

High-performance Gesture Recognition System

Yun Wu,¹ Shaoyong Yu,² and Mei Yang^{2*}

¹School of Software Engineering, Xiamen University of Technology,
No. 600, Ligong Road, Jimei District, Xiamen City 361024, China

²School of Mathematics and Information Engineering, Longyan University,
No. 1 Dongxiaobei Road, Xinluo District, Longyan City, Fujian Province 364012, China

(Received October 15, 2018; accepted January 30, 2019)

Keywords: gesture recognition, down sampling method, 2×2 average filter method, discrete wavelet transform, symmetric discrete wavelet transform

With advances in technology, motion sensing has been applied in a wide range of devices. The system used for motion sensing captures images via a camera and analyzes the acquired images, also known as human–computer interaction, wherein gesture-controlled machines are considered as the most convenient kind and are most commonly used. However, the existing gesture recognition algorithm often has a long computation time and uses a huge amount of memory, preventing the algorithm from being implemented in embedded systems. To alleviate these problems, in this paper, we propose to reduce the computation time of the system by employing the lifting-based discrete wavelet transform (DWT). The different frequency bands featured in a lifting-based discrete wavelet can swiftly distinguish the face region from the hand region, reduce the image resolution, and thus reduce memory usage. In addition to recognizing gestures in a swift and accurate manner, the gesture recognition approach proposed in this study is also compatible with embedded systems. Our experimental results suggest that the proposed approach can reduce the execution time by up to 66% while achieving a high identification rate.

1. Introduction

In the rapid development of science and technology, human–computer interaction is playing an essential role. Game consoles such as PS3 and Wii can all be directly operated via faces and gesture recognition, thus enhancing the enjoyment of games. Skin color detection is often applied in gesture recognition systems to roughly determine the position of humans, because skin color is regarded as one of the most distinguishing features of humans. Although the skin color of different races varies considerably, the difference mainly lies in the brightness rather than in the hue,⁽¹⁾ and skin color is still a unique feature that can best distinguish humans from other objects. For skin color detection, many approaches are available; for example, the RGB color space can be converted into the YCbCr color space to take advantage of the clustering effect of skin color particles in the YCbCr space, thereby capturing objects that fall under the category of skin color for further analysis. Alternatively, the skin color can be filtered out

*Corresponding author: e-mail: wulv07@163.com
<https://doi.org/10.18494/SAM.2019.2164>

by converting the RGB color space to the hue, saturation, value (HSV) color space and using the features of this space.⁽²⁾ Inevitably, in a real-life environment, there are many skin-color-like objects, which impose certain difficulties in detecting skin color accurately. To this end, background subtraction is often utilized to detect foreground patches or depth images, so as to distinguish the foreground from the background,⁽³⁾ which can lead to the accurate detection of skin color blocks. Other relatively viable and mature gesture recognition methods include extracting gestures by particle diffusion grouping,⁽⁴⁾ the use of a hidden Markov model, hand shape description using high-dimensional feature vectors,⁽⁵⁾ as well as hand recognition using a Haar feature coupled with a HOG feature.⁽⁶⁾ To distinguish the two regions and analyze hand blocks in particular, it is necessary to find the differences between hands and faces. Haar-like features are widely used to filter out facial features.⁽⁷⁾ By cutting images into rectangular blocks of various sizes and directions, Haar-like features can be used to calculate the sum of block brightnesses by applying image integration while carrying out a comparison between interblock brightnesses difference and the precalculated threshold value. However, the above-mentioned Haar feature detection is a weak classifier, making it necessary to establish a feature database of the target of interest in advance. With due practice, we can capture a good number of stronger features, which can be applied to form a cascaded classifier, which will significantly increase the detection accuracy. However, it may not be able to provide real-time operation owing to the large number of features that must be compared.

Owing to the advances in technology, the pixels of images have been increasingly refined. For skin color detection, CodeBook background modeling, Haar feature detection, and so forth, the computation time increases geometrically with the image size. To overcome this problem, in this paper, we propose the use of the discrete wavelet transform (DWT), a method capable of retaining a large amount of energy, to reduce the image resolution. Also, the characteristic features obtained from the DWT can be used to quickly distinguish hand blocks from face blocks in an image. That is, we can reduce the amount of follow-up system operation by separating hand and face blocks. In addition, we propose a recognition method based on gesture appearance, which has a high calculation speed and a high recognition rate of over 90%. In this paper, we expound on the proposed method of using the DWT to rapidly distinguish hand and face regions. We also describe the process of hand identification and the system flow in detail.

2. DWT for Gesture Recognition

The approach used for extracting gestures is of essential importance in gesture recognition. However, in addition to gestures, the system must also deal with the user's face. In this regard, the characteristics of the DWT are applied to distinguish the face from the hands to reduce the follow-up processing time while analyzing the hand regions and extracting gestures. Figure 1 shows the flow chart of our system. First, DWT encoding is carried out on the input image, thereby reducing the resolution; the subsequent processing involving skin color detection and background modeling is conducted with the reduced image, decreasing the operation time of the system. Then, the face and hand textures are analyzed using the high-low frequency

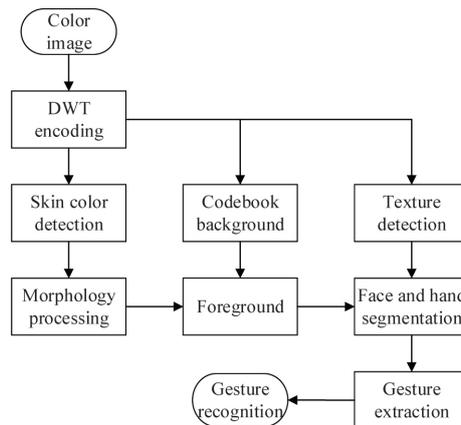


Fig. 1. Flow chart of system.

information produced from the DWT, thereby distinguishing hand and face regions by the difference between their textures.

2.1 DWT encoding

To speed up image processing, image reduction is commonly used to reduce the follow-up processing time. Three methods are commonly used for this purpose, i.e., the down sampling method, the 22 average filter method, and the DWT. All of them can be used to reduce the resolution and computational burden, where the DWT is most widely chosen for its excellent energy concentration ability and multiresolution, which can separately process different elements and largely retain original images upon reducing the resolution. However, the conversion process can be time-consuming.

In this study, we applied an upgraded version of the DWT, i.e., the lifting-based two-dimensional DWT,⁽⁸⁾ also known as the symmetric DWT (SMDWT), which can produce the same image quality and values as the 5/3 lifting-based distortion-free discrete wavelet converter. In the case of two-dimensional multi order operation, SMDWT appears to be convenient. In addition, this new method features a short critical path, fast calculation, regular signal processing, independent sub-band image processing, and so forth. Figure 2(a) shows the coefficients of a symmetric low-frequency shielding matrix, Fig. 2(b) shows the coefficients of a symmetric low-high-frequency mask matrix, and Fig. 2(c) shows the coefficients of a symmetric high-low-frequency shielding matrix used in SMDWT. Degraded images obtained after processing by SMDWT are shown in Fig. 3, where Fig. 3(a) shows low-frequency information, Fig. 3(b) shows low-high-frequency information, i.e., the vertical texture, and Fig. 3(c) shows high-low-frequency information, i.e., the horizontal texture.

Skin color is a very distinct feature for humans and is useful for distinguishing between human bodies and other objects. Many methods can be used to detect skin color. According to the literature,⁽¹⁾ the essential difference between the skin colors of oriental and Caucasian people lies in the brightness. On this basis, when brightness is removed from an image, particle swarm optimization (PSO) for the skin color of various humans has a high clustering tendency, which

1/64	-1/32	-3/32	-1/32	1/64
-1/32	1/16	3/16	1/16	-1/32
-3/32	3/16	9/16	3/16	-3/32
-1/32	1/16	3/16	1/16	-1/32
1/64	-1/32	-3/32	-1/32	1/64

(a)

1/16	-1/8	1/16
-1/8	1/4	-1/8
-3/8	3/4	-3/8
-1/8	1/4	-1/8
1/16	-1/8	1/16

(b)

1/16	-1/8	-3/8	-1/8	1/16
-1/8	1/4	3/4	1/4	-1/8
1/16	-1/8	-3/8	-1/8	1/16

(c)

Fig. 2. (a) Coefficients of symmetric low-frequency shielding matrix, (b) coefficients of symmetric low-high-frequency mask matrix, and (c) coefficients of symmetric high-low-frequency shielding matrix.

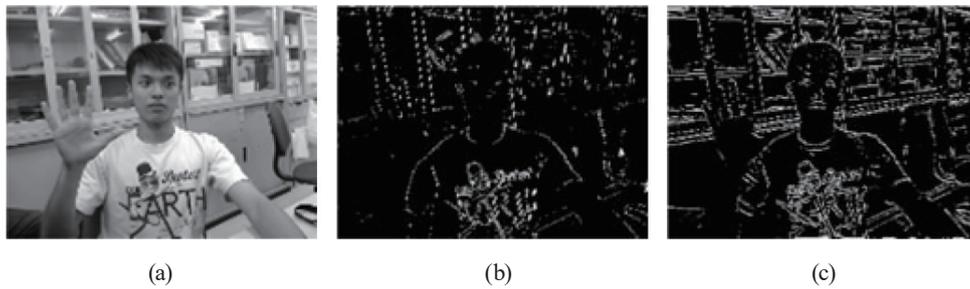


Fig. 3. (a) Low-frequency image, (b) low-high-frequency image, and (c) high-low-frequency image.

can achieve a rather high recognition rate. Thus, in the proposed method, the HVS color space is utilized to detect skin color blocks.⁽²⁾ The HSV color space divides the original RGB color space into three dimensions: hue (H), saturation (S), and value (V). By converting the original RGB color space to the HSV color space, the element with the greatest effect, brightness (V), will be filtered out, and the skin color is identified from the corresponding values of H and S given as

$$Skin(x,y) = \begin{cases} 1, & \text{if } 0 \leq H \leq 80 \\ 0, & \text{if } 0.1 \leq S \leq 0.75 \end{cases} \quad (1)$$

Figure 4(b) shows the skin color blocks of the image in Fig. 4(a) detected via Eq. (1), from which it can be perceived that, although the representative feature of the human body is the skin color, there are many skin-color-like objects in a real-life environment; such noise may undermine our experimental results. Therefore, the detected skin color blocks are fine-tuned as described in the following sections.

2.2 Morphology filtering processing—erosion and dilation

A morphological operation refers to the integration of an image and structural elements in a set operation to produce a new filtered image. With the dilation and contraction in the



Fig. 4. (Color online) (a) Input image and (b) skin color block detected using Eq. (1).

morphology method, noise can be filtered out and unmatched objects can be connected. The basic structural elements used in the morphological operation are shown in Figs. 5(a) and 5(b). The noise is filtered out by using the eight neighbors of a pixel in this method.

2.3 Codebook background

To capture the foreground, a background model is needed for reference. Codebook background modeling is adopted to set up the background model in the proposed method. In contrast to the previous background construction methods, the Codebook algorithm handles the background models in complex scenes by quantization clustering, and every pixel is quantized and represented by Codebook on the basis of the independent pixels in an image sequence. In this algorithm, the color and brightness of each pixel are sampled and then compared with the color distance and brightness in the color model, thereby determining whether the pixel is part of the background. If the answer is positive, the pixel is quantized into groups of Codeword. The background features are stored in the units of pixels; the Codebook of all pixels pieces together a complete background. Unlike the Gaussian mixture model, which requires the probability distribution to be calculated, Codebook only requires the color distance and brightness range of a pixel. Therefore, Codebook has low complexity and low memory usage. Also, in the initialization process, Codebook can swiftly extract the background of an image with moving foreground objects. As an adaptive background update and compression method, it is capable of handling local or global illumination changes, which is essential for real-time image processing.

2.4 Foreground

In this step, to filter out a background block with a color similar to human skin, after the foreground part is separated by the Codebook algorithm (see Sect. 2.4 of the 2.4 measure), the AND operation is performed on the foreground color and the skin color block is extracted by the 2.3 measure. The foreground skin color block is obtained, and the result is connected to the domain to reduce noise interference (as shown in Fig. 6).

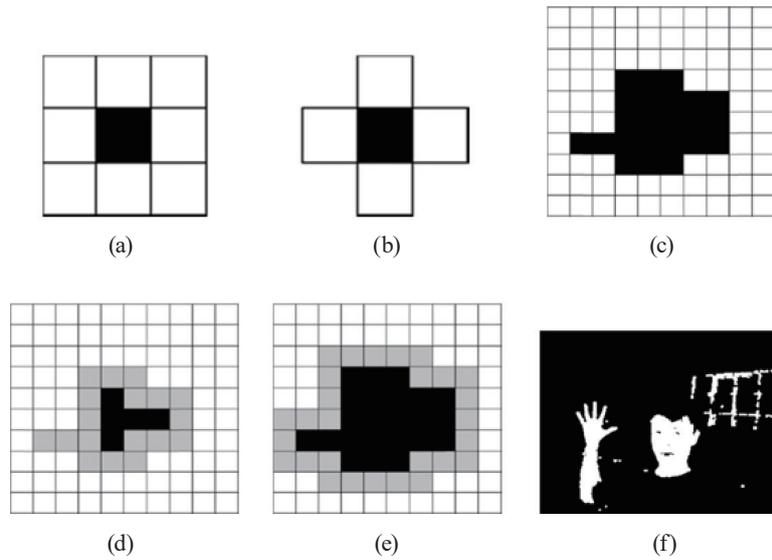


Fig. 5. (a) Eight neighbors of a pixel, (b) four neighbors of a pixel, (c) binarized image, (d) image eroded by eight neighbors, (e) image dilated by eight neighbors, and (f) dilated and contracted image.

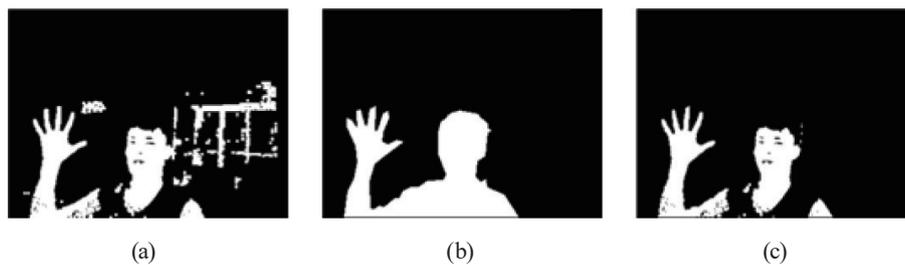


Fig. 6. (a) HSV skin color detection, (b) Codebook foreground, and (c) result of combining (a) and (b).

2.5 Features of DWT

The experimental results suggest that, when the edge texture is not taken into account, the texture features of a human face and palm are considerably different. Since the horizontal texture of a human face contains rich information, we calculate the horizontal texture strength of connected blocks as described in this section. The low- and high-low-frequency information can be obtained via the masking process of Figs. 2(b) and 2(c) in Sect. 2.1. From past experience, we found that the low- and high-low-frequency information represent the vertical and horizontal textures, respectively. In the case of facial textures, the horizontal texture contains a large amount of information, while the vertical texture contains a small amount of information. In this regard, the following processing focuses on analyzing the horizontal texture. To remove the interference caused by weak texture information, we first apply Eq. (3) to check the high-and-low-frequency information transformed by the wavelet transform. If the

value of $HL(x, y)$ is less than the threshold $Th0$, then the horizontal energy intensity is relatively low and can be ignored; if not, $HL(x, y)$ is expressed by $HL(x, y) = 255$. On a grayscale image, we further confirm the approach where a horizontal Sobel filter is used to mask G [Eq. (2)] by $HL(x, y) = 255$, as shown in Eq. (4). If the value convolved via the masked G is greater than the set threshold $Th1$, then $S(x, y) = 1$ is considered to be a texture point. Because the horizontal texture is in a continuous form rather than individual points, if any point satisfying $S(x, y) = 1$ meets the above conditions, then $Label(x, y) = 1$ is marked as a point with a strong horizontal texture, as shown in Eq. (5). This approach can phase-remove many independent points, resulting in more accurate horizontal texture information.

$$G = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

$$HL(x, y) = \begin{cases} 255, & \text{if } HL(x, y) \geq Th0 \\ 0, & \text{if } HL(x, y) < Th0 \end{cases} \quad Th0 = 10 \quad (3)$$

$$S(x, y) = \begin{cases} 1, & \text{if } G * g(x, y) \geq Th1 \ \& \ HL(x, y) = 255 \\ 0, & \text{if } G * g(x, y) < Th1 \ \& \ HL(x, y) = 255 \\ \text{where } Th1 = 100 \end{cases} \quad (4)$$

$$Label(x, y) = \begin{cases} 1, & S(x, y+1) = 1 \ | \ S(x, y-1) = 1 \\ 0, & \text{else} \end{cases} \quad (5)$$

$g(x, y)$ is a grayscale image, $HL(x, y)$ is the pixel value of the original image processed by the symmetric high-low-frequency mask matrix, $S(x, y)$ is the marker of a texture point, and $Label(x, y)$ is the marker of a strong texture point.

2.6 Segmentation of face and hands

The previous section suggested that a palm, which is rather smooth, contains less texture information than a human face. According to experimental results, it can be perceived that the texture characteristics of a human face and hand vary considerably. On this basis, we have been able to distinguish human hands and faces by building up statistics regarding the number of strong texture points in each connected block and examining the difference in the number of texture points, thus increasing the recognition rate. On the basis of the sizes of various blocks of facial skin color and their corresponding strengths for the consistency of texture points, we established a look-up table as a threshold reference. Note that light will alter the texture information to a certain extent. Therefore, our discussion only applies to experimental settings with a stable light source. When the number of texture points exceeds a set threshold,

the texture is regarded as a facial texture, as shown in the green frame in Fig. 7(b); if not, it is regarded as a hand texture, as shown in the red frame in Fig. 7(b).

The threshold table set up in this experiment is shown in Fig. 7(a). It can be perceived that the shorter the distance to the camera, the larger the number of strong texture points in the face. In addition, the high- and low-frequency information produced by the DWT may be lost owing to the movement of the object. Therefore, it is challenging to recognize face blocks in the case of fast movement. To overcome this problem, given that the movement of an object is continuous and uninterrupted, when a block has been determined to be in a facial region, the system carries out coordinate positioning and tracking, in which the block is given a fixed coordinate. Subsequently, any block within a certain range of the set coordinate is automatically regarded as a face block.

2.7 Gesture extraction

In Sect. 2.6, the face and hand were distinguished by the difference in the number of strong texture points. Therefore, instead of processing the entire image, only the blocks determined as hand blocks will be read as the region of interest and then processed, to remove a number of system operations in Fig. 8.



Fig. 7. (a) Graph of number of strong texture points and their distance to camera and (b) segmentation result of hand and face.

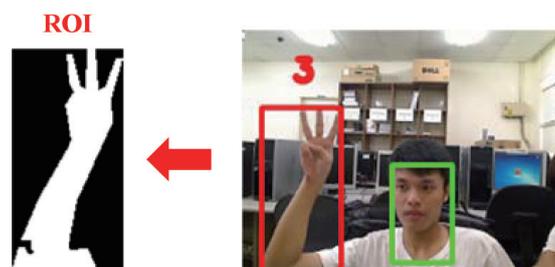


Fig. 8. (Color online) Region of interest.

3. Gesture Recognition

In terms of gesture recognition, the input image may be skewed, which is not conducive to the system's recognition process. Therefore, we first rotate the gestures, so as to closely analyze their features. We identify the wrist and the joints of the fingers and palm, removing the parts other than the palm. The feature data obtained by this analysis is utilized to distinguish the fingers and identify the gesture. The gesture recognition flow is shown in Fig. 9, where the four processes are described in the following subsections.

3.1 Rotate

Under normal circumstances, users' gestures are very likely to be skewed. It is necessary to rotate the gestures in this step to facilitate the subsequent processing. Prior to rotating the arm, we need to find out the skew angle of the arm. For this purpose, least-squares estimation⁽⁹⁾ is adopted here, in which a regression line is obtained by minimizing the sum of the squares of all the distances of the discrete points from the regression line. The rotation angle and slope can be obtained from the regression line. Next, the rotated image can be produced by introducing the rotation angle into Eq. (6), as shown in Fig. 10.

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (6)$$

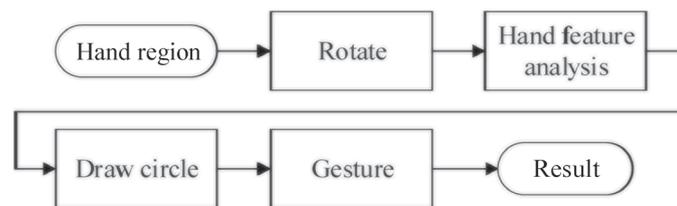


Fig. 9 Gesture recognition flowchart.



Fig. 10 (Color online) Rotated Image.

3.2 Hand feature analysis

Gestures are mostly made by placing the palm and fingers together, so the skin color blocks below the wrist are disposable. To scale down the complexity of the operation, we removed the part below the wrist, while the palm remained unchanged.⁽¹⁰⁾ Figures 11(a) and 11(b) show schematic diagrams of a human palm. It can be seen from Fig. 11(a) that the palm region is almost square, making it easier for us to distinguish the palm from the hand. We next consider the fingers. It can be seen from Fig. 11(a) that the lengths of the fingers and that of the palm are similar, with the ratio approximately ranging between 1 and 1.4. Therefore, to remove the part below the palm, we must first figure out the finger length in Fig. 11(b). For this purpose, we carry out a scan down from a finger as shown in Fig. 11(c). Since the width in the lateral direction increases markedly upon reaching the palm, we apply this feature to determine the position of the base of the fingers and the finger length. The base of the fingers is shown by the second horizontal line in Fig. 11(d). We observed that the ratio of the finger length to the wrist length is about 1–1.4. Therefore, with the base of the fingers as a reference, we determined the start of the wrist as a distance of 1.2 times the finger length below the base of the fingers, as shown by the third line in Fig. 11(d).

3.3 Draw circle

We improved the method in Ref. 10 as reposted in this section. Considering that the method proposed in this paper is affected by changes in radius, we derived a method of stabilizing

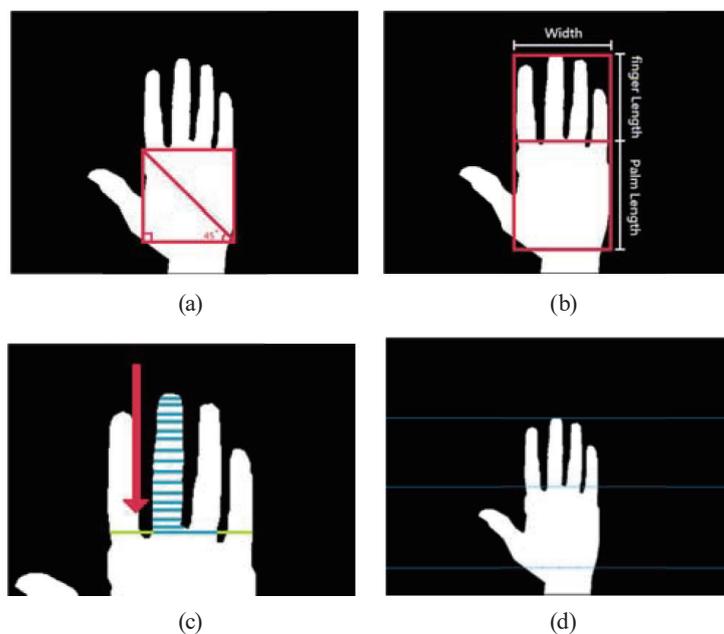


Fig. 11 (Color online) (a) Hand features, (b) hand features, (c) diagram for finding ends of fingers, and (d) hand segmentation diagram.

the radius. As reported in the previous section, we have already obtained the hand blocks to be removed, the width of the wrist, the center wrist, and so forth. As shown in the schematic diagram in Fig. 12(b), we found that we can distinguish the palm by drawing a circle with the middle point of the wrist as the center and the palm length as the radius. Next, by increasing the radius to 1.35 times the palm length, the fingers can be separated as shown by the blue line in Fig. 12(c); however, there may be certain errors when it comes to the thumb. To reduce the error, we draw semicircles with the points “1/4 wrist width” from the right and left of the palm center as an attempt to more accurately identify the thumb, as shown by the red line in Fig. 12(c).

3.4 Gesture

We first use the number of fingers to distinguish the gestures. A gesture with one finger is certainly gesture 1; a gesture with two fingers may be gesture 2, 6, or 7; a gesture with three fingers may indicate gesture 3 or 8; a gesture with four fingers may be gesture 4 or 9; and a gesture with five fingers must be gesture 5. Thus, only gestures with two, three, and four fingers require further identification. In Fig. 12(b), the orange blocks represent the extracted fingers. The coordinates of the highest point of each orange block are recorded; then the highest and lowest points among these coordinates are found. In each diagram in Fig. 13, the yellow line is no higher than the apex of the semicircle, and the blue line passing through the center point of the wrist divides the diagram into left and right sides. Figure 13(a) shows a schematic diagram of gesture 2. It can be seen that the highest and lowest points are both above the yellow line. Figure 13(b) is gesture 6, whose lowest point is below the yellow line, and its highest and lowest points are to the left and right of the center of the palm, respectively, that is, they lie on opposite sides of the blue line [Fig. 13(c)], i.e., gesture 7, has its lowest point below the yellow line, and the highest and lowest points are on the same side of the blue line. A gesture with three fingers may be gesture 3 or 8. The highest and lowest points of gesture 3 are above the yellow line, as shown in Fig. 13(d), whereas for gesture 8, as shown in Fig. 13(e), the highest and lowest points are on opposite sides of the yellow line. Finally, a gesture with four fingers may be gesture 4 or 9. As shown in Fig. 13(f), the highest and lowest points of gesture 4 are above the yellow line, whereas the highest and lowest points of gesture 9 are on opposite sides of the

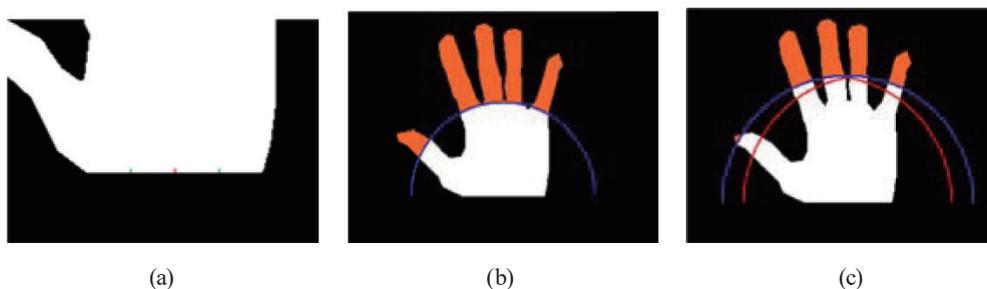


Fig. 12. (Color online) Schematic diagrams of (a) the center of the wrist, (b) the palm with a circle, and (c) the palm with semicircles.

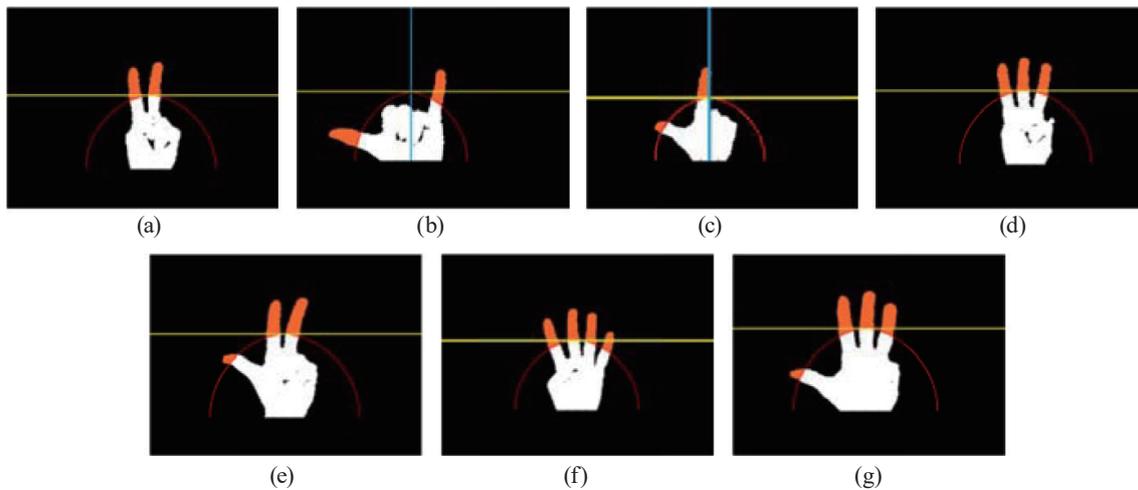


Fig. 13. (Color online) Diagrams of the relative positions of gestures.

yellow line [Fig. 13(g)]. By connecting the various domain markers, the coordinates of the top ends of the fingers can be obtained; such coordinates which are used to compare between the highest and lowest points and the information acquired from the previous step, including the wrist center and radii. Taking into account the relative positions and relevant information, we can recognize gestures, regardless of the distance, so as to avoid possible errors caused by different distances to the lens.

4. Experimental Results

In this work, the computer used for the simulation had an Intel Core i5-2410 CPU with 2.3 GHz working frequency and 4 Gb RAM. We implemented our method using Microsoft Visual Studio 2008 on a Windows platform.

A Logitech web camera with an image resolution of 640×480 was used to input the test images. The face and hand recognition rates were obtained after testing 500 images; we employed 9000 test images to find the accuracy rate for gestures 1–9, which are shown in Fig. 14, and the experimental results can be found in Tables 1 and 2. Table 1 shows that face detection has a relatively high accuracy rate, where the identification failures that occasionally occurred were mostly caused by the sides of faces with a very large turning angle, while faces within a certain range of angles can be easily detected. From Tables 2 and 3, it can be perceived that the gesture recognition system has a high speed and a high recognition rate of over 90% for gestures 1–9. Examples of unsuccessful samples are shown in Figs. 15–17, where several factors are considered to have caused identification failures, such as the nonpositive gesture shown in Fig. 15, the fingers too widely apart or too close, leading to errors when detecting finger tips, as shown in Fig. 16, and a disfigured hand shape due to skin color detection, as shown in Fig. 17. Table 3 gives the execution speeds of the various sections of the system; the average speed of the overall system is roughly 30 fps, which can satisfy real-time requirements.

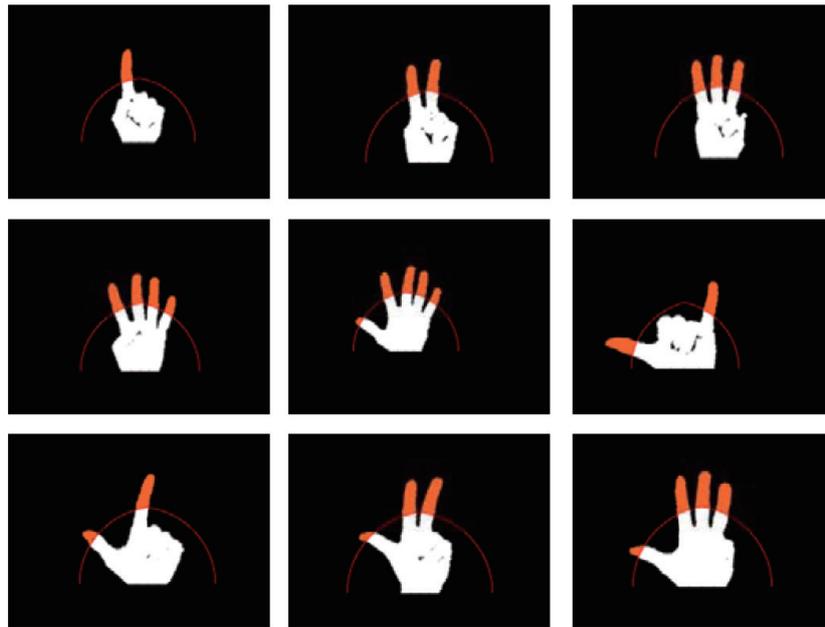


Fig. 14. (Color online) Examples of successful samples.

Table 1
Hand recognition results.

Gesture	Total no. of samples	Successful samples	Unsuccessful samples	Success rate (%)
1	1000	983	17	98.3
2	1000	991	9	99.1
3	1000	992	8	99.2
4	1000	971	29	97.1
5	1000	973	27	97.3
6	1000	961	39	96.1
7	1000	947	53	94.7
8	1000	980	20	98.0
9	1000	981	19	98.1

Table 2
Face recognition results.

Total no. of samples	Successful samples	Unsuccessful samples	Success rate (%)
500	481	19	96.2

Table 3
System execution times.

WT for face recognition			Gesture recognition system execution times		
	Execution time (ms)	fps		Execution time (ms)	fps
DWT transform	14.11209	70.86	Rotate	1.60219	624.15
Skin detect and morphology	3.51457	284.53	Hand feature analysis	0.38615	2589.67
Foreground	3.72797	268.24	Draw circle	0.92350	1082.84
Texture and face detect	2.44290	409.35	Gesture	0.48920	2044.15
Total	23.79753	42.02	Total	3.40104	294.03
			Total system		
			Execution time (ms)		fps
			29.61921		33.76

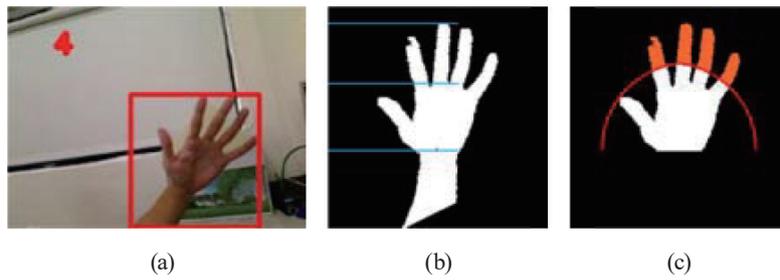


Fig. 15. (Color online) (a) Tested gesture 5, (b) finger segmentation diagram, and (c) result diagram.

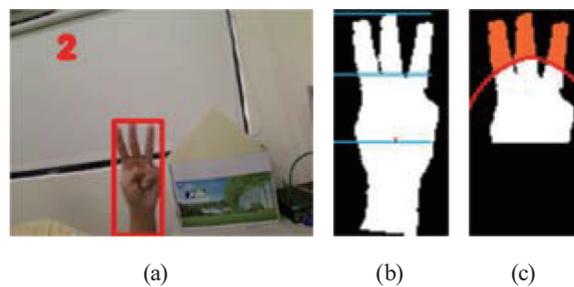


Fig. 16. (Color online) (a) Tested gesture 3, (b) finger segmentation diagram, and (c) result diagram.

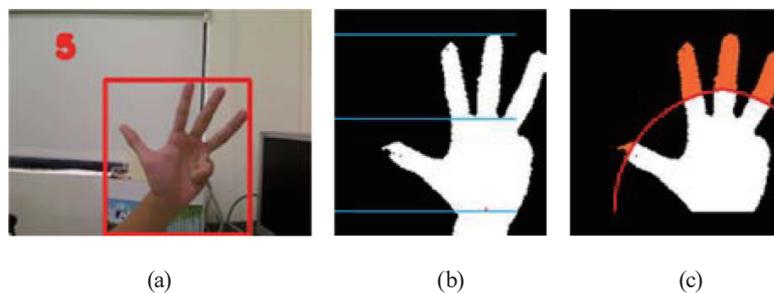


Fig. 17. (Color online) (a) Tested gesture 9, (b) finger segmentation diagram, and (c) result diagram.

5. Conclusion

We proposed the use of the symmetric DWT for gesture recognition to distinguish hand blocks from face blocks in an efficient and stable manner. The experimental results verified that the proposed method can significantly reduce the amount of computation involved in the system. In addition, by taking advantage of the features of the DWT, the hands and face can be distinguished much faster than by Haar feature detection, where the latter requires feature matching. Also, the face and hand blocks can be identified rather accurately with the DWT. For hand recognition, we proposed a simple and stable method to divide the palm from the fingers, whose recognition rate was over 90% in experiments. Because no complex sensing components are used in our method and the symmetric DWT used in our system is notably simpler than the original DWT, hardware-based development is promising, following further research and development.

Acknowledgments

This work was supported by the National Science Foundation of Fujian Province of China (No. 2018J01575) and the Project of Scientific Climbing Plan by Xiamen University of Technology (No. XPDKQ18015).

References

- 1 J. Yang and A. Waibel: Proc. 3rd Workshop Applications of Computer Vision (1996) 142–147.
- 2 G. Zhao, S. Wang, T. Wang, and J. Chen: Proc. 2nd Int. Conf. Future Generation Communication and Networking Symposia (2008) 107–110.
- 3 Y. Wen, C. Hu, G. Yu, and C. Wang: Proc. IEEE Int. Workshop. Haptic Audio Visual Environ. Games (2012) 72–77.
- 4 K.-J. Hsiao, T.-W. Chen, and S.-Y Chien: IEEE ICME (2008) 977–980.
- 5 A. Wilkowski: Proc. Conf. Human System Interactions (2008) 737–742.
- 6 S. Ghafouri and H. Seyedarabi: Proc. 21st Iranian Conf. Electrical Engineering (ICEE) (2013) 1–4.
- 7 E. ul Haqi, S. J. H. Pirzada, M. W. Baig, and H. Shin: Proc. IEEE 54th Int. Midwest Symp. Circuits and Systems (MWSCAS) (2011) 1–4.
- 8 C.-H. Hsia. IEEE Transa. Circuits Sys. Video Technol. **19** (2009) 1201.
- 9 D. W. Marquardt: Soci. Ind. Appl. Math. **11** (1963) 431.
- 10 J. C. Tsai: Oriental Institute of Technology (2012).

About the Authors



Yun Wu received her Ph.D. from Xiamen University in 2007. Her research interests lie in the areas of artificial intelligence and big data. Her scientific contribution to AI has more to do with soft computing and clustering algorithms. (ywu@xmut.edu.cn)



Shaoyong Yu received his Ph.D. from Xiamen University in 2017. He is also an associate professor in the School of Mathematics and Information Engineering, Longyan University. His research interests lie in the areas of computer vision and deep learning. (82018007@lyun.edu.cn)



Mei Yang is a lecturer and teacher in the School of Mathematics and Information Engineering, Longyan University. Her research interests lie in the areas of software development and artificial intelligence. (wulv07@163.com)