International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.8 (2015), pp.287-302
http://dx.doi.org/10.14257/ijmue.2015.10.8.29

A Real-Time Hand Detection System during Hand over Face Occlusion

Jun Xu and Xiong Zhang

Display Centre, School of Electronic Science and Engineering, Southeast University, China
xujun0730@gmail.com, zhangxiongseu@gmail.com

Abstract

This paper devises a vision-based hand detection system which can handle the situation of the hand and face occlusion. Firstly, human face is coarsely located by skin color detection and ellipse template matching and further determined by using the grayscales and positions of human eyes. Secondly, when the hand occludes the partial region of the face, a novel hand detection method based on Camshift tracking algorithm, force field method and Sobel edge extraction is developed to segment the hand. Finally, the positions of segmented hands are sent to the computer for controlling the cursor movement. In order to reduce the cursor jitter caused by wrong hand detection and generate a smooth trajectory of the cursor, the coordinates of hand positions are corrected by combining the least squares fitting method of orthogonal polynomial and an adaptive Catmull-Rom interpolation algorithm. Experimental results showed that the proposed method could detect hands accurately with the run time of 0.08s per frame and demonstrated a significant improvement in performance when the hand enters the face region. Moreover, our system accomplished smooth movements of the cursor by the vision-based hand detection.

Keywords: *Human-computer interaction, hand detection, complex backgrounds, occlusion, curve fitting*

1. Introduction

Hand gesture recognition is one of the primary topics in human-computer interaction. In the early study of this technique, gesture recognition was realized by using data gloves [1] or color markers on the fingertips [2]. Because those devices made the usage inconvenient and unnatural, most of recent works focus on the vision-based gesture recognition with naked hands. However, the difficulty is how to segment and track the hands in complex backgrounds, such as illumination variation, face interference, movements of other objects and so on.

Human body can be distinguished from other objects according to the color, so skin color is widely used to detect hands [3, 4]. However, the color-based methods may fail when the background contains other objects with the color of skin. Hence some color-independent features are employed, such as edges [5], contours [6] and textures [7]. In order to improve the recognition accuracy under complex backgrounds, these features are usually combined to locate hands. For example, Brasnett, P. *et al* [8] developed a weighted scheme which combined color, edge and texture cues to track hands. M. Gonzalez *et al*. [9] proposed a hand detection method based on skin color and edge orientation. The face was located by a specific skin model and registered as a face template. During the occlusion of face and hand, the object was determined as a hand by comparing the local gradient orientations of the object with that of the face template. In [10], a Malaysian sign language system based on skin segmentation and feature extraction was proposed. The centroids of head and hands were estimated by using the Kalman filter

ISSN: 1975-0080 IJMUE

and then their overlap region was located.

The features mentioned above are related to the appearance of objects. Moreover, some appearance-independent features have also shown good performance in hand gesture detection. For example, Chen-Chiung, H. *et al*. [11] developed a hand gesture system which employed an adaptive skin color model and motion history images. M. Kristan *et al*. [12] proposed a hand recognition system which could deal with the situations of hand occlusion and overlap. A local motion model was calculated from the optical flow of the target and a tracker based on color and motion was constructed by modifying a particle filter. In [13], a novel representation of image, called force field, was proposed to detect hand-over-face gestures. The detection rate reached 97%. Furthermore, P. Smith *et al*. [14] employed force field to model the regional structural changes of the hand images. Then a mixture model of Gaussians was used to segment hands during the hand-face occlusion.

Additionally, some cameras which can monitor depth information are used in some gesture interaction systems. In [15], the ambiguities caused by the overlap of hands were solved using skin color and depth data captured from a Time-of-Flight camera. In [16], different regions of skin color were divided by the depth data and an Expectation-Maximization algorithm. Since the hand was regarded as the closet skin-color object to the camera, it could be successfully extracted under overlap.

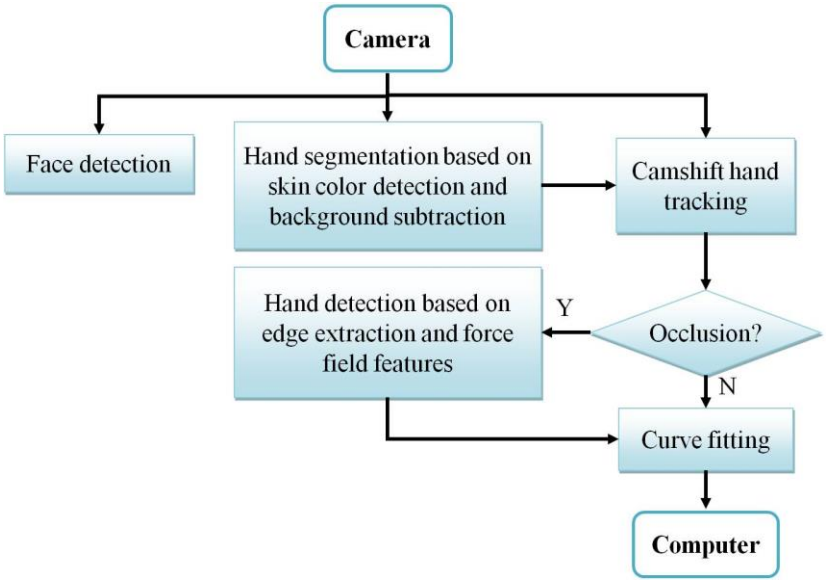


Figure 1. System Architecture

Since ordinary cameras with low cost are widely used in various devices, this paper proposed a gesture detection system based on a single ordinary camera which only provides 2D information. To deal with the poor discrimination of hands when the hand overlaps with the face, the system includes 5 modules as shown in Figure1. Firstly, face is detected by ellipse matching and features of human eyes. Once a face is located, the face region is registered and the hand detection mechanism is activated. Secondly, hands are detected by skin color segmentation and background subtraction and tracked by the Camshift algorithm [17]. When the hand approaches or enters the face region, a detector which combines edge extraction and force field features is used to segment the hand. Finally, the positions of the detected hand are sent to the computer and the cursor moves according to the trajectory of the hands. Because slight shake of hands or wrong detection of hand positions would lead to a rough movement of the cursor and even a significant skip, a curve fitting method combining polynomial least squares fitting and Catmull-Rom interpolation is proposed to smooth the trajectory of hand movement.

2. Face Detection

Face detection is a procedure to determine whether there are any faces in an image and, if present, provide the scale and location information of each face. Face detection methods can be classified into the knowledge-based method, the template matching method and the statistics-based method. Firstly, the knowledge-based method utilizes features of the face, such as geometrical shape, color, texture and so on. For example, R. Brunelli and T. Poggio [18] detected the face border and the nose by the integral projection of the edge image in the vertical direction, while located the mouth and eyes by the integral projection in the horizontal direction. The knowledge-based method is sensitive to the status variation of the face, so it is generally applied in the detection of frontal faces in the simple background. Secondly, the template matching method establishes face templates to describe facial characteristics and calculates the correlation values between the object and the templates. In [19], the face detection was achieved by template matching based on a linear transformation. Compared with the knowledge-based method, the accuracy of the template matching method is higher, but the operation speed is slower. Finally, in the statistics-based method, numerous face and non-face samples are trained to generate a classifier for distinguishing faces, such as eigenface method [20], artificial neural networks (ANN) [21], support vector machine (SVM) [22] and Adaboost method [23]. If learning samples are sufficient and comprehensive, the accuracy of the statistics-based method is superior to that of the other two methods, so how to choose typical samples becomes one of the critical steps.

In this paper, a face detection approach with a combination of skin color, features of human eyes and ellipse template matching is presented. Firstly, the input image is converted into a binary image using skin color detection. Then the connected domains with large areas are extracted to match the face templates. In order to accelerate the matching, two groups of ellipse templates with different sizes are built. As shown in Figure2, ellipse contour templates are employed to fit the connected domain. If the matching correlation is high, the domain is further checked by ellipse templates. Finally, the positions and grayscale characteristics of human eyes are used to estimate the candidates obtained from the step of ellipse matching. As shown in Figure3, assuming h is the height of the pattern, the horizontal projection of a face is searched above the line L_h at $3/8h$ until the first minimal point Y_e is found. The pattern in the range of Y_e-L to Y_e+L is projected in the vertical direction, where L is one-tenth width of the pattern. All the minimal points of the vertical projection are labeled as $\{X_{ei}|i=1,\ldots,n\}$. Consequently, all the points $\{P_i(X_{ei},Y_e)|i=1,\ldots,n\}$ where eyes would appear are obtained. For each point P_i, two square regions with the areas of $L\times L$ and the centers of $C1(X_{ei},Y_e)$ and $C2(X_{ei},Y_e+1.5L)$ are measured to calculate the average gray values, represented as G1 and G2 (Figure4). If any of the following conditions are met, P_i will be removed.

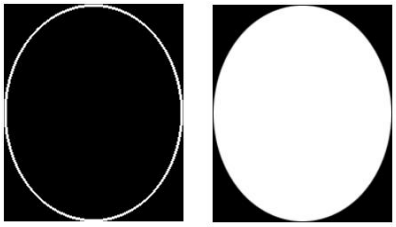
1. $G1>G2$

2. $G1 \le G2$ & $\frac{G2-G1}{G2}<20\%$

The remaining points are pairwise grouped to identify whether their positions meet the distribution features of human eyes as follows. Let *Xel* and *Xer* be the horizontal ordinates of the left and right point. Hence *Xer*-*Xel* is the distance between the two eyes, *Xel* is the distance between the left eye and its nearest face boundary, and *w*-*Xer* is the distance between the right eye and its nearest face boundary.

1. $Xer\text{-}Xel > \frac{1}{3}w$

2. $Xel \ge L$ & $w\text{-}Xer \ge L$



(a) (b)

Figure 2. Two Kinds of Face Template: (a) Ellipse Contour Template and (b) Ellipse Template

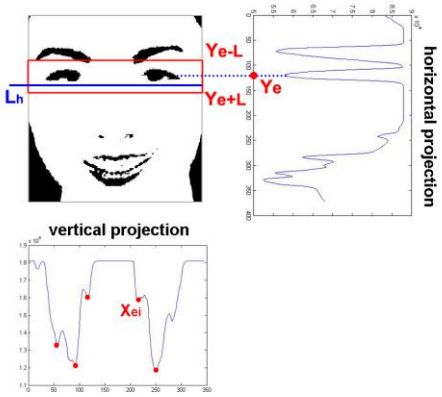


Figure 3. The Horizontal Projection and Vertical Projection of a Face Image

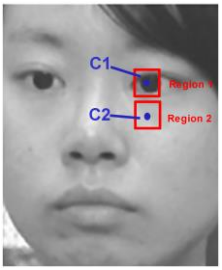


Figure 4. Gray Comparison of Two Square Regions Around the Eye

3. Hand Detection

Once a human face is detected, the mechanism of hand detection is activated. Skin color detection, background subtraction and edge extraction are widely used to segment hands from the background. However, these pixel-based methods always fail when the hand overlaps with the face because color changes slightly in the occlusion region. For example, the extracted contour of the hand is incomplete as shown in Figure 11. Nevertheless, there is a significant variation in the regional feature of the image, such as force field proposed in [14]. In this paper, an efficient method which combines Camshift tracking algorithm, force field and edge detection is proposed to esitimate hands during the occlusion.

3.1 Force Field

Work in [14] applies the concept of potential energy to image potential. Each pixel in the image is exerted forces by all the other pixels, so force field forms (Figure5). The

force at location r_j is given by

$$F(\vec{r}_j) = \sum_{\vec{r}_i \neq \vec{r}_j} I(\vec{r}_i) \frac{\vec{r}_i - \vec{r}_j}{\left|\vec{r}_i - \vec{r}_j\right|^3} \tag{1}$$

If unit test charges $t_1,\ldots,t_m$ are placed in the force field, they will travel as

$$\begin{aligned} t_{i,j} &= t_{i,j-1} + \bar{F}(t_{i,j-1}) \\ \bar{F}(t_{i,j}) &= \frac{F(t_{i,j})}{\left|F(t_{i,j})\right|} \end{aligned} \tag{2}$$

Where $t_{i,j}$ is the location of the test charge t_i at time j and $\bar{F}(t_{i,j})$ is the normalized vector at $t_{i,j}$. Every test charge t_i finally stabilizes at a potential well $t_{i,N}$ when the stopping criteria Eq.3 is met.

$$\frac{t_{i,j} - t_{i,j-\delta}}{\delta} \leq \frac{1}{2} \tag{3}$$

Figure 6(a) shows the travelling channels and locations of test charges after 100 iterations of Eq.2. The distance between the start and stop location of a test charge is given by

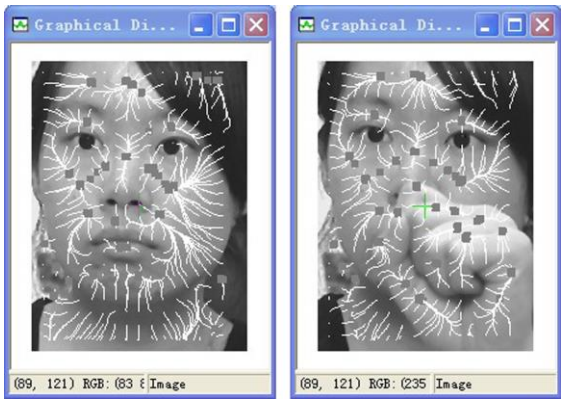
$$d = \left|t_{j,0} - t_{j,N}\right| \tag{4}$$

Once a hand enters the face region, the channels in the region of occlusion change significantly (Figure 6 (b)). Consequently, the travelling distances *d* of test charges vary. In [14], P. Smith *et al*. measured the variation using Gaussian Mixture Model (GMM) to distinguish the hand and the face.

Compared with background subtraction and mean shift tracking, the force field method shows better segmentation performances during the hand-face occlusion. However, although [14] modeled every 5th pixel for faster computation, the run time was still 15s per frame on a P4 laptop. Furthermore, some additional pixels near the occlusion region would be wrongly segmented.

(a) (b)

Figure 5. The Magnitude of the Force Field

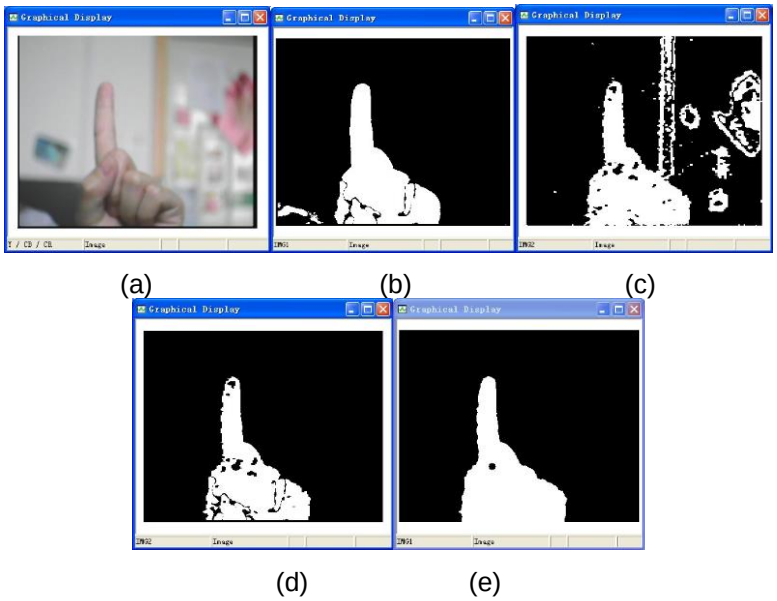


(a) (b)

Figure 6. Travelling Channels and Locations of Test Charges. (a) White Lines and Gray Squares Indicate the Travelling Channels and the Locations of Test Charges after 100 Iterations. (b) Channels Change Significantly During Occlusion

3.1 The Proposed Method of Hand Detection

Assuming that the hand dose not overlap with the face at the beginning, the hand can be initially located by background subtraction and skin color segmentation. As shown in Figure7, the input image is segmented by background subtraction (Figure7b) and skin color detection (Figure7c). The two segmentation results are processed by AND operation (Figure7d) and closing operation (Figure7e) to obtain the hand region. Then the hand region is tracked by the Camshift algorithm.



(a) (b) (c)

(d) (e)

Figure 7. Hand Segmentation. (a) Input Image, (b) Background Subtraction, (c) Skin Color Detection, (d) AND Operation and (e) Closing Operation

Table 1. Run Times Using Different Rectangular Grids of Test Charges and Different Iteration Times

		Iteration times		
		50	**100**	**500**
Place a test charge every n pixels	**n=1**	0.33s	0.45s	0.84s
	n=5	0.09s	0.10s	0.12s
	n=10	0.07s	0.08s	0.08s

When the hand approaches the face region, a novel approach of hand detection is proposed which combines Camshift tracking algorithm, force field method and edge detection. Firstly, the hand position is coarsely estimated by Camshift tracking algorithm. Then in order to reduce the computation cost, only travelling distances of test charges in the tracking window are calculated. As shown in Table 1, the run times of the method in [14] performed on our DSP platform indicate that the run time can be further shorten by reduce iteration times and numbers of test charges. Finally, Sobel edge detection is employed to detect the result of force field to improve the segmentation accuracy. The steps of our method are described in detail as follows.

Step 1: Forces of each pixel in the face region are calculated. For faster computation, the force of the point *k* is obtained by accumulating the points within 5-pixel distance of *k* as Eq.5. This approximation is reasonable because Eq.1 indicates that the forces exerted by the faraway pixels are very small.

$$F_x(k)=\sum_{l=k-5}^{k+5} I_l \times \frac{x_l - x_k}{\left(\sqrt{(x_l-x_k)^2+(y_l-y_k)^2}\right)^3}$$
$$F_y(k)=\sum_{l=k-5}^{k+5} I_l \times \frac{y_l - y_k}{\left(\sqrt{(x_l-x_k)^2+(y_l-y_k)^2}\right)^3} \quad (5)$$

$F(k)=(F_x(k), F_y(k))$ and $\overline{F}(k)=(F_x(k)/|F(k)|, F_y(k)/|F(k)|)$ are the force and the normalized force of *k*.

Step 2: Test charges are placed every 10 pixels. Assuming the test charge *i* is initially at (x_0, y_0), it will travel in the force field and the iteration represented by Eq.6 won't stop until the distance of $k_n(x_n, y_n)$ and $k_{n+1}(x_{n+1}, y_{n+1})$ is less than a threshold.

$$x_{n+1}=x_n+F_x(k_n)$$
$$y_{n+1}=y_n+F_y(k_n) \quad (6)$$

Finally, the test charge *i* will stop at the well (x_N, y_N) and the travelling distance is given by

$$d=\sqrt{(x_N-x_0)^2+(y_N-y_0)^2} \quad (7)$$

All the travelling distances of test charges in the face region are registered as a template set *D*.

$$\{D_i \in D \mid D_i=\sqrt{(x_{i,N}-x_{i,0})^2+(y_{i,N}-y_{i,0})^2}\}$$

Step 3: Once the template above is registered, the module of hand detection is activated. The hand is located by background subtraction and skin color detection.

Step 4: Camshift algorithm is employed to track and monitor the hand.

Step 5: When the hand enters the face region, test charges are uniformly placed in the tracking window as shown in Figure8a. Travelling distances of the test charges are figured out by Eqs.5-7.

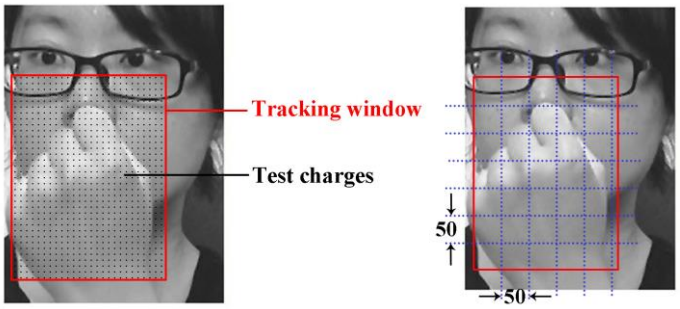
Step 6: The tracking window is divided into several squares of 50×50 pixels as shown in Figure8b. A sum of each square is calculated by

$$S = \sum_{i=1}^{50\times50} \left| d_i - D_i \right| \tag{8}$$

Where d_i and D_i are travelling distances of i in the current frame and in the template. If the maximum of S is larger than the given threshold, the corresponding square is determined as a part of the hand. Moreover, assuming the wells in the square are represented by ($x_{k,N}$, $y_{k,N}$), any test charge i whose well ($x_{i,N}$, $y_{i,N}$) meets the condition as Eq.9 is also determined as a part of the hand.

$$\sqrt{(x_{i,N} - x_{k,N})^2 + (y_{i,N} - y_{k,N})^2} \le \beta,\ k = 1, 2, ..., K \tag{9}$$

Step 7: Edges in the face region are extracted using the Sobel operator. As shown in Figure9, edge pixels are assigned to 0, while other pixels are assigned to 1. Thus a binary edge image is produced. Then the AND operation is performed between the edge image and the result obtained in step 6 to obtain the final result of hand detection.



(a) (b)

Figure 8. (a) Test Charges are Uniformly Placed Every 10 Pixels in the Tracking Window. (b) The Tracking Window is Divided Into Several 50×50 Squares

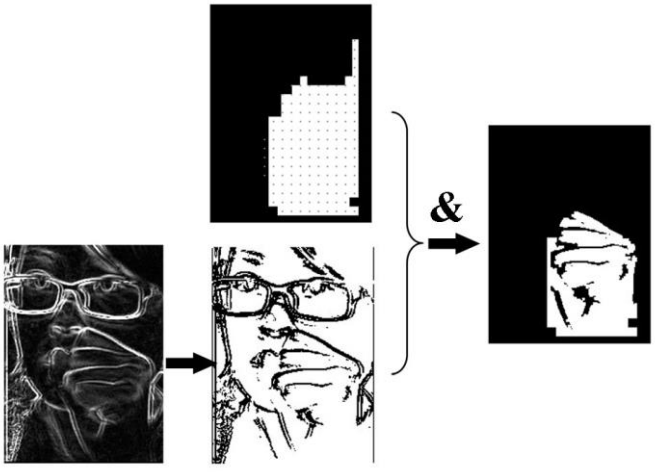


Figure 9. AND Operation Between the Result of Force Field Method and the Edge Image

4. Curve Fitting

The centroids of detected hands are sent to the computer and then the cursor will move with the hands. Hence the dragging function of a mouse is emulated by the bare-hand

gesture recognition. However, slight shake of hands or wrong recognition of hand positions would lead to a rough trajectory of the cursor and even a significant skip. In order to structure a curve which reflects the general trend of the known hand positions and smoothes the trajectory of hand movement, a curve fitting method based on polynomial least squares fitting and Catmull-Rom interpolation is proposed. Polynomial least squares fitting is applied to correct the detection results of hand positions and filter wrong results, and the Catmull-Rom method is adopted to build a trajectory curve of the hand by interpolating several points between every two successive positions of the hand.

4.1 Polynomial Least Squares Fitting

The trend of the hand movement can be predicted by fitting several hand positions in the previous frames. The fitting result is used to correct the hand position in the present frame. In this paper, least squares fitting method of orthogonal polynomial is employed. Given a set of hand centroids $P_i(x_i,y_i)$ (i=0,1,…,N), our task is to generate a second-order orthogonal polynomial p(x) as Eq.10 to approximate these centroids.

$$\begin{aligned} p(x) &= u_0\varphi_0(x) + u_1\varphi_1(x) + u_2\varphi_2(x) \\ &= u_0 + u_1(x-a_0) + u_2[(x-a_1)(x-a_0) - b_0] \end{aligned} \tag{10}$$

The unknown coefficients can be figured out from several observations $\{(x_i, y_i)\}_{i=0\ldots n}$ according to Eqs.11-13. These observations are the coordinates of hand centriods either directly obtained by hand detection or corrected by least squares fitting. In this paper, the coordinates of P_i are calculated by the previous three points (P_{i-1}, P_{i-2}, P_{i-3}) after correcting and the next point P_{i+1} before correcting.

$$\begin{cases} \varphi_0(x) = 1 \\ \varphi_1(x) = x - a_0 \\ \varphi_{k+1}(x) = (x-a_k)\varphi_k(x) - b_{k-1}\varphi_{k-1}(x),\ k = 1,2,...,n-1 \end{cases} \tag{11}$$

$$\begin{cases} a_k = \dfrac{(x\varphi_k, \varphi_k)}{(\varphi_k, \varphi_k)},\ k = 0,1,...,n-1 \\ b_{k-1} = \dfrac{(\varphi_k, \varphi_k)}{(\varphi_{k-1}, \varphi_{k-1})},\ k = 1,2,...,n-1 \end{cases} \tag{12}$$

$$u_k = \frac{(y, \varphi_k)}{(\varphi_k, \varphi_k)},\ k = 0,1,...,n \tag{13}$$

4.2 Catmull-Rom Interpolation

A piecewise smooth curve, called Catmull-Rom curve, is applied to draw the trajectory of the hands [24]. As shown in Figure10, N-1 points $P_i(x_i,y_i)$ (i=1,2,…,N-1) are interpolated between the second point P_2 and the third point P_3 using the coordinates of four points (Eq.14). (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4) are the coordinates of P_1, P_2, P_3 and P_4 respectively.

$$\begin{cases} x_i = a_i x_1 + b_i x_2 + c_i x_3 + d_i x_4 \\ y_i = a_i y_1 + b_i y_2 + c_i y_3 + d_i y_4 \end{cases} \tag{14}$$

$$\begin{cases} a_i = \frac{1}{2}(-t^3 + 2t^2 - t) \\ b_i = \frac{1}{2}(3t^3 - 5t^2 + 2) \\ c_i = \frac{1}{2}(-3t^3 + 4t^2 + t) \\ d_i = \frac{1}{2}(t^3 - t^2) \end{cases}, \quad t = \frac{i}{N}$$

The number of interpolation points *N-1* can be set as an arbitrary number. More interpolation points create a smoother curve, but reduce the computational efficiency. Thus in this paper the number of interpolation points adaptively changes according to the distance *D* between P_2 and P_3, which is represented as Eq.15. The shorter *D* is, the fewer the interpolation points are.

$$N - 1 = \frac{D}{32} \tag{15}$$

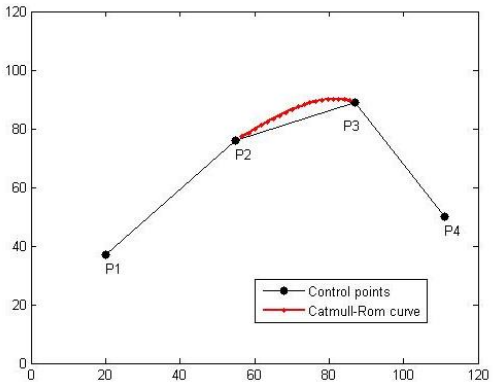


Figure10. Catmull-Rom Interpolation

5. Experimental Results and Analysis

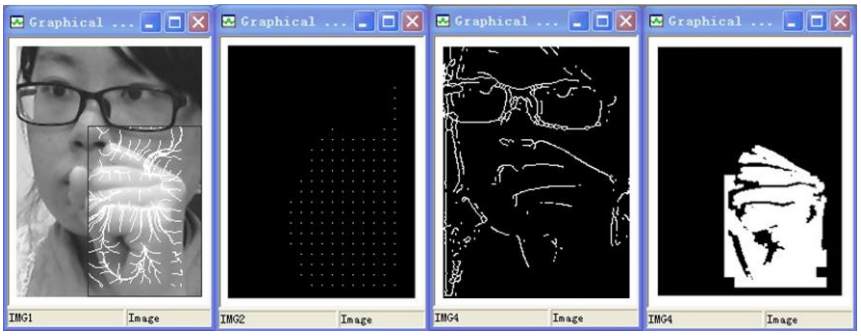
The proposed system was implemented on a DSP platform of TMS320DM643-600MHz. Since the goal of this paper was to realize hand detection in real time, image sequences were captured by a CCD camera at 30 fps with a resolution of 360×288 pixels. The experimental results show that the average run time of our method was 0.08s per frame, which was fast enough for the real-time processing.

Firstly, Figure11 shows the hand detection results of an image sequence during the hand-face occlusion. In the first column, the black rectangles in the luminance images are the tracking results of Camshift algorithm. The second and third columns show the results of force field and edge detection respectively. The binary images in the last column are the final segmentation results. As can be seen from Figure11, our method works well when the fist rotates (Figure11b,c), when the hand leaves or approaches the camera (Figure11f,g) and when the face rotates a small degree(Figure11e). Similarly, Figure12 shows the detection results of other hand postures, such as “paper” (Figure12a) and “point” (Figure12b,c). Moreover, two metrics of average rates is used to evaluate the detection performance as shown in Table 2. The true positive rate is the percentage of hand pixels which are classified as the hand in all hand pixels, and the true negative rate is the percentage of background pixels which are classified as the background in all background pixels. In Table 2, the two rates reach 87.64% and 98.26% which indicate the

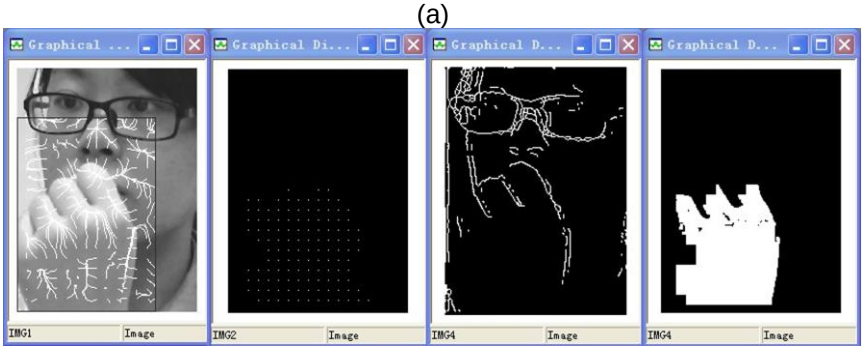
high segmentation accuracy of our method.

Table 2. True Positive Rate and True Negative Rate for the Test Image Sequence

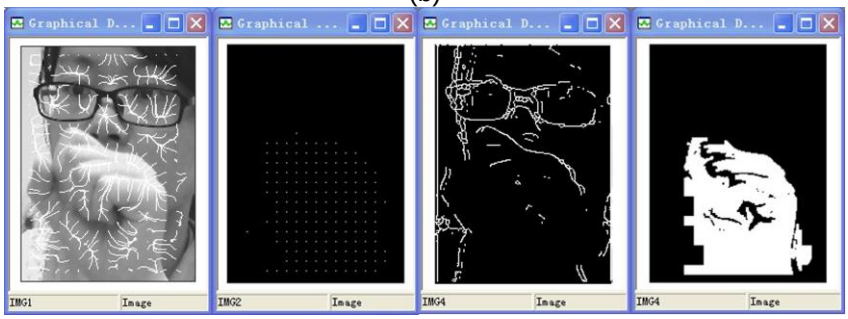
Number of frames	True positive rate	True negative rate
80	87.64%	98.26%



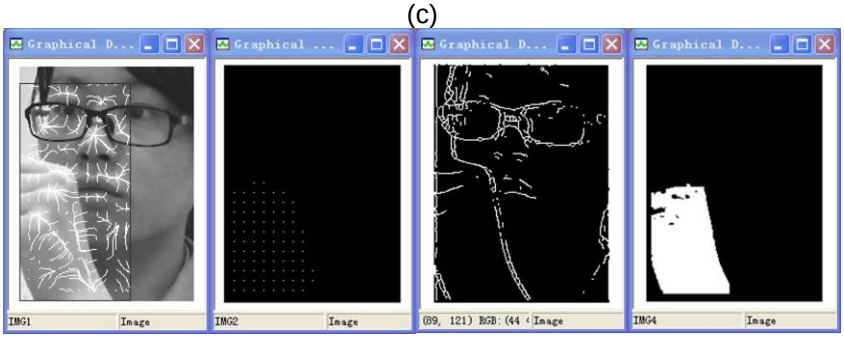
(a)



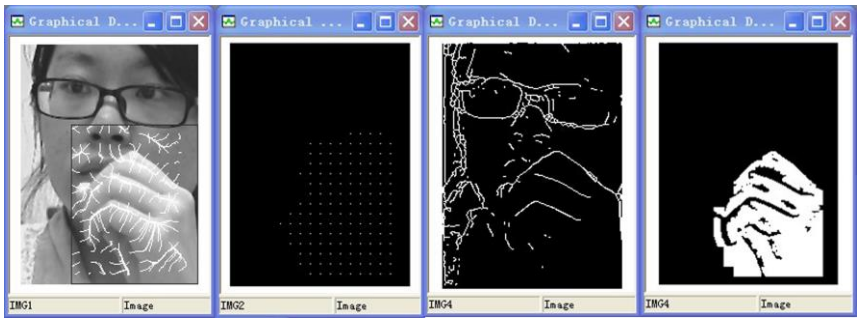
(b)



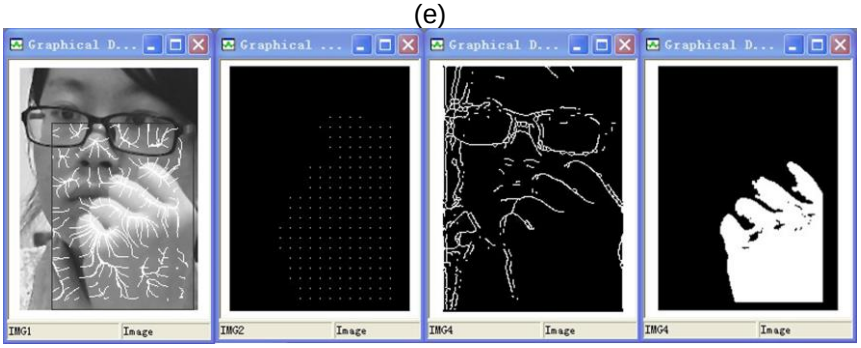
(c)



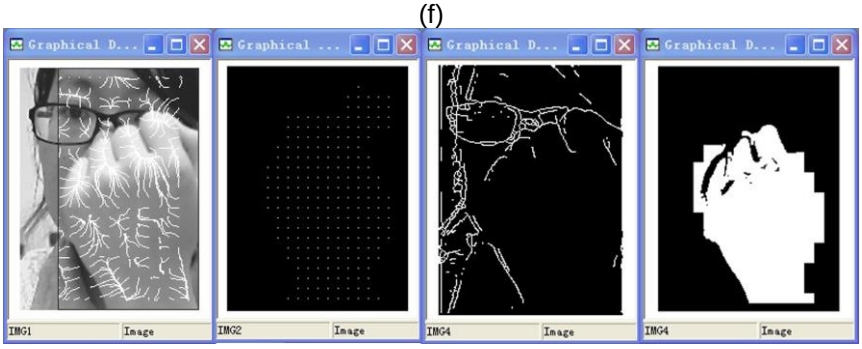
(d)



(e)

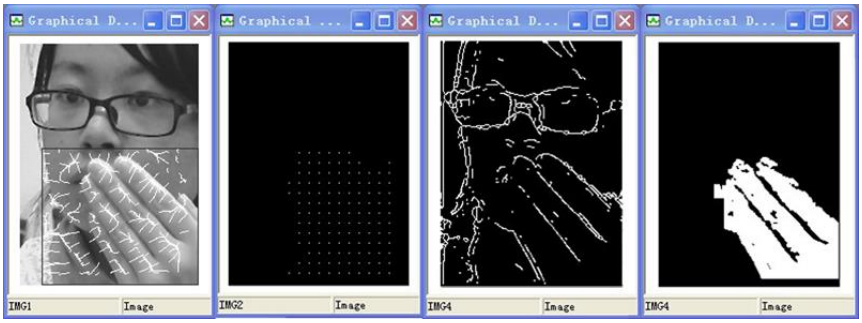


(f)

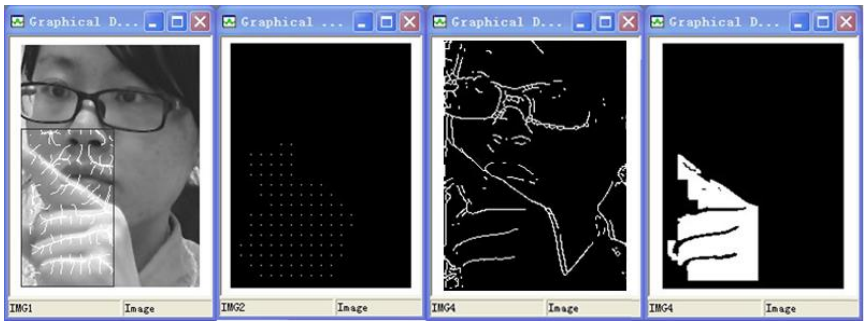


(g)

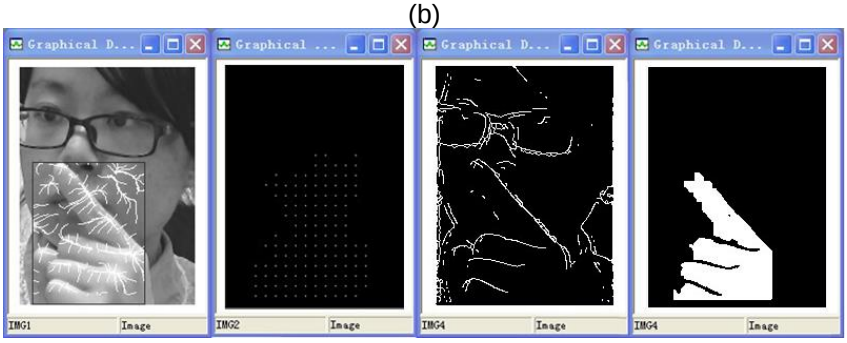
Figure 11. Hand Detection Based on the Proposed Method



(a)



(b)



(c)

Figure 12. Experimental Results of Other Hand Postures: Paper (a) and One Finger (b,c)

Secondly, the performance of curve fitting was test. Figure13a shows a trajectory of the detected hands. The coordinates of these hands are converted to the coordinates of the cursor by

$$\begin{cases} X_{cur} = C \cdot X_{cam} \\ Y_{cur} = C \cdot Y_{cam} \end{cases} \tag{16}$$

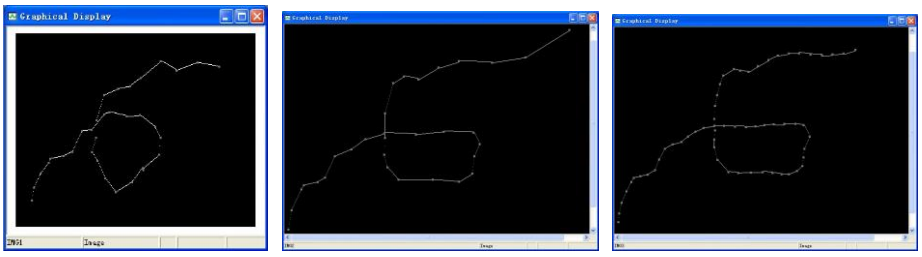
Where (X_{cam}, Y_{cam}) and (X_{cur}, Y_{cur}) are the coordinates in the camera coordinate system and in the cursor coordinate system respectively. In our experiments, C was set as 2.5. Then the coordinates of the cursor were calculated by the proposed curve fitting method. The metric of smoothness is employed to measure the performances. Assuming the angle between the vector $\overrightarrow{p_{i-1}p_i}$ and $\overrightarrow{p_{i-1}p_i}$ is computed by Eq.17, The smoothness s is calculated as the average of all angles (Eq.18). Note that larger s indicates better smoothness.

$$\begin{aligned} \theta_i &= \arccos \frac{\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i p_{i+1}}}{\left|\overrightarrow{p_{i-1}p_i}\right| \cdot \left|\overrightarrow{p_i p_{i+1}}\right|} \\ &= \frac{(x_i - x_{i-1})(x_{i+1} - x_i) + (y_i - y_{i-1})(y_{i+1} - y_i)}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \cdot \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \end{aligned} \qquad i = 1, 2, \ldots N-2 \tag{17}$$

$$s = \frac{1}{N-2} \sum_{i=1}^{N-2} \theta_i \tag{18}$$

Figure13b shows the experimental result of orthogonal polynomial fitting. The smoothness is 0.4553, 36% better than the original trajectory (Figure13a). As shown in Figure13c, the curve of Catmull-Rom interpolation creates the smoothness of 0.2479, 65% better than the original trajectory. Thus it is proved that the proposed method of

curve fitting greatly improves the smoothness of the hand movement.



(a) (b) (c)

Figure13. (a) An Original Trajectory of Hands, (b) Results of Orthogonal Polynomial Fitting and (c) Results of Catmull-Rom Interpolation

6. Conclusions

This paper proposes a hand detection system with a single camera which only provides luminance and chrominance information. The system can segment hands from the face region accurately and control the movement of the cursor by the trajectory of hand positions. Firstly, a face detection method is presented using skin color, human eyes features and ellipse template matching. In order to reduce the computation cost, two groups of ellipse templates are employed to establish a two-layer matching. Then the hand is detected based on skin color cue and background model and tracked by the Camshift algorithm. Once the hand enters the face region, a novel hand detection method is proposed specially for the hand over face occlusion. By combining Camshift method, force field and Sobel edge detection, the hand can be extracted from the face region with high accuracy and high speed. Finally, a curve fitting scheme by using the least squares fitting method of orthogonal polynomial and the Catmull-Rom interpolation is proposed to create a smooth trajectory of the cursor. In the future, we would optimize our system for hand detection of more postures.

Acknowledgements

This work was supported by the National Key Basic Research Program of China (No. 2010CB327705).

References

[1] S. S. Fels and G. E. Hinton, "Glove-Talk: a neural network interface between a data-glove and a speech synthesizer," Neural Networks, IEEE Transactions on, vol. 4,**(1993)**, pp. 2-8.
[2] J. Davis and M. Shah, "Recognizing hand gestures," in: J.-O. Eklundh (Ed.) Computer Vision — ECCV '94, Springer Berlin Heidelberg, **(1994)**, pp. 331-340.
[3] M. Jones and J. Rehg, "Statistical Color Models with Application to Skin Detection," Int J Comput Vision, vol. 46, **(2002)**, pp. 81-96.
[4] M. P. Paulraj, S. Yaacob, M. S. B. Z. Azalan and R. Palaniappan, "A phoneme based sign language recognition system using skin color segmentation," in: Signal Processing and Its Applications (CSPA), 2010 6th International Colloquium on, **(2010)**, pp. 1-5.
[5] R. Lionnie, I. K. Timotius and I. Setyawan, "An analysis of edge detection as a feature extractor in a hand gesture recognition system based on nearest neighbor," in: Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, **(2011)**, pp. 1-4.
[6] S. Nasri, A. Behrad and F. Razzazi, "A novel approach for dynamic hand gesture recognition using contour-based similarity images," Int J Comput Math, vol., **(2014)**, pp. 1-24.
[7] A. Kumar and D. Zhang, "Personal recognition using hand shape and texture," Image Processing, IEEE Transactions on, vol. 15, **(2006)**, pp. 2454-2461.
[8] P. Brasnett, L. Mihaylova, D. Bull and N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences," Image Vision Comput, vol. 25, **(2007)**, pp. 1217-1227.
[9] M. Gonzalez, C. Collet and R. Dubot, "Head Tracking and Hand Segmentation during Hand over Face

Occlusion in Sign Language," in: K. Kutulakos (Ed.) Trends and Topics in Computer Vision, Springer Berlin Heidelberg, **(2012)**, pp. 234-243.
[10] Y. F. A. Gaus and F. Wong, "Hidden Markov Model-Based Gesture Recognition with Overlapping Hand-Head/Hand-Hand Estimated Using Kalman Filter," in: Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on, **(2012)**, pp. 262-267.
[11] H. C. Chiung, L. D. Hua and D. Lee, "A real time hand gesture recognition system using motion history image," in: Signal Processing Systems (ICSPS), 2010 2nd International Conference on, **(2010)**, pp. V2-394-V392-398.
[12] M. Kristan, J. Perš, S. Kovačič and A. Leonardis, "A local-motion-based probabilistic model for visual tracking," Pattern Recogn, vol. 42, **(2009)**, pp. 2160-2168.
[13] M. Mahmoud, R. E. Kaliouby and A. Goneid, "Towards Communicative Face Occlusions: Machine Detection of Hand-over-Face Gestures," in: M. Kamel, A. Campilho (Eds.) Image Analysis and Recognition, Springer Berlin Heidelberg, **(2009)**, pp. 481-490.
[14] P. Smith, N. da Vitoria Lobo and M. Shah, "Resolving hand over face occlusion," Image Vision Comput, vol. 25, **(2007)**, pp. 1432-1448.
[15] M. V. D. Bergh and L. V. Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in: Applications of Computer Vision (WACV), 2011 IEEE Workshop on, **(2011)**, pp. 66-72.
[16] S. Handrich and A. A. Hamadi, "Multi hypotheses based object tracking in HCI environments," in: Image Processing (ICIP), 2012 19th IEEE International Conference on, **(2012)**, pp. 1981-1984.
[17] S. M. Nadgeri, S. D. Sawarkar and A. D. Gawande, "Hand Gesture Recognition Using CAMSHIFT Algorithm," in: Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on, **(2010)**, pp. 37-41.
[18] R. Brunelli and T. Poggio, "Face recognition through geometrical features," in: G. Sandini (Ed.) Computer Vision — ECCV'92, Springer Berlin Heidelberg, **(1992)**, pp. 792-800.
[19] Z. Jin, Z. Lou, J. Yang and Q. Sun, "Face detection using template matching and skin-color information," Neurocomputing, vol. 70, **(2007)**, pp. 794-800.
[20] M. U. Çarıkçı and F. Ö zen, "A Face Recognition System Based on Eigenfaces Method," Procedia Technology, vol. 1, **(2012)**, pp. 118-123.
[21] S. A. Nazeer, N. Omar and M. Khalid, "Face Recognition System using Artificial Neural Networks Approach," in: Signal Processing, Communications and Networking, 2007. ICSCN '07. International Conference on, **(2007)**, pp. 420-425.
[22] K. Hotta, "Robust face recognition under partial occlusion based on support vector machine with local Gaussian summation kernel," Image Vision Comput, vol. 26, **(2008)**, pp. 1490-1498.
[23] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen and Y. Wu, "AdaBoost-based face detection for embedded systems," Comput Vis Image Und, vol. 114, **(2010)**, pp. 1116-1125.
[24] J. Tsao, "Interpolation artifacts in multimodality image registration based on maximization of mutual information," Medical Imaging, IEEE Transactions on, vol. 22, **(2003)**, pp. 854-864.

Authors

Jun Xu, received her bachelor degree in School of Electronic Science and Engineering from Southeast University, China, in 2008. She is currently studying for a PhD in Display Centre of Southeast University. Her research interests include human-computer interaction, pattern recognition and image processing.

Xiong Zhang, received his PhD degree in School of Electronic Science and Engineering from Southeast University, China, in 2001. Currently he is a professor of Electronic Science and Engineering in Southeast University. His research interests include display technology, image processing, electronic material and so on.