary image are sectioned by the region growing method, and then the hand region is detected using the finger skin pixel algorithm. After determining the hand region, the computing region is reduced by the region of interest method. Finally, the hand counters are detected using 4-Neighbor.

3.2.1 Region growing method

Region growing is a region-based image segmentation method. It can gradually expand a small region into a large one. When a neighbor pixel of a region conforms to requirements, the pixel will become part of the region, until no pixel conforms. The region growing(14) conditions are as follows.

i) $\bigcup_{i=1}^{n} R_i = R$
ii) R_i is a connected region, $i = 1, 2, \ldots, n$
iii) $R_i \bigcap R_j = \emptyset$ for all $i = 1, 2, \ldots, n$

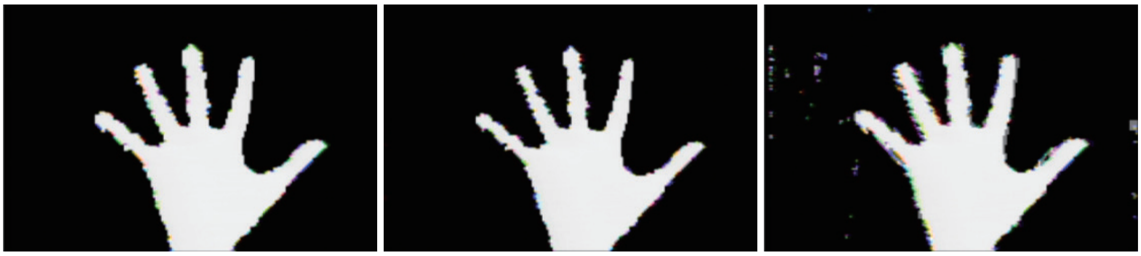

(a) (b) (c)

Fig. 5. (Color online) (a) Source image, (b) use erosion, and (c) dilation to reduce noise.

iv) $P(R_i) = TRUE$ for $i = 1, 2, \ldots, n$
v) $P\left(R_i \bigcup R_j\right) = FALSE$ for any adjacent region R_i and R_j

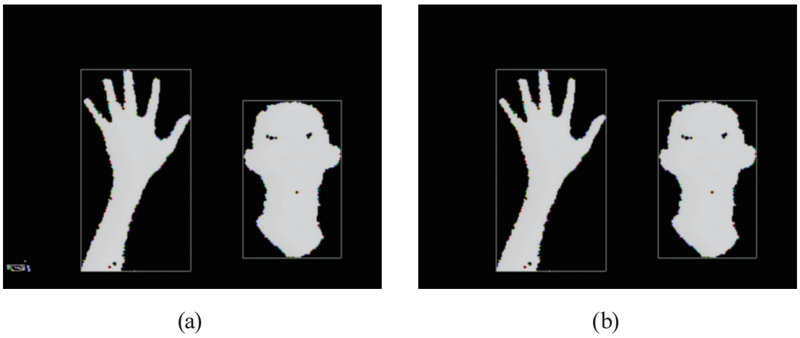
i) Each region must be present in the image R. ii) Each region is connected. iii) Each pixel can only be classified into one region. iv) The pixels have the same characteristics within the same region. v) The pixels in different adjacent regions have different characteristics.

First, we scan entire image pixels. An arbitrary white pixel is set as a starting point for region growing, and it expands outwards until it has no neighboring white pixels. We can identify skin color regions from an image, as shown in Fig. 6(a). Figure 6(b) shows face and hand regions after noise in Fig. 6(a) is removed with a threshold.

3.2.2 Identify a hand using finger skin pixel algorithm

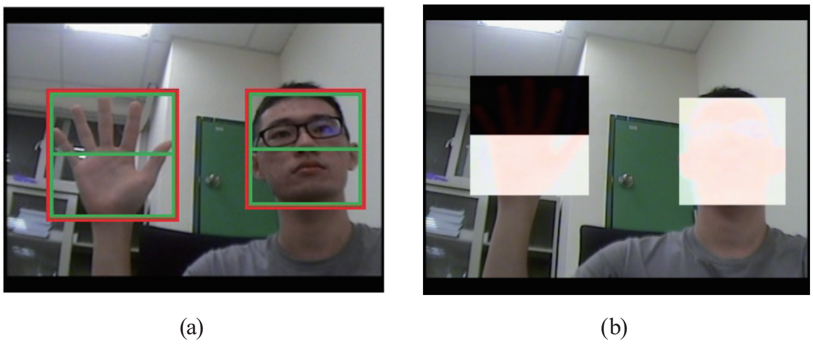
The hand and face are identified after region growing, and then the hand will be distinguished by hand detection. The finger skin pixel (FSP) algorithm is proposed to determine whether the area is a hand or a face using hand features, as follows.

First, the hand and face regions found by region growing are divided into two identification regions shown by the red blocks in Fig. 7(a). We further divide each identification region into 2



(a) (b)

Fig. 6. (a) Region growing result. (b) After removing noise.



(a) (b)

Fig. 7. (Color online) (a) Identification regions for FSP. (b) Identification results of the 2 × 1 matrices.

× 1 matrices, as shown by the green matrices in Fig. 7(a). The percentage of skin color pixels in each rectangular matrix box is calculated. The hand gestures are defined in our research with the fingers pointing upwards and natural stretch characteristics.

Table 1 shows the occupancy rates of skin color pixels in the 2 × 1 matrices of the hand performing different hand gestures. The occupancy rates are less than 30% in the upper rectangle, and are greater than 40% in the lower rectangle. Table 2 shows the occupancy rates of skin color pixels in the 2 × 1 matrices of the face with and without glasses. The occupancy rates of the upper and lower rectangles are both higher than 40%. Hence, the occupancy rate of 40% is defined as the threshold. The rectangle is set as a black block when the occupancy rate of skin color pixels is lower than 40%; otherwise, it is set as a white block. The identification results of the 2 × 1 matrices are shown in Fig. 7(b). The hand has a black and white matrix, and the face has only a white block; therefore, the black and white matrix is the first feature matrix of a hand.

The FSP identification regions with 2 × 1 matrices only can roughly identify a hand or a face from a skin color image, and the identification rate cannot reach 100% when using 2 × 1 matrices. A more sensitive detection is required. The identification regions of FSP are then divided into 8 × 8 matrices, and the threshold of the occupancy rate of skin color pixels is the same as the 40% of the 2 × 1 matrices. The identification results for different hand gestures using 8 × 8 matrices are shown in Fig. 8.

The black and white matrices are defined as feature matrices. In accordance with the features of the face and hand, the feature matrices of FSP are categorized into four groups, as shown in Fig. 9. The first feature matrix is the 2 × 1 matrix, which is used to roughly identify the skin color region of a hand or a face, as in Fig. 7. The black and white blocks of 8 × 8 matrices in Fig. 8 are categorized into the second, third, and fourth feature matrices. The second feature matrix is used to identify the vertical fingers, and this feature is the most frequently occurring in our recognition samples. The third feature matrix group is used to identify the lower left corner or the lower right corner of the hand. The fourth feature matrix group is used to identify a thumb, a pinkie or a sloping finger.

Table 1
Occupancy rates of skin color pixels in the hand matrices for different hand gestures.

Gesture	Rectangle	
	Upper (%)	Lower (%)
Gesture 0	27.35	43.25
Gesture 1	12.56	52.53
Gesture 2	28.87	57.65
Gesture 3	27.76	57.35
Gesture 4	29.44	57.85
Gesture 5	27.63	42.39
Gesture 6	15.61	61.51
Gesture 7	14.04	57.54
Gesture 8	21.84	60.38
Gesture 9	24.14	44.32

Table 2
Occupancy rates of skin color pixels in the face region.

Glasses	Rectangle	
	Upper (%)	Lower (%)
With glasses	49.88	74.99
Without glasses	63.11	66.71

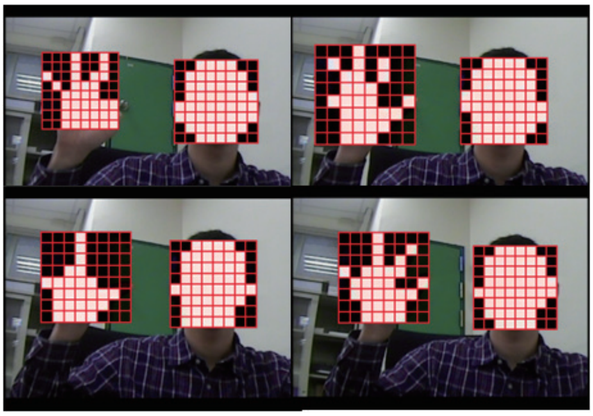


Fig. 8. (Color online) Identification results of 8 × 8 matrices.

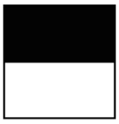
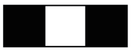
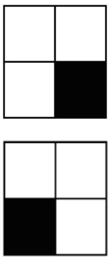
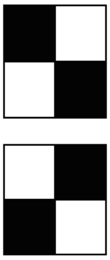
	Feature matrix group			
	First	Second	Third	Fourth
Feature matrix				
Weighting	0.4	0.3	0.2	0.1
Occurrence matrix	2 × 1 matrix	8 × 8 matrix	8 × 8 matrix	8 × 8 matrix

Fig. 9. FSP algorithm feature matrices.

Figure 10 shows the identification results with different hand gestures using the FSP feature matrices with complex backgrounds. The identification results with the second feature matrix are shown in the first row of Fig. 10, the third feature matrix in the second and third rows of Fig. 10, and the fourth feature matrix in the fourth and fifth rows of Fig. 10.

Finally, the total weightings in the FSP identification regions are calculated for a more accurate identification of the hand region. The weighting of each feature matrix is defined by the feature occurrence probability, as shown in Fig. 9. An FSP identification region is the hand region if it has a larger total weighting equal to or greater than 0.6 and the face region has a smaller total weighting.

3.2.3 Hand contour detection

The region of interest (ROI) is used to reduce calculation time. The operation region is focused on the hand. The ROI is limited to inside the rectangle whose sides are 50 pixels from the FSP frame. ROI will be the next frame search region, as shown in Fig. 11(a). The image

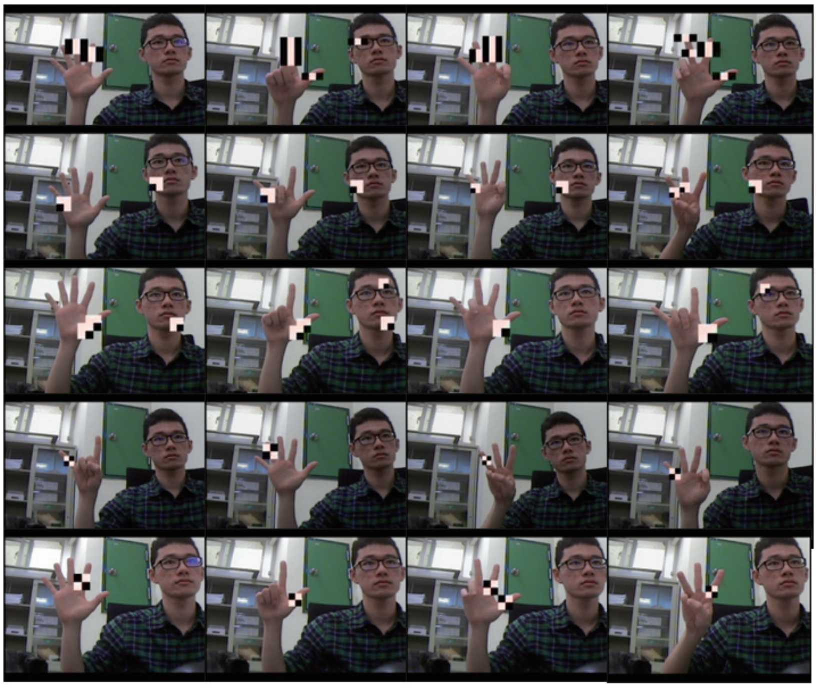


Fig. 10. (Color online) Identification results of the FSP feature matrices of images with complex backgrounds.

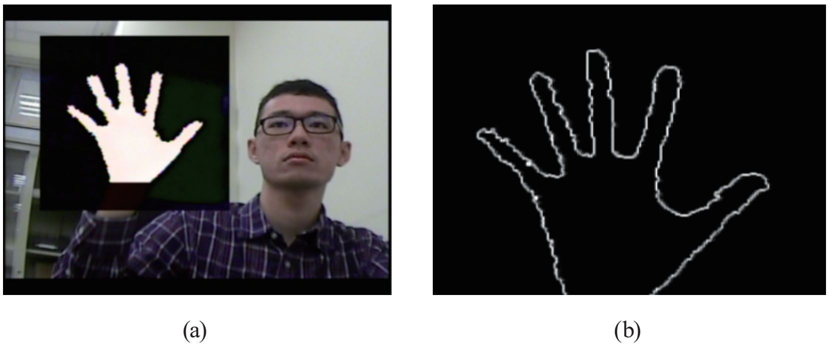


Fig. 11. (Color online) (a) The ROI is limited to inside the rectangle whose sides are 50 pixels from the FSP frame. (b) Use 4-Neighbor to find the hand contour.

process can be reduced by about 65% by adopting ROI. The distance between the hand and the camera is 80 cm in our experiments.

The hand contour is detected as follows. We scan each pixel in the ROI binary image, and set the pixel with a value of zero as the center point of 4-Neighbor. We use 4-Neighbor to examine neighbor pixels for the center point. When one of the four neighbor pixels has a value of 1, it is a boundary pixel. Finally, the hand contour is found by 4-Neighbor, as shown in Fig. 11(b).

4. Hand Gesture Recognition and Gesture Tracing

4.1 Hand gesture recognition

After the hand contour is found, the hand gesture recognition process is as follows. First, we define the operation regions for finger identification, and the thumb and pinkie are recognized on the basis of significant shape features. Second, a circle is defined from the hand center for threshold decomposition to count the number of fingers. Then, hand gestures are recognized.

4.1.1 Thumb and pinkie

First, we remove the wrist part whose width is less than 75 pixels. Then, we draw two rectangles with 30 pixel width from the left and right edges of the frame, as shown in Fig. 12.

The operation regions are the yellow and red boxes in Fig. 12. The skin color proportion in the operation regions is calculated to determine whether a thumb or a pinkie exists using the shape parameter method.(4) When the area of skin color pixels in the operation region is smaller than 13% in our experiments, a thumb or a pinkie exists in it. Otherwise, when the area of skin color pixels is larger than 13%, the hand gesture does not have a thumb or a pinkie. The identification method can quickly and effectively determine the existence of the thumb and pinkie, and improves recognition accuracy and variability.

4.1.2 Center point

To reduce the amount of calculation, the gesture center point $C(x_c, y_c)$ is found using the frame proportion. The y-coordinate y_c is set at the 25% height of the frame, and the x-coordinate x_c is chosen by the following three methods. i) When the hand gesture has a thumb without a pinkie, x_c is set at the 40% width from the left edge of the frame. ii) When the hand gesture has a pinkie without a thumb, x_c is set at the 60% width. iii) If the hand gesture has no thumb and no pinkie, or has both a thumb and a pinkie, x_c is set at the half-width. For example, the gesture center point C shown in Fig. 12 is the center point of the hand gesture with all fingers extended, including the thumb and pinkie.

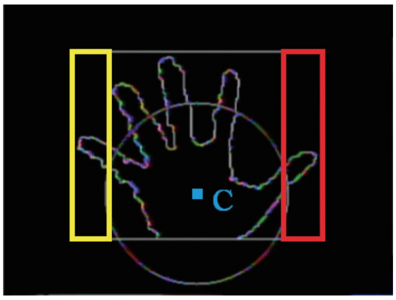


Fig. 12. (Color online) Identification ranges of the thumb and pinkie.

4.1.3 Threshold decomposition

A contour circle is defined as a circle with the radius β and a center at the gesture center point $C(x_c, y_c)$. The radius β is calculated by dividing the length L of the long side of the hand frame by the threshold α. The values of the threshold α in the range of 2.0 to 2.5 have high identification rates in our experiments, and the identification rate is the highest when α is equal to 2.3. Therefore, we use the α value of 2.3 to determine the radius β, i.e., $\beta = L/2.3$, and then we draw the contour circle as shown in Fig. 12. The arbitrary point (x_b, y_b) on the contour circle can be calculated using

$$\beta = \sqrt{(x_b - x_c)^2 + (y_b - y_c)^2}. \tag{2}$$

We calculate the number of crossover points of the contour circle and the hand contour for counting fingers. Hand gestures are recognized on the basis of the number of fingers and the significant shapes of the thumb and the pinkie.

4.2 Gesture tracing

The calculation of the hand gesture center point $C(x_c, y_c)$ was described in Sect. 4.1.2. The points $C_1(x_{c1}, y_{c1})$ and $C_2(x_{c2}, y_{c2})$ are the gesture center points of the first frame and last frame for one hand gesture, respectively. The slope m is defined as Eq. (3). The angle θ between the line $\overline{C_1C_2}$ and the X-axis is defined as Eq. (4). The coordinate differences Δx and Δy between $C_1(x_{c1}, y_{c1})$ and $C_2(x_{c2}, y_{c2})$ are defined as Eq. (5).

$$m = \frac{y_{c2} - y_{c1}}{x_{c2} - x_{c1}} \tag{3}$$

$$\theta = \tan^{-1} m \tag{4}$$

$$\Delta x = x_{c2} - x_{c1}, \Delta y = y_{c2} - y_{c1} \tag{5}$$

The hand trace is calculated using the coordinate differences Δx and Δy. When Δx is positive, then the hand trace direction is toward the right; otherwise, it is toward the left. When Δy is positive, the hand trace direction is downward; otherwise, it is upward. We define the hand trace directions as right, upward, left, and downward in the white area in Fig. 13. The black areas with θ_b are confusion areas with no effect on hand tracking, where

$$\theta - \frac{\pi}{12} < \theta_b < \theta + \frac{\pi}{12}, \; \theta = \frac{(2n+1)\pi}{4}, \; n = 0, 1, 2, 3. \tag{6}$$

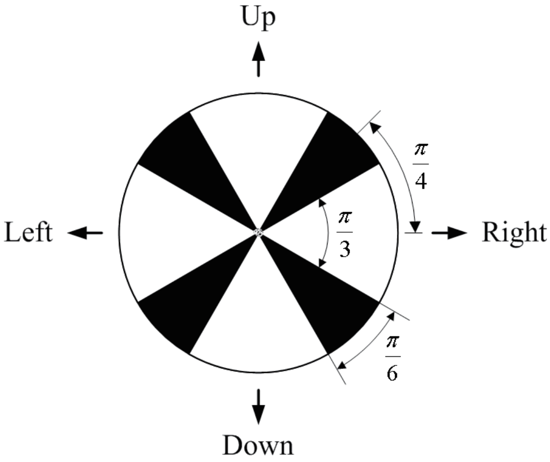


Fig. 13. Identification range of moving direction of one hand.

5. Experimental Results and Application Results

We define hand gestures using the data presented in Sect. 4. First, we use threshold decomposition to obtain the number of fingers, and then determine whether there exists a thumb or a pinkie. We define ten types of hand gestures in our experiments, as shown in Fig. 14. The number at the upper right corner is the recognition result corresponding to the hand gesture in each picture. The hand can be easily distinguished from a complex background and the hand gesture can be recognized with high accuracy. The experiment results of the defined hand trace directions of upward, downward, left, and right are shown in Fig. 15 and the recognition result of the hand trace direction is shown in the upper right corner of each picture. The hand trace direction can be easily traced even for a user with different hand gestures.

Figure 16 shows the confusion matrix of experimental results. The number of test samples for each gesture is 100 from each of ten testers, and the total number of test samples is 1000. The mean recognition rate is 94.3%, and the average recognition time is 0.105 s. Gestures 4, 5, and 8 have 100% recognition rates. For gesture 1, the recognition system has an accuracy rate of 95%. The recognition system mistakes gesture 1 for gesture 2 with a 4%, and for gesture 7 with a 1% misrecognition rate. Some testers show gesture 1 with a relaxed curled thumb; when the curled thumb is calculated as a finger, it becomes gesture 2. When the relaxed curled thumb is calculated as a thumb, then it becomes gesture 7. Gesture 9 consists of holding the ring finger curled while extending the thumb and other fingers. Similarly, gesture 0 consists of holding the ring and middle fingers curled. Some testers find it difficult to curl the ring finger or middle finger without curling the thumb, and it contributes to the low recognition rates of gestures 9 and 0. The hand recognition rates can be improved by training users to match the defined hand gestures.

The DSP DM6437 is used in our hand gesture recognition system. The output of DM6437 is transferred to an infrared remote control circuit through RS-232. The infrared remote controller

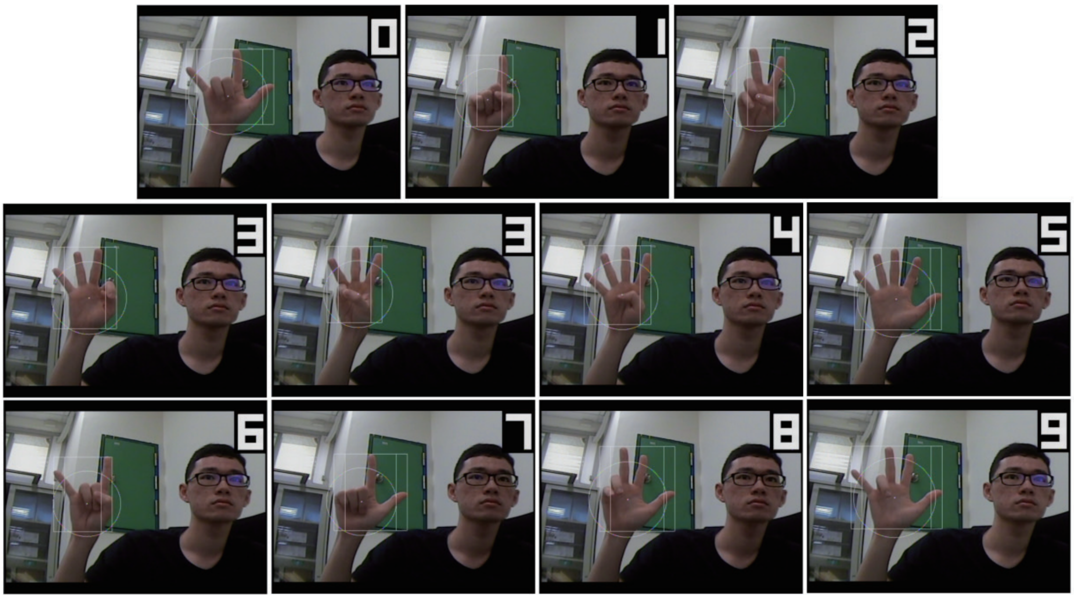


Fig. 14. (Color online) Defined hand gestures in our experiments.

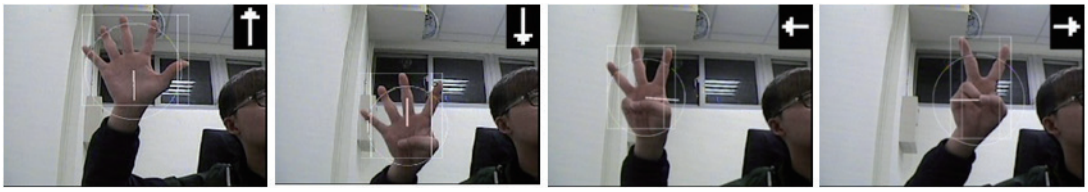
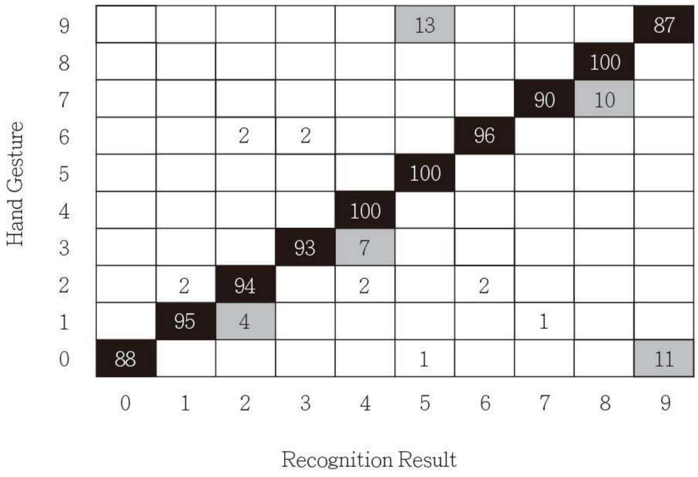


Fig. 15. (Color online) Defined hand tracing directions in our experiments.

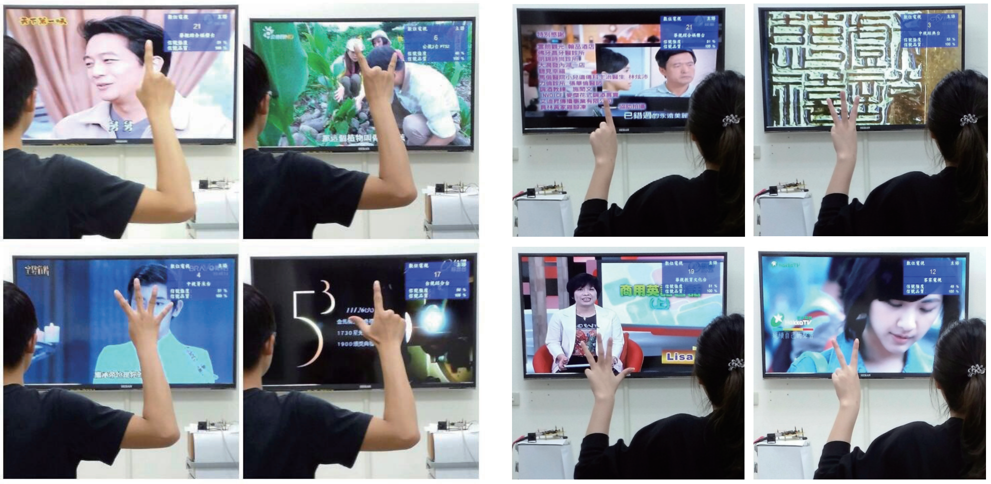


	0	1	2	3	4	5	6	7	8	9
9						13				87
8									100	
7								90	10	
6			2	2			96			
5						100				
4					100					
3				93	7					
2		2	94		2		2			
1		95	4					1		
0	88					1				11

Fig. 16. Confusion matrix of experimental results.

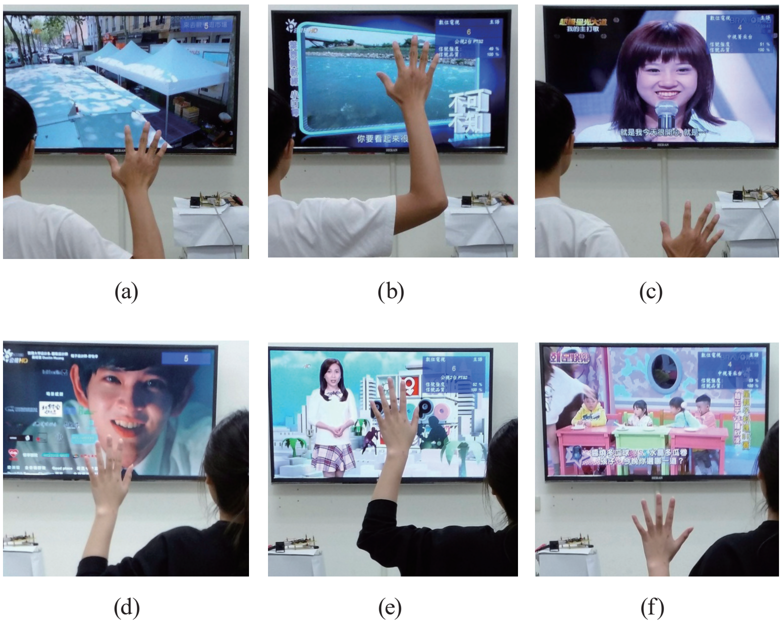
designed with 89SC51 and SC51P0304 circuits is used to control the TV functions. The CCD camera and the DM6437 EVM are set on a table, as shown at the bottom right of the images in Fig. 17(a). The CCD camera captures images, and the images are transferred to DM6437 to identify the hand gestures. To achieve channel and volume control, the infrared commands are created by the DM6437 in accordance with the predefined gestures, and are emitted through the 89C51 and SC51P0304 circuits.

Figure 17 shows the experimental results of TV channel control using hand gestures. Figure 17(a) shows a user gesturing with his right hand to control the TV channel, and Fig. 17(b) shows a user gesturing her left hand to control the TV channel. The TV channel number is shown at the upper right corner of the TV. The user changes the TV channels using only hand gestures, which is more user-friendly than a remote controller. The TV channel and volume can also be easily controlled by hand tracing in our experiments, as shown in Figs. 18 and 19. One user controls the TV channel by right-hand tracing, as shown in Figs. 18(a)–18(c), and another user controls the TV channel by left-hand tracing, as shown in Figs. 18(d)–18(f). Figures 18(a) and 18(d) show the original TV channel; the user increases the TV channel number by moving his/her hand up, and decreases it by moving his/her hand down, as shown in Figs. 18(b) and 18(e), and Figs. 18(c) and 18(f), respectively. TV volume is decreased when the user moves his/her hand to the left, as shown in Figs. 19(a) and 19(c), and increased when he/her moves his/her hand to the right, as shown in Figs. 19(b) and 19(d). A user can easily change the TV channel and volume through hand gestures and hand tracing.



(a) (b)

Fig. 17. (Color online) Results of TV channel control experiment using hand gestures.


(a) (b) (c)

(d) (e) (f)

Fig. 18. (Color online) Results of TV channel control experiment using hand tracing.

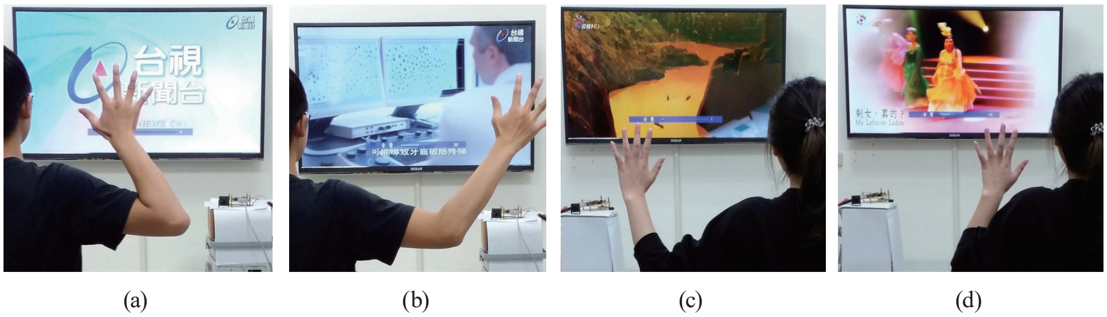

(a) (b) (c) (d)

Fig. 19. (Color online) Results of TV volume control experiment using hand tracing.

6. Conclusions

The real-time hand gesture recognition system based on DSP and its application is described in our research. We propose an FSP algorithm to quickly and easily distinguish the hand in a complex image. The ROI is used to reduce the amount of computation. To detect hand features, we use significant shape features to recognize a thumb or a pinkie. Then a contour circle is defined from the hand contour center for threshold decomposition to count the number of fingers. Our hand gesture sensing and recognition system has a high recognition rate and detects the hand trace direction easily.

The results of our research show that our hand gesture recognition system can conveniently control the TV channel and volume through hand gestures and hand tracing. It will be handy when the remote control is lost and will also increase the amount of exercise for a person. Our research successfully establishes a good HCI that can be applied in other situations, such as a medical environment, where the hand cannot touch equipment.

References

1 J. Wachs, M. Kolsch, H. Stern, and Y. Edan: Commun. ACM **54** (2011) 60.

2 S. Jin, Y. Li, G.M.Lu, J. X. Luo, W. D. Chen, and X. X. Zheng: Symp. VR Innovation (IEEE, 2011) 317.

3 Z. Ren, J. Meng, and J. Yuan: Proc. 8th Information, Communications and Signal (IEEE, 2011) 1.

4 M. Panwar: Conf. Computing, Communication and Applications (IEEE, 2012) 1.

5 X. Bai and L. J. Latecki: IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 1282.

6 Y. C. Kuo, C. T. Tsai, and C. H. Chang: Sens. Mater. **29** (2017) 713.

7 Z. Ren, J. Yuan, J. Meng, and Z. Zhang: IEEE Trans. Multimedia **15** (2013) 1110.

8 Q. Fei, X. Li, T. Wang, X. Zhang, and G. Liu: Proc. WRI Global Congr. Intelligent Systems (IEEE 2009) 139.

9 X. T. Nguyen, L. H. P. Nguyen, T. T. Bui, and H. T. Huynh: Symp. Signal Processing and Information Technology (IEEE 2012) 286.

10 S. Nikhil, M. Saima, B. Ramya, and R. K. Govind: Conf. Signal and Image Processing (IEEE 2010) 39.

11 Texas Instr.: Digital Media Processor DM6437 Datasheet, http://www.ti.com/lit/ds/symlink/tms320dm6437.pdf (accessed April 2018).

12 V. Chikane and C. S. Fuh: J. Inf. Sci. Eng. **22** (2006) 497.

13 Z. H. Al-Tairi, R. W. Rahmat, M. L. Saripan, and S. P. Sulaiman: J. Inf. Process. Syst. **10** (2014) 283.

14 S. Y. Wan and W. E. Higgins: IEEE Trans. Image Process. **12** (2003) 1007.