A Dynamic Hand Gesture Recognition System Incorporating Orientation-based Linear Extrapolation Predictor and Velocity-assisted Longest Common Subsequence Algorithm

Min Yuan[1], Heng Yao[2], Chuan Qin[3], Ying Tian[4]
Shanghai Key Lab of Modern Optical System, and Engineering Research Center of Optical Instrument and System, Ministry of Education, University of Shanghai for Science and Technology, Shanghai 200093, China
[1][e-mail: yuanmin_2013@126.com]
[2][e-mail : hyao@usst.edu.cn]
[3][e-mail : qin@usst.edu.cn]
[4][e-mail : tianying@usst.edu.cn]
*Corresponding author: Heng Yao

Received June 15, 2016; revised April 19, 2017; accepted May 25, 2017; published September 30, 2017

Abstract

The present paper proposes a novel dynamic system for hand gesture recognition. The approach involved is comprised of three main steps: detection, tracking and recognition. First, the gesture contour captured by a 2D-camera is detected by combining the three-frame difference method and skin-color elliptic boundary model. Then, the trajectory of the hand gesture is extracted via a gesture-tracking algorithm based on an occlusion-direction oriented linear extrapolation predictor, where the gesture coordinate in next frame is predicted by the judgment of current occlusion direction. Finally, to overcome the interference of insignificant trajectory segments, the longest common subsequence (LCS) is employed with the aid of velocity information. Besides, to tackle the subgesture problem, i.e., some gestures may also be a part of others, the most probable gesture category is identified through comparison of the relative LCS length of each gesture, i.e., the proportion between the LCS length and the total length of each template, rather than the length of LCS for each gesture. The gesture dataset for system performance test contains digits ranged from 0 to 9, and experimental results demonstrate the robustness and effectiveness of the proposed approach.

Keywords: Dynamic gesture recognition, adaptive linear extrapolation, longest common subsequence (LCS), velocity information

This work was supported by the National Natural Science Foundation of China (61672354).

https://doi.org/10.3837/tiis.2017.09.017 **ISSN : 1976-7277**

1. Introduction

Recently, dynamic gesture recognition has become one of the hottest research topics in human computer interaction (HCI) and has played an important role in virtual reality, smart home and automatic control. Different from the traditional mouse and keyboard, dynamic gesture recognition technique provides a more natural way for interaction. Furthermore, rather than traditional ways for static gesture recognition, dynamic gesture recognition can offer more semantics and real-time user experiences.

A dynamic gesture recognition system mainly consists of three key techniques: detection, tracking and recognition [1]. The action of gesture detection is to distinguish the hands from the surrounding environment by their characteristics, such as skin-color, outline, motion and texture, etc. Due to the interference of illumination, deformation and complex background environment, the precision and stability are crucial for any gesture detection methods. There are many models proposed in recent years to detect moving hands or other skin-color objects such as faces in complex environments. A skin-color threshold model was employed by Padam *et al.* [2] in the skin detection procedure, where the hand is supposed to be the largest skin-color object, and thus the non-hand components were filtered by comparing their areas. According to the skin-color clustering on the color space, the statistical models, such as elliptic boundary model and mixture Gaussian model, were employed to detect skin-color regions [3, 4]. Both of them have optimal time spent and can be applied to a low-complexity background. In [5], a mean-shift segmentation algorithm was employed to segment the image into homogeneous regions, which were then labelled by using the AdaBoost classification method. For mean-shift based methods, the target color histogram was required in advance and its detection time will greatly increase with the increase of search number of iterations. The FloatBoost learning algorithm was proposed by Li *et al.* [6] to extend the original AdaBoost algorithm by incorporating the idea of floating search into AdaBoost. FloatBoost used the backtracking technique to remove the weak classifiers once they no longer contributed to the decrease of the training error rate. To improve the hand detection accuracy, the Kinect sensor was applied to capture both the color image and its corresponding depth map [7]. With the addition of depth information, the hand object can be detected in the cluttered backgrounds and lighting conditions.

As for moving targets tracking, CamShift algorithm was employed by [8, 9] according to the change of color probability distribution, to adaptively control the size of the track window and the distribution pattern of target in tracking. CamShift based methods performed well in the cases of simple and constant backgrounds. Particle filter algorithm [10, 11] randomly took some posterior probability samples to approximately represent the whole posterior probability density distribution of the target state variables and used the current value and historical observations to estimate the current state of the target. Due to a relatively high computational complexity, it is difficult to meet the requirements of real time tracking in dynamic gesture recognition system. Kalman filter was one of the widely used tracking algorithms [12, 13], and it established the state model of system by maximizing the posteriori probability of history measurements to forecast the target state. Traditional linear extrapolation method was employed in [14, 15] with the assumption that the hands always maintain uniform linear motion on both horizontal and vertical directions in each neighboring frames.

For gesture recognition, the hidden Markov model (HMM) [16,17] was a widely used approach to solve the problem of gesture classification. HMM based recognition methods treat

each gesture as a set of states associated with the probabilities of initial, transitional, and output states, which were learned from the training data. Then the most probable gesture category was obtained by applying a model with the maximal probability. Conditional random field (CRF) [18,19] was a discriminative probabilistic model, yet it avoided the independence assumption and Markov property. Recently, some longest common subsequence (LCS) based methods have been proposed for dynamic gesture recognition [20, 21]. Specifically, in [20], most probable longest common subsequence (MPLCS) was proposed to measure the similarities between the probabilistic templates and the gesture for recognition, where the probabilistic templates for each pre-defined gesture patterns were trained beforehand. In addition, the algorithm permitted a more general representation of the data set by taking into account the possible trajectory distortions with different probabilities through using a probabilistic 2-D template rather than a deterministic 1-D template. In [21], most discriminating segment-longest common subsequence (MDSLCS) was proposed, the algorithm obtained a more discriminative classifier via extracting and recognizing the discriminating segments rather than the full gestures. Note that the discriminating segments in [21] was defined as a subgesture which is the most distinguishable subsegment relative to other gesture segments.

In this paper, an efficient gesture recognition method is put forwarded which can achieve a higher recognition rate by using a standard 2D camera. There are three highlights in this paper. The first is that we combine the three-frame difference method and skin-color elliptic boundary model in gesture detection procedure to effectively detect hand gestures in the skin-color like backgrounds. The second is that, to improve the accuracy of gesture tracking in the case of occlusion and hands-overlapping, we propose a direction-based adaptive linear extrapolation predictor, where the to-be-predicted coordinate is determined according to an occlusion direction judgment. Thirdly, in the process of gesture classification, we apply an improved LCS algorithm by incorporating the velocity information to remove the interferential subgestures. Furthermore, we seek to get the most probable matched template by comparing the relative LCS length of each gesture, i.e., the ratio between the length of LCS and the total length of each template, rather than the length of LCS for each gesture.

In the sections that follow, Section 2 is devoted to the exposition of the proposed system; Section 3 deals with the experimental results and the corresponding analysis. Finally, Section 4 is the conclusion reached.

2. Proposed system

2.1 Overview of the system

Fig. 1 shows an overview of the proposed system. Firstly, a user feeds a gesture to the system and the gesture within the scene is captured by the camera instantaneously. Next, the captured video sequence is transmitted to the computer, and gesture recognition algorithm is activated to match the detected gesture with our pre-defined patterns. Specifically, three steps, i.e., gesture detection, tracking and recognition, are taken to recognize the dynamic gesture from the video sequence with interference, and to eliminate the distractions and produce an accurate result with some improvements. The improvements of above-mentioned three steps will be explained in detail in the following sub-sections. Finally, the output of the system will show whether the captured sequence contains any significant gesture or merely meaningless gesture. If it is the former, the system will recognize the most probable gesture from our pre-established gesture template set.

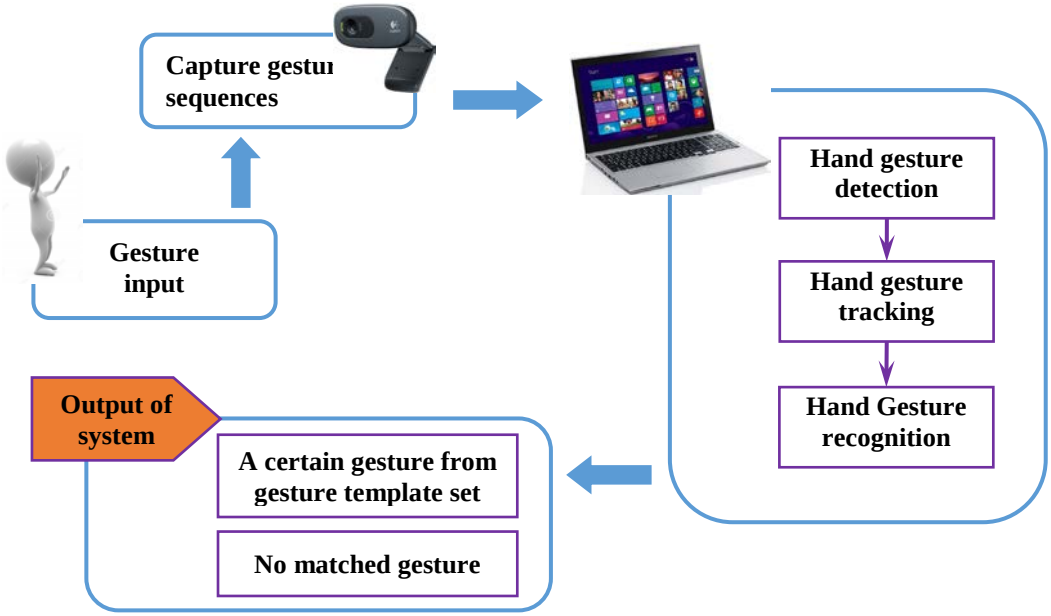


Fig. 1. Overview of the proposed system

2.2 Hand gesture detection

Gesture detection refers to the process of identifying the hand regions from a captured video sequence. A high accuracy and efficient detection algorithm can contribute to the improvement of the final recognizing performance. In our study, we employ the motion and skin-color features to detect the dynamic hand gesture. Specifically, we extract the motion and skin color features by using three-frame difference and the skin-color elliptic boundary model respectively. Using this combination, the gesture can be detected effectively in a complex background environment.

Three-frame difference is one of the popular methods for motion detection [22, 23]. Firstly, we extract three consecutive frames from the to-be-detected video sequence and convert them into grayscale intensity images, where the converted $(\tau\text{-}1)^{th}$, τ^{th} and $(\tau\text{+}1)^{th}$ gray frames are denoted by $f_{\tau\text{-}1}$, f_{τ} and $f_{\tau+1}$, respectively. Next, two binary maps, denoted by D_1 and D_2, are determined according to the absolute difference between $f_{\tau\text{-}1}$ and f_{τ}, and the absolute difference between f_{τ} and $f_{\tau+1}$, respectively.

$$D_1(x,y)=\begin{cases}1, & |f_\tau(x,y)-f_{\tau-1}(x,y)|\geq T\\ 0, & |f_\tau(x,y)-f_{\tau-1}(x,y)|<T\end{cases} \tag{1}$$

$$D_2(x,y)=\begin{cases}1, & |f_{\tau+1}(x,y)-f_\tau(x,y)|\geq T\\ 0, & |f_{\tau+1}(x,y)-f_\tau(x,y)|<T\end{cases} \tag{2}$$

where $f_\tau(x, y)$ represents the intensity value of pixel at location (x, y) in the τ^{th} frame, and T is a preset threshold, which is empirically set as 10 in our experiments, to remove the tiny movement area (such as face area). Finally, the target contours map D is obtained by executing the logical AND operation between D_1 and D_2.

$$D(x,y)=D_1(x,y)\,\&\,D_2(x,y) \tag{3}$$

Fig. 2 illustrates a schematic diagram of three-frame difference, where the three rectangles in the left column indicate a moving object in three continuous frames. Beyond that, the wide red borders in the middle column indicate the difference between adjacent two frames (top for

$f_{\tau\text{-}1}$ and f_{τ}, and bottom for f_{τ} and $f_{\tau+1}$), and tiny red borders in the right column represent the manipulation of three-frame difference by equation (3).

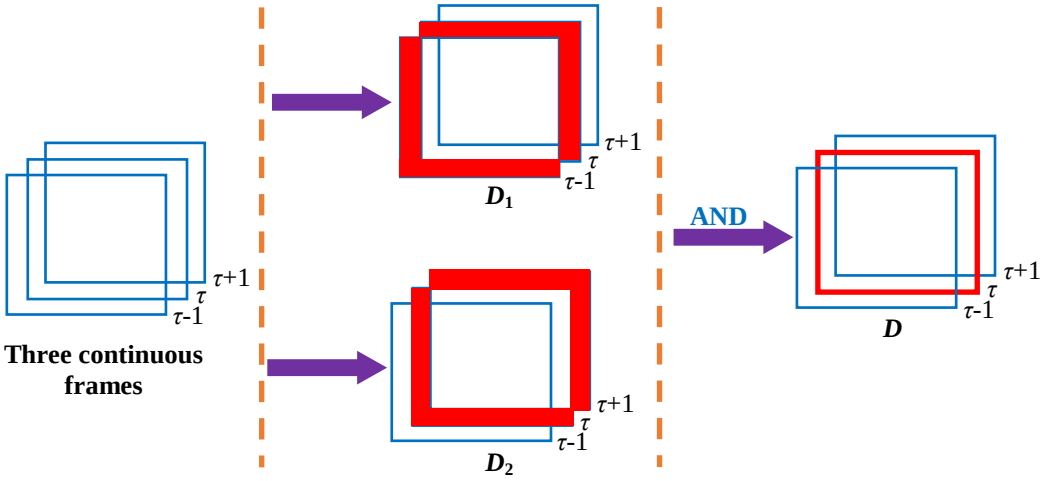


Fig. 2. The schematic diagram of three-frame difference method

Once the moving regions are identified, we employ the skin-color elliptic boundary model, which is a statistical model and firstly proposed by Hsu *et al.* [3], to detect skins on the moving regions. As demonstrated in [3], a pixel will be classified as a skin-color pixel once it satisfies the following equations:

$$\frac{(x-eC_x)^2}{a^2}+\frac{(y-eC_y)^2}{b^2}=1 \tag{4}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}=\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} C_b & -C_x \\ C_r & -C_y \end{bmatrix} \tag{5}$$

where, C_x=109.38, C_y=152.02, θ=2.53 rad, eC_x=1.60, eC_y=2.41, a=25.39, b=14.03. C_b and C_r represent the blue and red chrominance of a given pixel in the YC_rC_b color space, respectively. More details can refer to [3].

For the sake of description, we take the original frame shown in **Fig. 3(a)** as an example. First, a preliminary motion region, shown in **Fig. 3(b)**, is extracted using three-frame difference method. As can be seen from **Fig. 3(b)**, the preliminary region does not contain complete outlines, and even worse, there are full of irregular holes. To improve the detection accuracy, we fill the holes using morphological dilation operation to generate a binary moving area mask as shown in **Fig. 3(c)**. To refine the gesture region, we use the skin-color elliptic boundary model to localize the skin regions in the pre-determined moving area mask, as shown in **Fig. 3(d)**. Finally, we mark the target with the red ellipse as shown in **Fig. 3(e)** by fitting the contours of the target.

Fig. 4 shows the flowchart of the proposed method. It should be noted that in the process there are short periods of time when the hand gesture is static or merely with a small range of motion, and therefore, the three-frame difference method may not work properly. To avoid this kind of failure, we record the numbers of skin blobs, according to the detected isolated skin regions, in the current and preceding frames, denoted by N_{τ} and $N_{\tau\text{-}1}$, respectively, and compare the sizes of them. In normal conditions, i.e., when $N_{\tau} \geq N_{\tau\text{-}1}$, we directly extract the contours of skin blobs and execute ellipse fitting. While, in the condition of $N_{\tau} < N_{\tau\text{-}1}$, i.e., some skin blobs to be detected are in the static state, we generate an extended motion area by

applying the logical OR operator onto the detection map of current frame and that of preceding frame. Then, another skin color detection operation is executed on the redefined motion area before generating the final detected gesture ellipses. At this point, we have explained how to extract precise gesture regions from input video sequence.

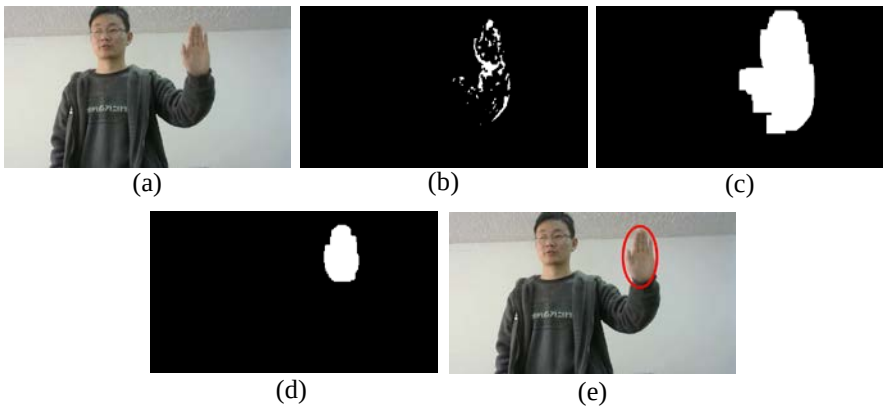


Fig. 3. Gesture detection result for an input video sequence: (a) original frame, (b) preliminary motion area generation by applying three-frame difference method, (c) morphological process, (d) binary detection map by using skin-color elliptic boundary model, and (e) final refined detection result by using ellipse fitting

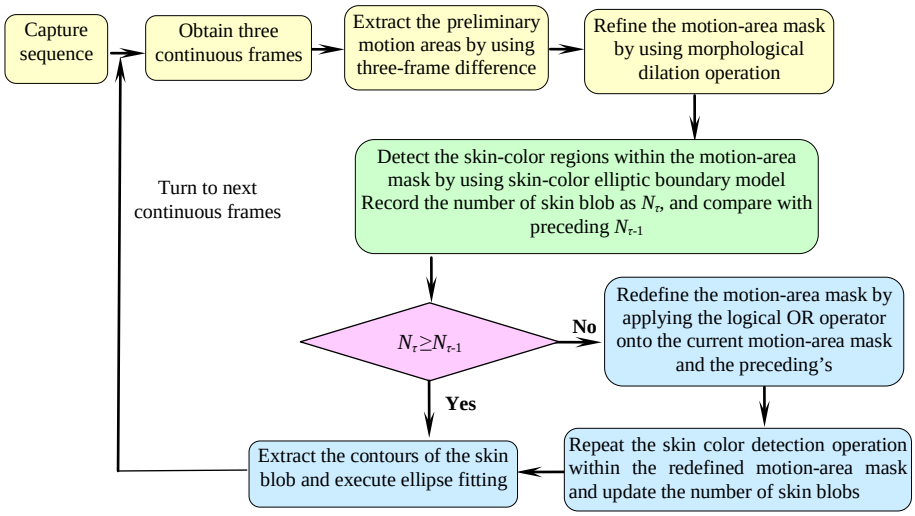


Fig. 4. The flowchart of hand gesture detection process

2.3 Dynamic gesture tracking

After gesture detection, the next step is to track dynamic gesture. Considering the influence of occlusion and hands-overlapping, a dynamic gesture tracking method is proposed in this sub-section with the assistance of improved linear extrapolation predictor.

Traditional linear extrapolation employed in [14, 15] is based on the assumption that the hands always maintain uniform linear motion on both x and y coordinates between neighboring states. Therefore, the predicted position can be represented as $(x_\tau+\Delta x_\tau, y_\tau+\Delta y_\tau)$, where (x_τ, y_τ) is the coordinate of the current location, Δx_τ and Δy_τ are the displacements between the current and the preceding frames along x and y coordinates respectively. However,

actually, a free-air gesture usually follows random irregular movement and the velocity changes a lot even in the neighboring frame. To reduce these prediction deviations, we use the average displacement of former two frames as the future-frame displacement. **Fig. 5** shows the prediction diagram of linear extrapolation based on the average displacement, and the predicted location, denoted by (x_p, y_p), can be calculated by:

$$x_p = x_\tau + \frac{1}{2}(\Delta x_\tau + \Delta x_{\tau-1}) = \frac{3}{2}x_\tau - \frac{1}{2}x_{\tau-2} \tag{6}$$

$$y_p = y_\tau + \frac{1}{2}(\Delta y_\tau + \Delta y_{\tau-1}) = \frac{3}{2}y_\tau - \frac{1}{2}y_{\tau-2} \tag{7}$$

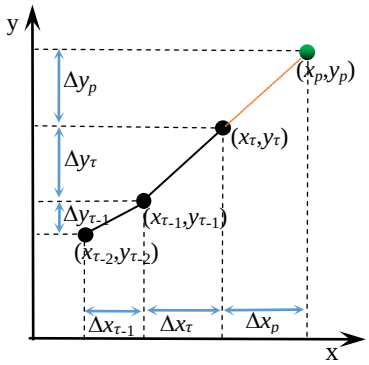
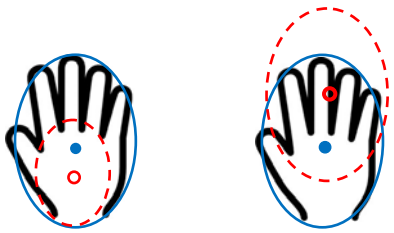


Fig. 5. Linear extrapolation based on the average displacement

However, the deviations between the gesture centroid and actual position, caused by unstable illumination, slight hand shaking, and other interference as shown in **Fig. 6**, are inevitable in the natural environment. Specifically, as can be seen in **Fig. 6(a)**, the detected centroid (red hollow circle) is deviated from actual coordinate (blue solid circle) because only part of the hand is detected, while in **Fig. 6(b)**, the deviation is caused by a slightly hand shaking. Since the detected centroid will be ceaselessly updated in the following frames, therefore, the deviation merely exists a short time and its influence can be neglected in most normal cases. Nevertheless, in the case of occlusion or hands-overlapping, this kind of deviation will deteriorate the final tracking result due to the vacancy of the detectable occluded gesture positions. For most tracking methods, the missing coordinates are replaced by the predicted position and the prediction error is expanded during the entire occlusion process. **Fig. 7** shows an example of a hand obscured by a book. Predicted position deviates from the actual position in frame 192, and is beyond the view of the camera in frame 197. In **Fig. 7(d)**, the obscured hand is falsely tracked as a new gesture target when the hand is pulled away from the book in frame 202.



(a) (b)

Fig. 6. The deviation between the detected gesture centroid and actual position:
(a) the hand not fully detected and (b) a slight hand shaking

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

Fig. 7. A Tracking failure caused by an inaccurate predictor: (a) frame 187, (b) frame 192, (c) frame 197 and (d) frame 202

Fig. 8 schematically displays the process of the proposed prediction method in the occasions of occlusion or hands- overlapping, where the time τ is the approaching instant of occlusion or hands-overlapping. **Fig. 8** shows the situation where the actual direction of the target is approximately parallel to the x axis, and a deviation has existed in time τ, where the actual position and the detected gesture centroid are represented by a blue and black solid circles, respectively. If we predict the centroid in the next frame by equations (6) and (7), the predicted coordinate (x_p, y_p), represented by a green solid circle, will deviate from the actual point (a blue solid circle) by a large margin and the deviation will increase until the end of occlusion and hands-overlapping.

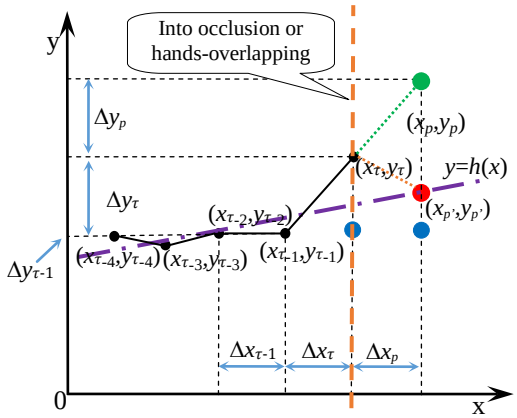


Fig. 8. The proposed gesture prediction method using adaptive linear extrapolation predictor

In order to obtain a more accurate prediction coordinate in the case of occlusion or overlapping, the coordinate (x_p', y_p'), represented by a red solid circle in **Fig. 8**, is determined by following two steps:

Step 1: fit a straight line according to the former five detected gesture centroids, represented by $\mathcal{L}$: $y = h(x)$

Step 2: group the moving gesture directions into two occlusion categories according to the slope of the line $\mathcal{L}$, denoted by k here. If $k \in$ (-1, 1), the direction of occlusion is regarded as horizontal and x_p' is directly calculated by equation (6), while y_p' is calculated by

$$y_p' = h(x_p') \tag{8}$$

otherwise, y_p' is calculated by (7) and x_p' is calculated by

$$x_p' = h^{-1}(y_p') \tag{9}$$

where, h^{-1} is the inverse function of h.

Then, by referring to [15], we associate the detected hand region ellipses, denoted by yellow

solid ellipses with centroids C_1 and C_2, with their corresponding prediction coordinates, denoted by points P_1 and P_2 as shown in **Fig. 9**. Now we define a metric $M(P, C)$ to measure the distance between the detected gesture centroid (x_c, y_c) and the predicted coordinate (x_p, y_p) as follows:

$$M(P,C) = \sqrt{(x_c - x_p)^2 + (y_c - y_p)^2} \tag{10}$$

Besides, we set a threshold γ to determine whether a matching is successful or not, where γ is decided by the major semi-axis of the detected ellipse in the preceding frame. If $M(P, C) \leq \gamma$, such as $M(P_1, C_1)$ indicated in **Fig. 9**, the detected centroid is regarded as a successful matching with its corresponding prediction coordinate, i.e., the detected centroid is considered to belong to a previous tracking object and is added to its corresponding trajectory. Otherwise, if every probable $M(P, C)$ is larger than γ, the detected object, such as C_2 in **Fig. 9**, is considered as a new target, and meantime, the mismatched prediction coordinates, such as P_2 in **Fig. 9**, are removed to suppress the expansion of prediction error. Thus far, we have interpreted how to track the gesture trajectory with an adaptive linear extrapolation predictor.

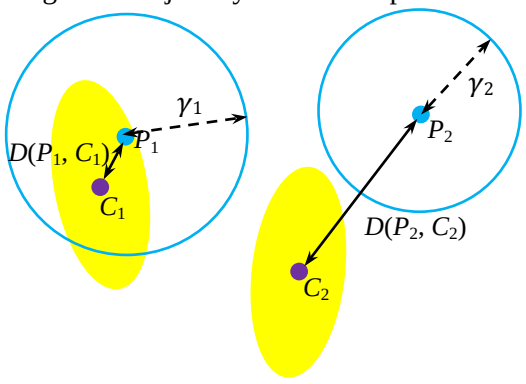


Fig. 9. Gesture matching for the detected centroids in current frame (C_1 and C_2) and their corresponding prediction coordinates from the preceding frame (P_1 and P_2)

2.4 Gesture recognition incorporating improved LCS and velocity information

In order to establish whether the tracked trajectory falls into the corresponding gesture category in line with semantics, a classification method integrating an improved LCS and velocity information is employed to reduce the influence of the interferential and transitional parts in a gesture trajectory as shown in **Fig. 10**.

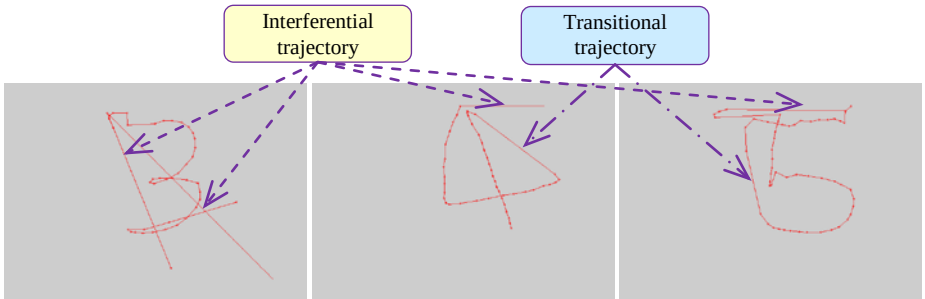


Fig. 10. The interferential and transitional parts in trajectories of digits “3”, “4” and “5”

Since any input gesture trajectory is composed of a series of points and their corresponding joining lines, hence, all gestures can be represented by a set of direction vectors as shown in **Fig. 11** (a), where we take the digit gesture “3” as an example. To facilitate the matching of the

obtained trajectory with pre-constructed gesture templates, we quantify all the direction vectors into integers in the range of [0, 15] using the sixteen direction vector diagram as shown in **Fig. 11** (b). The quantification value Q of the gestures is determined by

$$Q=\begin{cases}\theta, & \text{if } \Delta x_c \geq 0 \text{ and } \Delta y_c \geq 0\\ \theta+8, & \text{if } \Delta x_c < 0\\ \theta+16, & \text{if } \Delta x_c \geq 0 \text{ and } \Delta y_c < 0\end{cases} \tag{12}$$

where

$$\theta=\left\lfloor \frac{8}{\pi}\arctan\left(\frac{\Delta y_c}{\Delta x_c}\right) \right\rfloor \tag{13}$$

In Equation (13), the operator $\lfloor \cdot \rfloor$ indicates a round down operation and, Δx_c and Δy_c represent the displacements of gesture positions in current and preceding frames along x and y axis, i.e.,

$$\begin{cases}\Delta x_c = x_c(\tau) - x_c(\tau-1)\\ \Delta y_c = y_c(\tau) - y_c(\tau-1)\end{cases} \tag{14}$$

where $(x_c(\tau), y_c(\tau))$ and $(x_c(\tau\text{-}1), y_c(\tau\text{-}1))$ represent the coordinates of the gesture centroids at current frame τ and preceding frame $\tau - 1$, respectively.

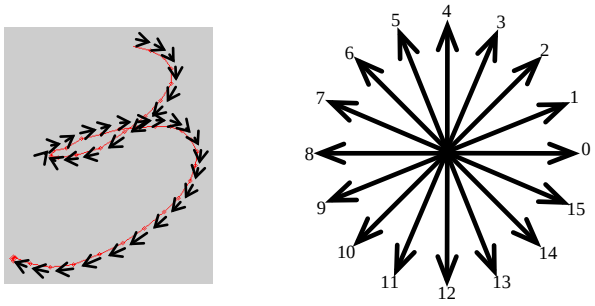


Fig. 11. Quantification of the gesture: (a) encoding for digit gesture "3" and (b) direction vector diagram

Velocity information is introduced as an assistance to eliminate the interference of the interferential and transitional trajectories, and thus, improving the recognition performance. From our observation, the velocities of movements for most interferential and transitional parts are higher than the meaningful parts. That is because meaningful hand gestures are driven by a careful and conscious thought in most occasions, while most of interferential gestures are caused by some subconscious movements. In order to measure the velocity of movement for each frame, here, we set the displacement of two gesture centroids in current and preceding frames as our velocity metric. As shown in **Fig. 10**, the velocities of interferential and transitional parts are much higher than other parts, i.e., they are shown to have relatively larger displacements. Then we set a velocity threshold, which is set to 40 pixels in our experiments, to distinguish high-speed or low-speed segments. If the displacement between two coordinates is larger than our preset threshold, its corresponding Q is changed to 16, i.e., a number out of its original range [0, 15].

To classify the gestures, we compare the similarity of a given numerical string and template strings by using an improved LCS algorithm. The traditionally LCS method measures the similarity by calculating the length of the longest common subsequence. Given two strings S_1 and S_2 with length I and J respectively, $S_1(i)$ and $S_2(j)$ are the i^{th} member of the string S_1 and the

j^{th} member of S_2 respectively, and $L(i, j)$ represents the value of the LCS matrix at the location of (i, j). **Fig. 12** shows an example of LCS measurement, where we take S_1= {2, 8, 6, 9, 4, 1} and S_2= {5, 2, 6, 8, 3, 9, 1} for example. The $L(i, j)$ is calculated as follows:

1. Both $L(0, j)$ and $L(i, 0)$ are set at 0
2. Rest $L(i, j)$ are calculated by

$$L(i,j)=\begin{cases}1+L(i-1,j-1), & \text{if } S_1(i)=S_2(j)\\ \max(L(i-1,j),L(i,j-1)), & \text{if } S_1(i)\neq S_2(j)\end{cases} \tag{15}$$

The length of the longest common subsequence is determined by the value of $L(I, J)$. In our example, the LCS length is 4, and the longest common subsequence is {2, 6, 9, 1}.

	j	**0**	**1**	**2**	**3**	**4**	**5**	**6**
i		S_1	**2**	**8**	**6**	**9**	**4**	**1**
0	S_2	0	0	0	0	0	0	0
1	**5**	0	0	0	0	0	0	0
2	**2**	0	1	1	1	1	1	1
3	**6**	0	1	1	2	2	2	2
4	**8**	0	1	2	2	2	2	2
5	**3**	0	1	2	2	2	2	2
6	**9**	0	1	2	2	3	3	3
7	**1**	0	1	2	2	3	3	4

Fig. 12. LCS matrix

The LCS can rapidly match the targets with predefined templates by using feature distance costs. However, it does not perform well in the case of subgestures, which is an unavoidable problem in all gesture recognition studies. A subgesture is a particular gesture that is a segment for another gesture as well. It exists widely in the gesture recognition system. For instance, the letter “c” is a subgesture of letter “d”, “e” and “o”, the digit “1” is a subgesture of digit “4” and “7”. Taking the recognition of digit “1” for example, the LCS length between our quantized trajectory and template of digit “1” coincides with the LCS length between ours and template “4” or “7”. In other words, in some occasions, digit “1” is falsely recognized as “4” or “7”. To solve this problem, we recognize gesture by comparing the ratio, denoted by R, between the length of the longest common subsequence and the total length of corresponding template, i.e., R is defined by

$$R(g,m)=\frac{L(g,m)}{T(m)} \tag{16}$$

where, $L(g. m)$ represents the length of LCS of a given gesture g and template m, and $T(m)$ represents the length of template m. Therefore, the recognized gesture is determined by seeking the largest R with a fixed g and all probable templates. Otherwise, if there is no appropriate template for matching, i.e., every R is smaller than the threshold, the system will regard the input as an unrecognized gesture.

3. Experimental Results and Analysis

In Section 2, we have presented the whole dynamic gesture recognition system in the order of detection, tracking and recognition. In this section, the accuracy and the effectivity of the

proposed method are to be experimentally evaluated. The program development environment is Microsoft visual studio 2010 and OpenCv 2.3.1. In addition, all the algorithms are executed on a PC with an Intel Core 2 Duo CPU T6570@ 2.10 GHz, a 64 - bit windos8.1 system and 4G RAM. The whole test process is videotaped in real time by Logitech HD Webcam C270.

3.1 Evaluation of hand gesture detection

In this sub-section, we evaluate the detection performance of the proposed method. It should be noted that the whole test process is videotaped under the complex background with many same skin-color regions to evaluate the detection performance of the proposed method. To demonstrate the efficiency of the proposed method, **Fig. 13** shows the hand detection results of the proposed approach and some other state-of-art methods, i.e., detection algorithm based on threshold value model [2], skin-color elliptic boundary model [3], Meanshift algorithm [5] and Haar-like classifier [24]. In addition, to evaluate the consuming time of the proposed method, we counted the average time of detection process in seven continuous frames for all five comparative methods as shown in **Fig. 14**. As shown in **Fig. 13** and **Fig. 14**, in all of these methods, the threshold value model and the skin-color elliptic boundary model take the shorter detection time but they have poor accuracy, and both of them falsely detected the same skin-color regions, such as the door and the desks. Since both methods merely extract the skin-color feature as the sole characteristic for gesture detection, it is inevitable to have some faults, especially under the background with same skin-color regions. For the other three algorithms, all hands are detected accurately. However, Meanshift is improper in real-time system due to its requirement of pre-statistical color histogram. Moreover, the time for Meanshift is the longest one as shown in **Fig. 14**. As for Haar-like classifier based method, training classifier is a time-consuming process and it needs a certain amount of positive and negative samples. In summary, to seek a trade-off between accuracy and real-time execution, the proposed detection scheme, by combining the motion and skin-color features, exhibits the best performance among all comparative methods.

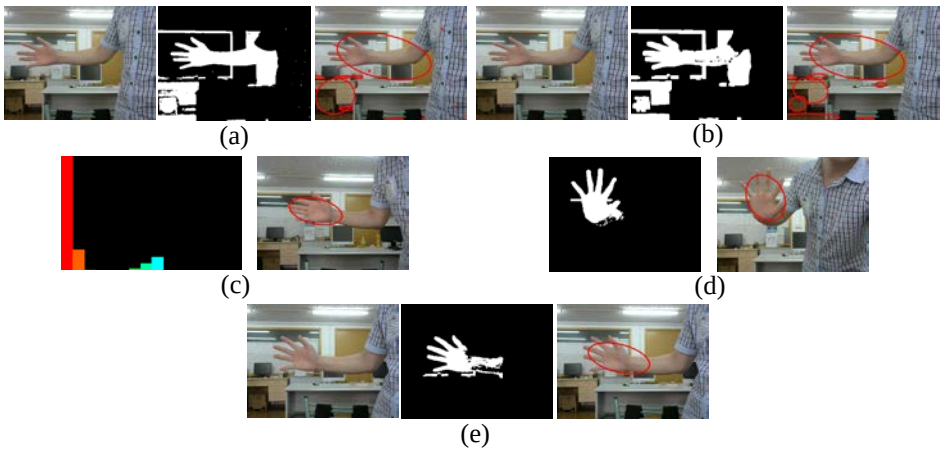


Fig. 13. The results of the hand gesture detection: (a) the detection result by threshold value model, (b) the detection result by skin-color elliptic boundary model, (c) the detection result by Meanshift based algorithm, (d) the detection result by Haar-like classifier based method and (e) the detection result of the proposed method

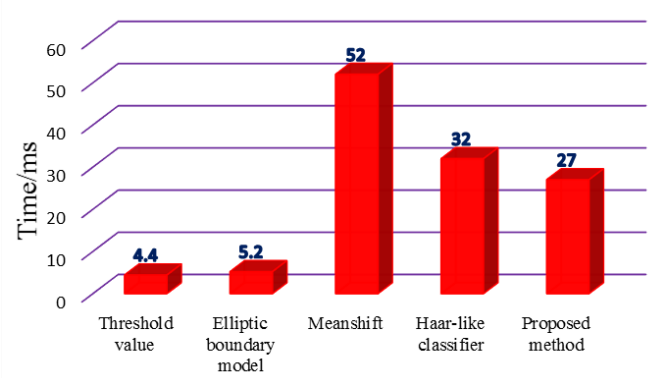


Fig. 14. Time cost comparison of five gesture detection algorithms

3.2 The accuracy of hand gesture tracking

To evaluate the tracking accuracy of the proposed method, we compare the actual trajectory with three predicted trajectories determined by three different predictors, including uniform velocity predictor employed in [14], Kalman filter predictor employed in [13] and the proposed predictor. **Fig. 15** illustrates an example for tracking accuracy comparison, where the authentic trajectory is denoted by red dot, and the trajectories obtained by the predictors of uniform velocity, Kalman filter and the proposed adaptive linear extrapolation are denoted by green square, orange rhombus and purple triangle, respectively. For quantitative comparison, we evaluate the performance by comparing the deviations between the actual locations and predicted locations. The deviation E is calculated by

$$E = \sqrt{(x_a - x_p)^2 + (y_a - y_p)^2} \tag{17}$$

where, (x_a, y_a) and (x_p, y_p) are the coordinates of the actual location and the predicted location, respectively. **Fig. 16** shows the comparison of deviations for three predictors at each indexed location. As shown in **Fig. 16**, in terms of deviation, the proposed method has the best performance in index of 2, 3, 6, 7, 8, 9, 11, 12, 13, 14, and the average deviation of the proposed method in **Fig. 16** is 3.374 pixels, while the average deviations by uniform velocity predictor and Kalman filter are 4.174 pixels and 6.257 pixels, respectively. It goes without saying that lower deviation lead to better tracking performance.

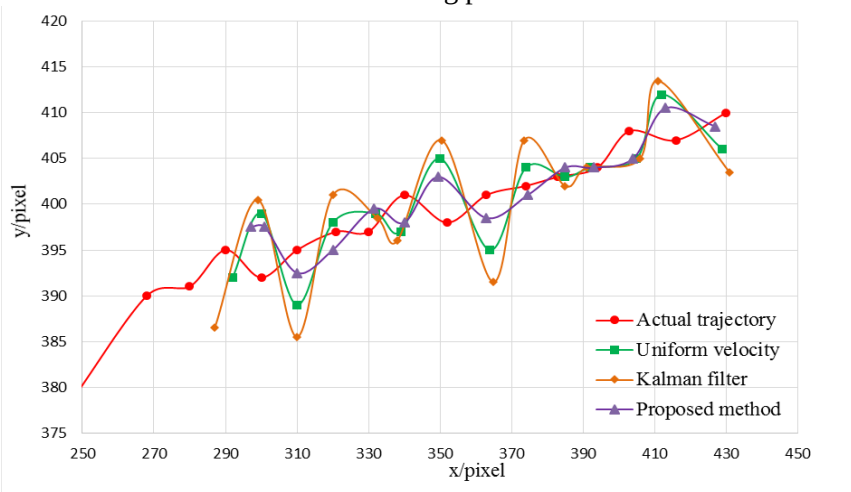


Fig. 15. The trajectory of the actual and prediction

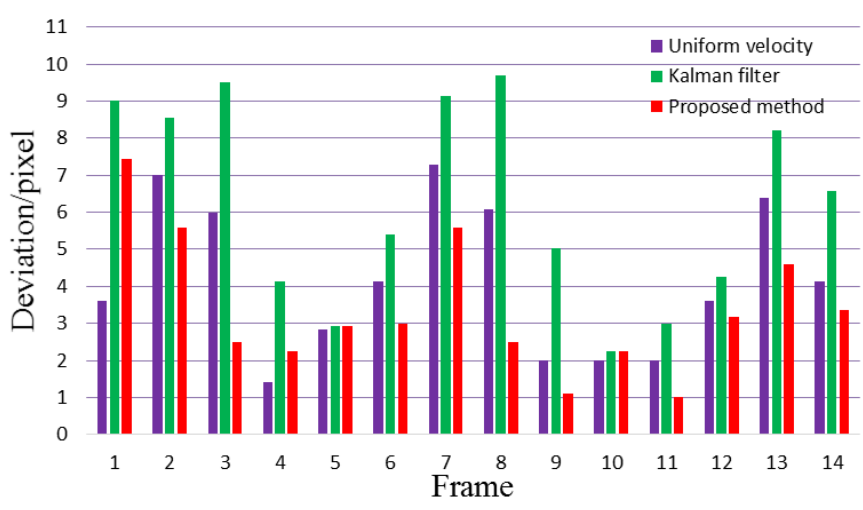


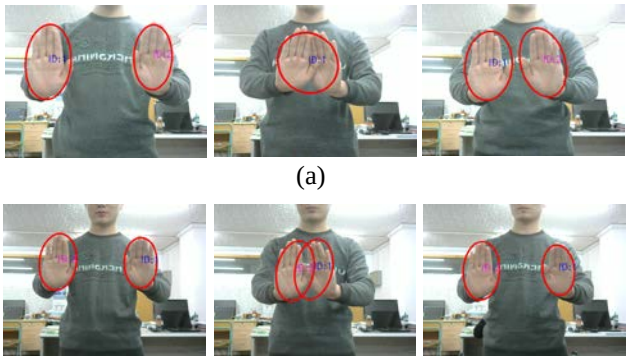
Fig. 16. Comparisons of deviations for three predictors

Next, the effectiveness of the proposed tracking method in the case of hand occlusion and hands-overlapping are verified in **Fig. 17** and **Fig. 18**, respectively. Specifically, **Fig. 17(a)** and **(b)** show the tracking results without or with the proposed predictor in the case of hand occluded by a book, respectively. As can be seen from comparison results, the trajectory can be very effectively tracked in the collusion process by the improved predictor. **Fig. 18(a)** and **(b)** show the tracking results without or with the proposed predictor in the case of hands overlapping, respectively. Comparing with the middle column of **Fig. 18**, two hands in overlapping condition can be correctly detected as the isolated targets by using our predictor. By using the predicted data rather than the actual data in some irregular cases such as occlusion and hands-overlapping, hand object can be well identified without any discontinuous and undistinguishable trajectory.

(a)

(b)

Fig. 17. Tracking results in the case of hand occlusion:
(a) the tracking results without predictor and (b) the tracking results of the proposed method



(a)

(b)

Fig. 18. Tracking results in case of hands overlapping:
(a) the tracking results without predictor and (b) the tracking results of the proposed method

3.3 The accuracy of hand gesture recognition

Trajectory classification is a challenging task in a dynamic gesture recognition system. To evaluate the accuracy of the proposed gesture recognition algorithm, we implemented the proposed method through recognizing the digit gestures from “0” to “9”. Before carrying out recognition procedure, we establish the digit gesture templates for matching with tracked trajectory as shown in **Fig. 19**. It should be pointed out that the templates we defined are more in line with our writing habits. Then, eight people (three females and five males) were invited to participate in our experiments, and each person executed 50 repetitions of each of 10 gesture digits in the front of our camera. Note that the digit gesture was captured and recognized in real time in our system. Therefore, we collected a total of 400 samples of each digit as our test dataset. In order to eliminate the confusion between the digits “0” and “6”, we calculated the distance between the start point and the end point and the distance of digit “6” is greater than digit “0” obviously. Finally, we counted the recognition rate of each digit and listed in **Table 1**. As can be seen from **Table 1**, in some cases, the digits “4” and “7” are mistakenly recognized as digit “1”. This is due to the fact that the digit “1” is a subgesture of both digit “4” and “7”, and the non-subgesture segments in “4” and “7” may not fully tracked. Eventually, we achieve an average recognition rate of 99.2%.

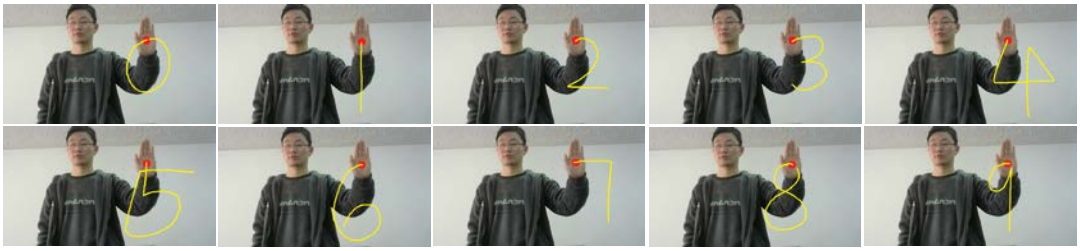


Fig. 19. Digit gestures

Table 1. Recognition rates of digit gestures (%)
Rows correspond to the ground truth digits, and colums correspond to the classified digits. The "UG" column shows unrecognized gesture, and " - "stands for zero.

	0	1	2	3	4	5	6	7	8	9	UG
0	**98.7**	-	-	-	-	-	1.3	-	-	-	-
1	-	**99.5**	-	-	-	-	-	0.3	-	0.2	-
2	-	-	**99.7**	-	-	-	-	-	-	-	0.3
3	-	-	0.6	**99.2**	-	-	-	-	-	-	0.2
4	-	0.8	-	-	**99.2**	-	-	-	-	-	-
5	-	-	-	0.5	-	**99**	-	-	-	-	0.5
6	0.3	-	-	-	-	-	**99.7**	-	-	-	-
7	-	1.3	-	-	-	-	-	**98.5**	-	-	0.2
8	-	-	-	-	-	0.3	-	-	**99.2**	-	0.5
9	0.3	-	-	-	-	-	-	-	-	**99.7**	-

We also make a comparison of recognition rates with other state-of-art methods, such as traditional LCS, MPLCS [20] and MDSLCS [21] based gesture recognition methods. The recognition rates of each digit ranged from 0 to 9 are shown in **Fig. 20**. By testing the classical LCS method in our dataset, it achieves an average recognition rate of 75.02%, which is lower than [20, 21] and us, especially in the cases of recognizing digit "4", "5" and "7". This is due to the influence of interference and subgestures. All the remaining comparison methods exhibit the satisfying recognition accuracies, specifically, MPLCS achieved a recognition rate of 98.7% as shown in [20], while for MDSLCS, it has an accuracy of 92.6%. Due to the improvements on all three phases (detecting, tracking and recognition), the recognition rate by our method outperforms the state-of-the-art recognition methods. Beyond that, the consuming time for the recognition of each gesture is around 80-110ms, which has been able to meet the real-time requirement. Note that the above provided time is based on the codecs without any optimization. However, it should be pointed out that, the test dataset in our algorithm is captured by a standard 2D camera, while the datasets in [20, 21] captured by 3D cameras. Therefore, the orientation options for a user interacting with a camera are restricted due to the lack of depth information. In addition, the ultimate goal of us is to achieve the effective recognition of continuous gestures. However, we have to admit that in the current stage, the accurate segmentation between two consecutive gestures is still a problem, which will be one of our future works.

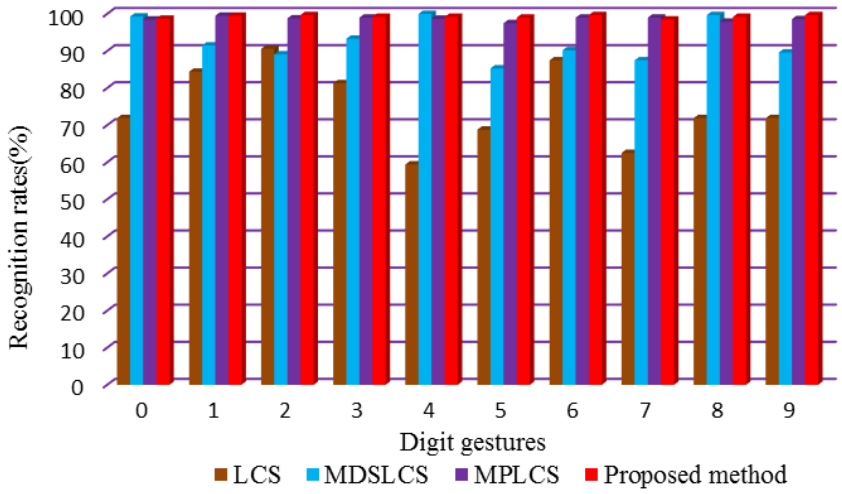


Fig. 20. Comparisons of the recognition rates by the proposed method and other state-of-the-art methods

4. Conclusion

In this paper, we have designed a novel dynamic gesture recognition system which allows users to unconstrainedly write the digit gestures or other gestures, whose template is included in a pre-established template set. Firstly, in gesture detection process, we combine the three-frame difference method and skin-color elliptic boundary model to detect the hand gesture without any other tiny moving regions. Next, we employ an adaptive linear extrapolation predictor to extract the gesture trajectory even in the cases of occlusion and hands- overlapping. Finally, we propose a relative length metric LCS algorithm incorporating velocity information for trajectory classification. Experiments have demonstrated the effectivity and robustness of the proposed system.

References

[1] Siddharth S. Rautaray and Anupam Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1-54, January, 2015. Article (CrossRef Link)

[2] S. Padam Priyal and Prabin Kumar Bora, "A robust static hand gesture recognition system using geometry based normalizations and Krawtchouk moments," *Pattern Recognition*, vol. 46, no. 8, pp. 2202-2219, August, 2013. Article (CrossRef Link)

[3] Rein Lien Hsu, Mohamed Abdel Mottaleb and Anil K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, no. 5, pp. 696-706, May, 2002. Article (CrossRef Link)

[4] Wei Ren Tan, Chee Seng Chan, Pratheepan Yogarajah and Joan Condell, "A fusion approach for efficient human skin detection," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 138-147, February, 2012. Article (CrossRef Link)

[5] Marko Subašić, Sven Lončarić and Adam Heđi, "Segmentation and labeling of face images for electronic documents," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5134-5143, April, 2012. Article (CrossRef Link)

[6] Stan Z. Li and Zhenqiu Zhang, "FloatBoost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112-1123, September, 2004. Article (CrossRef Link)

[7] Zhou Ren, Junsong Yuan, Jingjing Meng and Zhengyou Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110-1120, August, 2013. Article (CrossRef Link)

[8] Min Chun Hu, Ming Hsiu Chang, Ja Ling Wu and Lin Chi, "Robust camera calibration and player tracking in broadcast basketball video," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 266-279, April, 2011. Article (CrossRef Link)

[9] Kuo Hsien Hsia, Shao Fan Lien and Juhng Perng Su, "Moving target tracking based on CamShift approach and Kalman filter," *Applied Mathematics & Information Sciences*, vol. 9, no. 1, pp. 395-401, 2015. Article (CrossRef Link)

[10] Peixun Liu,Wenhui Li, Ying Wang and Hongyin Ni, "On-road multi-vehicle tracking algorithm based on an improved particle filter," *IET Intelligent Transport Systems*, vol. 9, no. 4, pp. 429-441, May, 2014. Article (CrossRef Link)

[11] Junghyun Kwon, Hee Seok Lee, Frank C. Park and Kyoung Mu Lee, "A Geometric Particle Filter for Template-Based Visual Tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 4, pp. 625-643, September, 2013. Article (CrossRef Link)

[12] Cheng Tse Chiang, Po Hsuan Tseng and Kai Ten Feng, "Hybrid unified Kalman tracking algorithms for heterogeneous wireless location systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 702-715, Febrary, 2012. Article (CrossRef Link)

[13] Emmanuel Marilly, Arnaud Gonguet, Olivier Martinot and Frederique Pain, "Gesture interactions with video: From algorithms to user evaluation," *Bell Labs Technical Journal*, vol. 17, no.4, pp.

103-118, March, 2013. Article (CrossRef Link)
[14] Heung-ii Suk, Bong Kee Sin and Seong Whan Lee, "Hand gesture recognition based on dynamic Bayesian network framework," *Pattern Recognition*, vol. 43, no. 9, pp. 3059-3072, September, 2010. Article (CrossRef Link)
[15] Antonis A. Argyros and Manolis I. A. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," *Lecture Notes in Computer Science*, vol. 3023, no. 3, pp. 368-379, 2004. Article (CrossRef Link)
[16] Hyeon Kyu Lee and Jin H. Kim, "An HMM-based threshold model approach for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961-973, October, 1999. Article (CrossRef Link)
[17] K. M. Vamsikrishna, Debi Prosad Dogra and Maunendra Sankar Desarkar, "Computer-vision-assisted palm rehabilitation with supervised learning," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 5, pp. 991-1001, May, 2016. Article (CrossRef Link)
[18] Cristian Sminchisescu, Atul Kanaujia and Dimitris Metaxas, "Conditional models for contextual human motion recognition," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 210-220, November–December, 2006. Article (CrossRef Link)
[19] Sotirios P. Chatzis, Dimitrios I. Kosmopoulos and Paul Doliotis, "A conditional random field-based model for joint sequence segmentation and classification," *Pattern Recognition*, vol. 46, no. 6, pp. 1569-1578, June, 2013. Article (CrossRef Link)
[20] Darya Frolova, Helman Stern and Sigal Berman, "Most probable longest common subsequence for recognition of gesture character input," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 871-880, June, 2013. Article (CrossRef Link)
[21] Helman Stern, Merav Shmueli and Sigal Berman, "Most discriminating segment-Longest common subsequence (MDSLCS) algorithm for dynamic hand gesture classification," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1980-1989, November, 2013. Article (CrossRef Link)
[22] Jinfu Yang, Wanlu Yang and Mingai Li, "An efficient moving object detection algorithm based on improved GMM and cropped frame technique," in *Proc. of IEEE International Conf. on Mechatronics and Automation*, pp. 658-663, August 5-8, 2012. Article (CrossRef Link)
[23] Jinhui Lan, Min Guo and Xiaojie Liu, "Real-time detection algorithm for moving vehicles in dynamic traffic environment," in *Proc. of IEEE International Conf. on Electro/Information Technology (EIT)*, pp. 1-6, May 9-11, 2013. Article (CrossRef Link)
[24] Paul Viola and Michael J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, May, 2004. Article (CrossRef Link)

Min Yuan received his B.S. degree in Power Electronics and Power Drives from University of Shanghai for Science and Technology, Shanghai, China, in July 2014. He is currently pursuing a master's degree at University of Shanghai for Science and Technology. His current research interests include human-machine interactive system, digital image processing and pattern recognition.

Heng Yao received the B. Sc. degree from Hefei University of Technology, China, in 2004, the M. Eng. degree from Shanghai Normal University, China, in 2008, and the Ph. D. degree in signal and information processing from Shanghai University, China, in 2012. Currently, he is with School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include multimedia security, image processing, and pattern recognition.

Chuan Qin received the B.S. degree in electronic engineering and the M.S. degree in signal and information processing from Hefei University of Technology, Anhui, China, in 2002 and 2005, respectively, and the Ph.D. degree in signal and information processing from Shanghai University, Shanghai, China, in 2008. Since 2008, he has been with the faculty of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, where he is currently an Associate Professor. He was with Feng Chia University at Taiwan as a Postdoctoral Researcher and Adjunct Assistant Professor from July 2010 to July 2012. His research interests include image processing and multimedia security. He has published more than 60 papers in these research areas.

Ying Tian received the B. Sc. degree in electrical engineering and automation from East China University of Science and Technology in 2010 and received Ph.D. degree in control theory and control engineering from East China University of Science and Technology in 2015, China. She is currently a lecturer in University of Shanghai for Science and Technology, Shanghai, China. Her research interests include big data analysis, process monitoring, fault diagnosis and quality control of complex industrial process.