International Journal of Computational Intelligence Systems, Vol.4, No. 2 (April, 2011).

Real-Time Hand Gesture Recognition using Motion Tracking

Chi-Man Pun[*], **Hong-Min Zhu**
Department of Computer and Information Science, University of Macau
Macau SAR, China
E-mail: {cmpun, ma86560}@umac.mo

Wei Feng
School of Computer Science and Technology, TianJin University
TianJin, China
E-mail: wfeng@ieee.org

Received: 28-06-2010
Accepted: 31-01-2011

Abstract

This paper proposes a real-time recognition system for hand gestures, which serves as an alternative of human computer interface. We focus on hand gestures performed by hand motions, and 10 hand signed digits are recognized. Gesturing hand in each video frame is firstly extracted by skin-subtraction approach. The track of hand motion with maximal stand deviation is selected for gesture. And we use the histogram models to recognize the gesture tracks. The system achieves a recognition rate of 97.33%.

Keywords: Hand gesture recognition, hand detection, motion tracking, and gesture classification.

1. Introduction

Human–computer interaction (HCI) is the study of interaction between users and computers. Interfaces have evolved from text-based interfaces through 2-D graphical-based interfaces, to multimedia-supported interfaces. In current virtual environments (VE) applications, keyboards, mice, wands and joysticks are still the most popular and dominant input HCI devices. However, they are inconvenient and unnatural at some viewpoint. These techniques may become a bottleneck in the effective utilization of the available powerful computing systems. Instead, VE applications require utilizing several different modalities and technologies and integrating them into a more immersive user experience [1]. Devices that sense body position and hand gestures, speech and sound, facial expression, haptic response, and other aspects of human behavior or state can be used so that the communication between the human and the VE can be more natural and powerful. Hand gestures are a powerful human-to-human communication modality.

To use human hands as a natural HCI, glove-based

[*] Corresponding author.

Published by Atlantis Press

devices have been used to capture human hand motions. However, the gloves and their attached wires are still quite cumbersome and awkward for users to wear, and moreover, the cost of the glove is often too expensive for regular users. With the latest advances in the fields of computer vision, image processing, and pattern recognition, real-time vision-based hand gesture classification is becoming more and more feasible for human–computer interaction in VE. Early research on vision-based hand tracking usually needs the help of markers or colored gloves to make the image processing easier. In the current state-of-the-art vision based hand tracking and gesture classification, the research is more focused on tracking the bare hand and recognizing hand gestures without the help of any markers and gloves.

Meanwhile, the vision-based hand gesture recognition system also needs to meet the requirements, including real-time performance, accuracy, and robustness. There has been various tools proposed for vision-based gesture recognition systems, we review some of them in section 2.

Section 3 gives the detail of our proposed real-time hand gesture recognition system, which involves hand detection, motion tracking and gesture classification procedures. Experimental results for each stage are given in section 4. Finally we conclude the paper in section 5.

2. Related Works

Hand location is the first stage of the generic hand gesture recognition system, considered as a feature extraction step used for the estimation of parameters of the chosen gesture model. Two types of cues are often used in the process of locating gesture operator: 1) color cues and 2) motion cues.

Color cues are applicable because of the characteristic color of the human skin. Most of the color segmentation techniques rely on histogram matching [2] or employ a simple look-up table approach [3], [4] based on the training data for the skin and possibly its surrounding areas. The major drawback of color-based localization techniques is the variability of the skin color footprint in different lighting conditions. This frequently results in undetected skin regions or falsely detected non-skin textures.

Motion cue is also commonly applied for gesturer localization and is used in conjunction with certain assumptions about the gesturer. For example, it is usually the case the gesturer is stationary with respect to the (also stationary) background. Hence, the main component of motion in the visual image is usually the motion of the arm/hand of the gesturer and can thus be used to localize her/him. This localization approach is used in[5]. The disadvantage is there are occasions when more than one gesturer is active at a time (active role transition periods) or the background is not stationary.

Once the gesturer is localized, the desired set of features can be detected. Hand and arm silhouettes are among the simplest, yet most frequently used features [6]. Contours represent another group of commonly used features [7]. Several different edge detection schemes can be used to produce contours. A frequently used feature in gesture analysis is the fingertip [8]. A simple and effective solution to the fingertip detection problem is to use marked gloves or color markers to designate the characteristic fingertips. In case if the motion trajectory is used for representing gestures, the center of gravity of a detected hand region can also be a usable feature, as in our proposed system.

To recognize extracted features as some specific gesture, there are different tools based on the approaches ranging from mathematical statistical modeling, computer vision and pattern recognition, image processing, connectionist systems, soft computing, etc. Most of the problems have been addressed based on statistical modeling, such as Principle Component Analysis (PCA), Hidden Markov Models (HMMs) [9], Kalman filtering [10], more advanced particle filtering [11] and condensation algorithms. Finite State Machine (FSM) has been effectively employed in modeling human gestures [12]. Connectionist approaches [11], involving multilayer perceptron (MLP), timedelay neural network (TDNN), and radial basis function network (RBFN), have been utilized in gesture recognition as well.

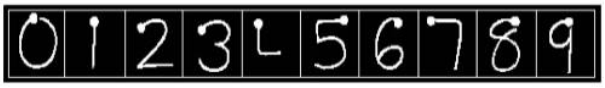


Figure 1. Palm’s Graffiti Digits.

The authors in [9] defined an one hand sign recognition system，both spatio-temporal Dynamic

Space Time Warping (DSTW) [13] and HMM are used to train a gesture model, which works with a fixed number of multiple candidates. A gesture can be recognized even when the hand location is highly ambiguous, and the background may be arbitrary and even contain other moving objects, and hand-over-face occlusions are allowed. Given the high computational complexity of the DSTW approach for a high number of fixed candidates, they introduced the pruning of classifiers in order to reduce cost. However, the system is still not able to operate in real-time or near real-time.

We will also evaluate our proposed gesture recognition system in the 10 Palm Graffiti Digits database as in [9], where users perform gestures corresponding to the 10 digits (shown in Figure 1). Gesturing hand is detected using skin-subtraction and the gesture classification is handled by gesture histogram models, which achieves a high recognition rate as well as the demand of real-time computing.

3. Proposed System

Our hand gesture recognition system follows the generic framework of a typical recognition system as shown in Figure 2, which involves two main stages:

i) . Motion tracking served as gesture feature extraction, which forms the motion track of gesturing hand to express the meaningful gestures, gesturing hand is detected in each frame and its center point is used for tracking the movement. The recognition rate of gestures will highly depend on the reliability of this procedure, and the processing speed of the system is also dominated by the efficiency of movement tracking;

ii). Gesture model training/matching firstly trains a model for each class of gesture, using a set of features extracted in motion tacking. And all tested tracks of unknown gesture are matching to these models and classified as the gesture with maximal similarity.

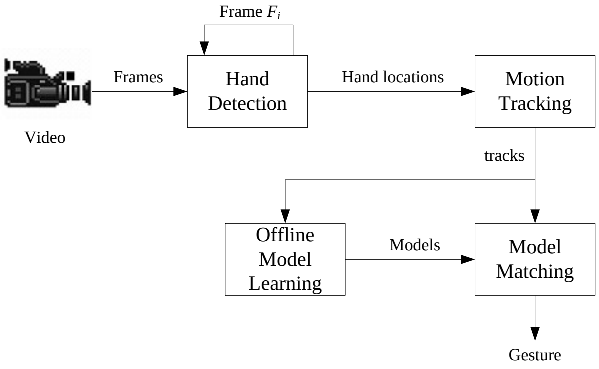


Figure 2. Framework of proposed system

Considered as a feature extraction process, hand motion tracking is the first of two stages in our hand gesture recognition system, which extracts the track of gesturing hand movement as a set of hand locations. Figure 3 summarizes the steps of hand motion tracking. Single frame processes above the point line shows the hand detection along with some post-processing procedures, which is to find center coordinates of detected hand regions in each frame. Hand motion tracking algorithm then form multiple track candidates from all calculated centers, and select the track with best feature measurement as the meaningful gesture performed by a user.

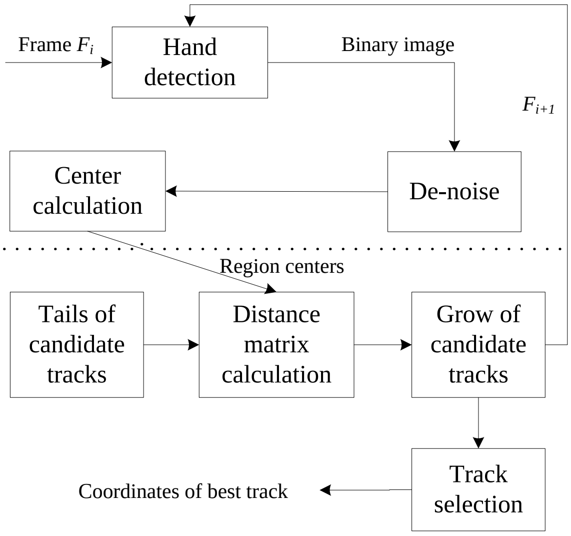


Figure 3. Workflow of gesture tracking

3.1. *Hand detection with skin-subtraction*

To form the track of hand motion, we need to locate the gesturing hand in each video frame. For hand detection and feature extraction, [9] use a generic skin color histogram to compute the skin likelihood image, which is applied for skin detection. Given skin and non-skin histograms of manually labeled skin/non-skin pixels we can compute the probability that a given color value belongs to the skin or non-skin classes:

$$P(rgb \mid skin) = \frac{s[rgb]}{T_s} \qquad (1)$$

$$P(rgb \mid \neg skin) = \frac{n[rgb]}{T_n} \qquad (2)$$

Where *s[rgb]* is the pixel count contained in bin *rgb* of the skin histogram, *n[rgb]* is the equivalent count from the non-skin histogram, and *Ts* and *Tn* are the total counts contained in the skin and non-skin histograms, respectively.

Other than the training based approaches, to develop a user independent skin detection algorithm, the standard range of skin color is available for different color spaces, Table 1 gives skin color ranges in RGB[14], HSV and YCbCr [15] color spaces.

Table 1. Skin color ranges

Color space	Skin color range
RGB	R > 95 & G > 40 & B > 20 Max(R, G, B) – Min(R, G, B) > 15 | R-G | > 15 & R > G & R > B
HSV	1 <= H <= 1.388 0.23 <= S <= 0.63 0.25 <= V <= 1
YCbCr	77<=Cb<=127 133<=Cr<=173

We adopt YCbCr for our skin classification, since RGB solution is not reliable when skin is under the lighting reflection, which is a common problem associate with skin color classification. YCbCr separates out the luminance signal (Y) that can be stored with high resolution or transmitted at high bandwidth, and two chroma components (CB and CR) that can be bandwidth-reduced, subsampled, compressed, or otherwise treated separately for improved system efficiency. And HSV solution is proved to be more time consuming although it’s more realistic to human vision.

Compared to skin color based hand detection that has the possibility to classified static objects in the background as skin object, background subtraction is another attractive solution for hand detection since it detects movable objects by intensity difference between two frames. Gesture that we focus on is performed by temporal hand motion, and while background subtraction can be applied in our hand detection, it may also detect other moveable objects. Both solutions will bring unexpected region as part of gesturing hand into movement tracking procedure which may failure the track extraction.

We propose the so-called *skin-subtraction* hand detection solution benefit from both skin classification and background subtraction approaches. Skin-liked objects in each frame are classified firstly, and then static skin regions from background are subtracted out by binary operations with “skin” image of the background frame. The detail of the hand detection is given as follows:

```
Algorithm I –hand detection of skin-subtraction
Input: a “background” frame $F_0$ and a gesture frame $F$
Step 1: calculate the binary images $I_0$ and $I$ for $F_0$ and $F$
    respectively. ex.: $I(j, k) = 1$ if $F(j, k) \in R_{skin}$ . $R_{skin}$
    is the skin range in YCbCr.
Step 2: $I$=exclusive-or($I$, $I_0$), remove static skin objects
    in $F_0$.
Step 3: $I$=AND($I$, NOT ($I_0$)), remove region of gesturing
    hand from $F_0$.
Step 4: $I$=closing (opening ($I$)).
Step 5: for each region $R_n$ in $I$, calculate $C(R_n)$ = center
    ($R_n$).
Output: $\{C(R_n) \mid n=1, 2 \ldots\}$
```

Step 1 classifies each pixel in the gesture frame and background frame as a skin pixel if it falls into the skin range of YCbCr color space, which will result two binary skin images. Step 2 performs logical exclusive-or between two skin images, which removes static skin-like pixels from background, the head of the user and the non-gesturing hand. As the “background” frame used for our experiment is selected from each user’s video sequence independently, and it’s not a pure background frame without any foreground, which cause gesturing hands from two frames are retained. This unexpected hand cluster from background frame can be further removed step 3, a logical AND operation between the result of step 2 and complement of background skin image.

Step 4 performs image opening followed by image closing operation on the binary image from step 3, which first removes isolated skin pixels as noise and then removes small holes inside skin blocks. Only the cluster of skin pixels with the size of boundary exceeds a threshold should be considered as a skin region, and the other detected pixels need to be removed as noise.

Step 5 calculates the center coordinate of gravity for each detection skin region, which will be used in motion tracking procedure to form the track of hand motion.

3.2. *Multi-candidates motion tracking*

After we get the features from hand detection procedure as a serial of centers coordinates of detected hand regions in each frame, the next task is to track the trajectory of hand movement. There are self-occlusions of 2D projected hand during gesturing, and the number of detected hand regions in each frame may be different. This problem challenges motion tracking that:

i) The gesture being tracked may be cut because of hand region lose or regions appear with long distance.
ii) Invisible region by occlusion belongs to the tracked gesture may appear again after a while.
iii) Regions in the tracked gesture may be not extracted from the gestured hand.

According to these uncertainties, we keep track of multiple gesture candidates, and tracks of gestures are updated based on the distance between region centers in current frame and the tail of each tracks, instead of regions detected in previous frame. Finally the one that best describes the trajectory of hand movement is selected.

Denote *C(I, M)* as detected *M* regions centers in frame *I*, and *G(N)* as *N* trajectory candidates being tracked. The hand motion tracking algorithm is given as:

```
Algorithm II – hand motion tracking
Input: region centers detected in each frame.
Step 1: initialize the start of $N=M$ gestures $G(N)$: $G(n) = C(I=1, m)$.
Step 2: For each frame $I>1$, do
Step 2.1: construct $L(N)$ which are tail locations of each
    gesture $G(n)$. Calculate matrix $D(N, M)$: distances
    between $C(I, M)$ and $L(N)$.
Step 2.2: repeatedly select $D(n, m)=min(D(N, M))$.
    If $D(n, m) < T_0$: append $C(I, m)$ to $G(n)$ and delete
        $D(n, m)$.
    Else: initialize $G(N+1)$ with start location of $C(I,
        m)$.
Step 3: select $G(N_0)$ that has the maximal standard
    deviation.
Output: coordinates sequence in $G(N_0)$.
```

Step 1 initializes *N* trajectory candidates, with the start location of each candidate assigned by each hand region center in the first frame. From second frame, an unequal number of *M* region centers may possibly be detected. With *N* tail points from the existing *N* candidates, step 2.1 calculates a *MxN* distance matrix that each center in current frame has a distance to the tail of each candidate. Step 2.2 repeatedly finds the minimal distance from distance matrix which is between *m*th center and *n*th candidate, if the distance bellows the threshold the center will be appended to the candidate; otherwise the center will initialize a new candidate with itself as the starting location. The problem that a continuous gesture cut by occluded region is handled in this algorithm, as long as the current region doesn't appear far away, the gesture can also be reconnected.

When all trajectory candidates are formed, we select the best motion track which describes the hand movement. One possible solution is based on the path lengths of candidates as motion performed by the most active gesturing hand tends to have the longest path. However, this idea may not robust when other objects keep moving in a small area that the candidate can also has a long path. In step 3 we use instead the standard deviation of the center coordinates along the gesture path, which is shown to be more reliable.

3.3. *Gesture recognition by histogram models*

Of those tracks of motions extracted from video sequences, we human can easily understand the meaning of a gesture and classify them into one of ten digits through 0 to 9, while it is not the case for computers. Thus the gesture recognition follows motion tracking, is to recognize the extracted tracks as meaningful gestures.

Figure 4 shows the workflow for our recognition steps of detected hand-signed digit tracks. In model matching stage, unknown tracks are classified into one of 10 gesture models according to their similarity with models. The performance of the hand gesture recognition system is concluded by the recognition rate, the percentage of tracks can be correctly classified.

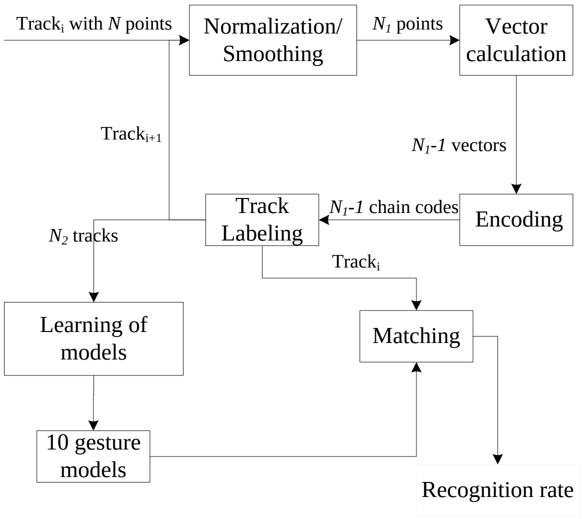


Figure 4. Workflow of gesture recognition

3.3.1 Pre-processing

According to the variation of gesturing speed, the same digit gesture performed by different users and different gestures performed by the same user have greatly different number of points in their motion tracks. To achieve user and digit independent recognition, all gesture tracks are needed to be firstly normalized to unified length of *N* points. We apply least-squares B-form spline approximation to normalize and smooth two dimensions of track points respectively

Another pre-processing step is the encoding of each track. We make use of vector to measure the direction at each step of hand movement, which result in *N-1* vectors for each track, and vectors are further be classified into 8 categories (see Figure 5) based on the ranges of direction they are in. This is referred as encoding of chain code.

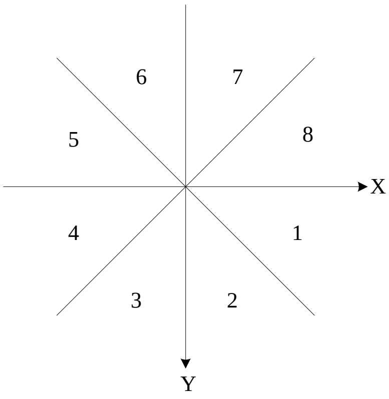


Figure 5. Categories of vector directions

3.3.2 Training

After all tracks are encoded by chain code, we manually label a set of tracks for each class of gesture through 0 to 9, for the training of gesture models. We propose a simple classification solution based on histogram distribution to model each gesture class, which is shown to be distinguishable among a finite set of gesture categories.

According to the chain code, there are 8 bins for each gesture's histogram model, the bin values are the average bin-wise counts of tracks with the same label. An unknown track is matched with ten gesture models and is classified into the model with minimal distance defined as:

$$dist = \min_{i=1}^{10}\left(\sum_{j=1}^{8}\left(|M(i,j) - T(j)| * w_{ij}\right)\right) \quad (3)$$

$$w_{ij} = 1 - M(i,j) / \sum_{k=1}^{8} M(i,k) \quad (4)$$

where *M(i, j)* is the count in bin *j* of gesture *i*'s histogram and *T(j)* is the count in bin *j* of an unknown track *T*. We also assign a weight to each bin consider the fact that when the count differences are the same between two bins, the bin in the model with larger count should has a smaller distance.

4. Experimental Results

We give experimental results of different stages of our proposed real-time hand gesture recognition system: hand detection, motion tracking and gesture recognition. The recognition result is compared with the one that we motivated [9].

Similar to the easy test set of hand-signed digits in [9], we test our system on our own provided data set. We also provide 30 video sequences and each sequence contains 10 gestured digits performed by the same user, that there are 300 tracks will be extracted in total, except that the users in our videos are required to wear long sleeve to ensure the precision of hand detection. Further more, the number of sequences from each user is not equal in our data set. The template of 10 digits through 0 to 9 is given in Figure 1, notice that “4” and “5” are different from our usual practice as they are required to be gestured by one stroke. We assume that the segmentation of 10 digits in each video is known, and we are not concern on the segmentation issue currently.

4.1. *Result of hand detection*

The frames we used for hand detection here are from the dataset of [9], with the intention to explain the reason of providing our own videos.

Figure 6 shows an example of skin-subtraction based hand detection. (b) and (d) are binary skin images for background frame (a) and current frame (c) respectively. Instead of subtract with the original grayscale frames in background subtraction, we perform the logical exclusive-or operation between (b) and (d) to get (e), which removes static skin-like pixels, and the head of the user and non-gesturing hand are also removed except their boundaries. The gesturing hand in two frames is retained as two main clusters, one of which from the background frame can be removed by a further logical AND operation between (e) and NOT(b), as shown in (f). And finally the region of our interested hand will be retained after de-noise process (g).

As the user wears short sleeve shirt, we can see that the detected hand region is longer than the fist which contains part of arm. This may lead motion tracking to be unstable as then center of region cannot represent the location of fist. So the gesturers in our dataset are required to wear long sleeve shirt instead to ensure the precision of hand detection.

Another problem that may fail the hand detection based on skin-subtraction is the color of shirt. If it is skin-liked and the gesturing hand appears in front of shirt, as shown in Figure 7, the region of hand will be deliquesced by skin pixels from shirt in background frame after exclusive-or operation, which can further be removed by de-noise process. Only a region of elbow retained as it is not intersected with cloth. This serves as another motivation to provide our dataset that gesturers are required to wear cloth with color not close to skin.

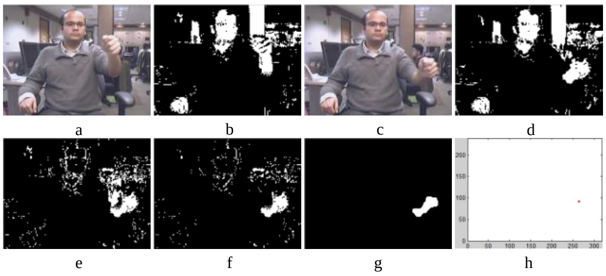


Figure 6. Hand detection of skin-subtraction: a) background frame. b) Skin in background. c) 20th frame. d) Skin in 20th frame. e) Exclusive-or of (b) and (d). f) AND (e, NOT (b)). g) De-noise. h) Region center.

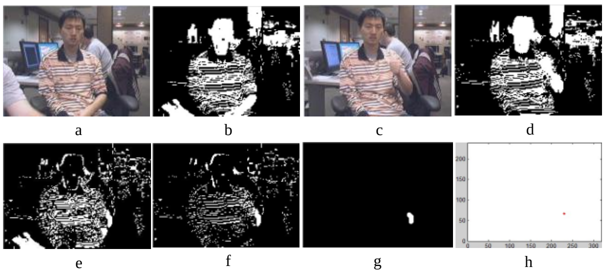
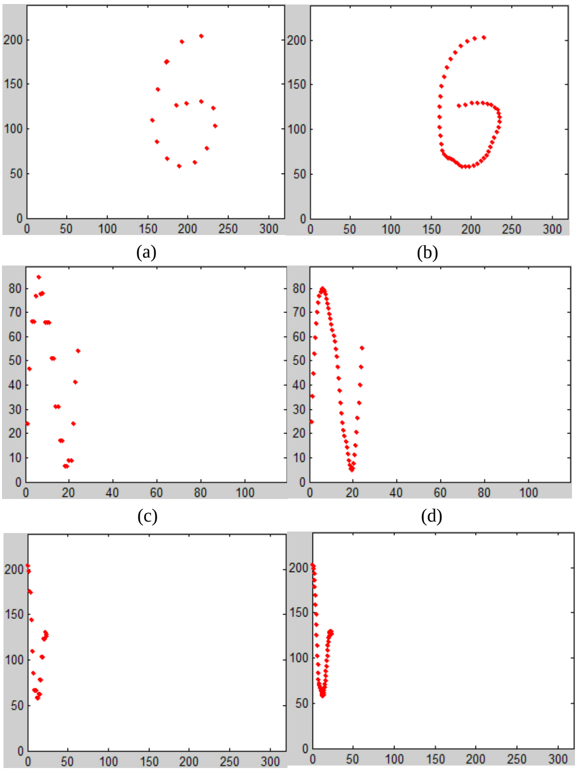


Figure 7. Problem of skin-liked shirt

4.2. *Result of motion tracking*

Motion tracking extracts the best track from multiple trajectory candidates according to stand deviation measurement. Each track is smoothed to a unified number of 51 points, which are converted to 50 vectors and further encoded to 50 chain codes. Figure 8(a) shows a detected motion track for digit “6” which consists a set of (x, y) points. (c)(e) shows x and y coordinates respectively of the original track, while (d)(f) are their 51 points normalization result. (b) gives the final track of digit “6” after normalization.

Figure 9 shows some example tracks of different digit gestures performed by different gesturers. The first column shows last frames in each digit video sequence and cumulative coordinates of detected hand region centers from each frame are plotted which indicate the change of hand positions. While the second column shows a serial of positions in each gesture with the number of points varying, the third column gives the normalized tracks that each track is interpreted with 51 points, and tracks are also smoothed.



(a) (b)

(c) (d)

（e） (f)

Figure 8. Normalization of digit track “6”

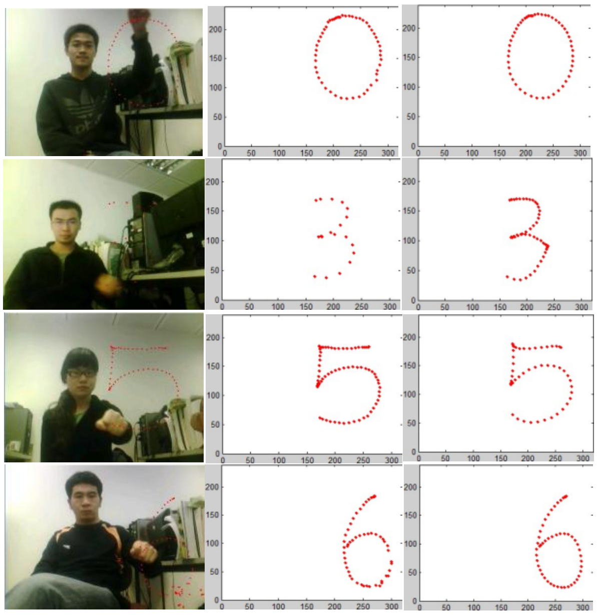


Figure 9. Examples of motion tracking. First column: last frame of gesture; Second column: selected best track; Third column: normalized track.

4.3. *Result of gesture recognition*

When we extracted all 300 motion tracks represented by chain codes from 30 video sequences, the next step is to classify each of tracks in to one of ten digits. We select 10 tracks for each digit class, totally 100 tracks to train the statistical histogram gesture models. Our histogram based solution didn’t take the temporal correlation among parts of a digit into consideration, but based on the statistical distribution of the movement directions (chain code), which is a “non-segment” solution compared to HMMs approach.

After 10 models are trained for each digit, we test the models by 300 tracks which are classified according to their histogram distances to ten models. The result of recognition rate is given in 2nd and 3rd columns of Table 2. As we can see from the result for each digit, all 30 tracks of digit “2” can be recognized, while the digit “4” can only achieve the recognition rate of 56.67%. The overall recognition rate is 86.33% while 41 tracks are incorrectly classified.

Table 2. Recognition rate in segment/non-segment models.

digit	Non-segment model		Segment model	
	Recg. count	Rate(%)	Recg. count	Rate(%)
0	29/30	96.67	28/30	93.33
1	29/30	96.67	30/30	100.00
2	30/30	100.00	30/30	100.00
3	25/30	83.33	30/30	100.00
4	17/30	56.67	30/30	100.00
5	23/30	76.67	29/30	96.67
6	28/30	93.33	28/30	93.33
7	24/30	80.00	29/30	96.67
8	28/30	93.33	29/30	96.67
9	26/30	86.67	29/30	96.67
overall	259/300	86.33	292/300	97.33

We summarize 41 miss-classified gestures in Table 3. There are 12 “4” incorrectly classified as 7 and 5 “7” incorrectly classified as “4”, that these 17 tracks takes more than 40% out of 41 tracks. Certainly we can improve the overall recognition rate by distinguishing “4” from “7” correctly and vice verse. We take the fact that both of digits “4” and “7” contain two dominating parts: a horizontal line and a vertical line, except that the order is different and the histogram model we used cannot distinguish the temporal difference. Our histogram model can be enhanced by segmented into two sub-models with the equal size, that is, 25 chain codes as the first half of each track are segmented to calculate the first sub-model for each digit, and is the same for the second sub-model. With this segmented histogram model, “4” and “7” are surely distinguishable, since first half chain codes of “4” (mainly a vertical line) has a significant distance to “7” (mainly a horizontal line) on the first sub-model which is also the same for second half codes. Results of gesture track recognition are summarized in the last two columns of Table 2. The overall recognition rate has been greatly improved to 97.33% that only 8 out of 300 tracks are miss-classified, while at most 2 out of 30 tracks for each digit.

Table 3. Count of incorrect classification by non-segment model

Gesture	**Classified as**	**Count**
0	8	1
1	9	1
3	2	2
	4	1
	5	2
4	2	1
	7	12
5	3	5
	6	2
6	0	1
	8	1
7	1	1
	4	5
8	0	1
	6	1
9	0	2
	6	1
	8	1

Our dataset is generated in office environment which is similar to the easy dataset of Alon and Athitsos’s work [9], with no distracters while the other moving objects are possible, except that users wear long-sleeved non-skin-liked shirts. We also assume that the gesture segmentation is known. They achieve the best detection rate on easy dataset of 94.6% when the subgesture reasoning is used and K=4, 5, 6 candidate hand regions are retained in each frame. Our recognition rate of 97.33% is comparable although we didn’t include subgesture reasoning since the gesture segmentation is known.

The runtime of our gesture recognition is mostly cost on the hand detection and motion tracking stage, which can achieve an average speed of 60 fps. According to the frame rate of current off-the-shelf cameras, our system can support the real-time processing requirement.

5. Conclusion

In this paper we proposed a hand gesture recognition system using motion tracking, which served as an alternative for new generation of human computer interface in modern computer vision systems.

As color cues based and motion cues based solutions are most commonly used for detect object in video frames, we proposed the skin-subtraction hand detection which benefit from both solutions. The best motion track is extracted from multiple track candidates based on their stand derivation measurement. Each track of gesture digit is normalized and smoothed, and encoded into chain code for training models of each gesture class. Compared with Hidden Markov Models (HMMs) tool use in our motivation, we proposed a simple model on the histogram distribution which is shown to be reliable for gesture classification. We achieve a recognition rate of 97.33% out of 300 digit gestures and the computational efficiency is around 60 fps which can support the requirement of real time applications.

There are several issues need to be handled in our future work. 1) Improve the hand detection and gesture tracking to be reliable and stable when the hand occludes with skin-liked objects. 2) Improve gesture recognition by HMMs. our histogram model may not be reliable for classification when the gesture classes are extended. Temporal relation is necessary to be considered. 3) Identify the start/end frame of each gesture by segmentation. 4) Extend the system to handle distracters. Other object like human move beside the gesturing person will make the gesture tracking even challenging.

Acknowledgements

The authors would like to thank the referees for their valuable comments. This work was supported in part by the Research Committee of the University of Macau.

References

[1] M. Turk, "Gesture recognition," *Handbook of Virtual Environment Technology,* 2001.

[2] S. Ahmad, "A Usable Real-Time 3D Hand Tracker," *IEEE Asilomar Conf.,* 1994.

[3] R. Kjeldsen, and J. Kender, "Finding Skin in Color Images," *Proc. Int'l Conf. Automatic Face and Gesture Recognition, Killington, Vt.,* pp. 312-317, Oct. 1996.

[4] F. K. H. Quek, T. Mysliwiec, and M. Zhao, "Finger Mouse: A Freehand Pointing Interface," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland*, pp. 372-377, June. 1995.

[5] W. T. Freeman, and C. D. Weissman, "Television Control by Hand Gestures," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland*, pp. 179-183, June 1995.

[6] J. J. K. a. T. S. Huang, "Vision-Based Hand Modeling and Tracking," *Proc. IEEE Int'l Conf. Computer Vision, Cambridge, Mass.*, June 1995.

[7] D. M. Gavrila, and L. S. Davis, "Towards 3D Model-Based Tracking and Recognition of Human Movement: A Multi-View Approach," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland*, pp. 272-277, June 1995.

[8] J. Lee, and T. L. Kunii, "Model-Based Analysis of Hand Posture," *IEEE Computer Graphics and Applications*, pp. 77-86, Sept. 1995.

[9] J. Alon, V. Athitsos, Q. Yuan *et al.*, "A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 31, no. 9, pp. 1685-1699, 2009.

[10] G. Welch, and G. Bishop, "An introduction to the Kalman filter," *Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, Tech. Rep. TR95041*, 2000.

[11] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," *Proc. IEEE,* vol. 92, no. 3, pp. 469-484, Mar. 2004.

[12] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recogn., Grenoble, France,* pp. 410-415, Mar. 2000.

[13] J. Alon, V. Athistos, Q. Yuan *et al.*, "Simultaneous localization and recognition of dynamic hand gestyres," *IEEE Motion Workshop*, pp. 254-260, 2005.

[14] J. Kovac, P. Peer, and F. Solina, "Human Skin Colour Clustering for Face Detection," *The IEEE Region 8 Computer as a tool EUROCON 2003,* vol. 2, pp. 144-148, Sept. 2003.

[15] G.-Z. Mao, Y.-L. Wu, M.-K. Hor *et al.*, "Real-Time Hand Detection and Tracking against Complex Background," *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 905-908, 2009.