

Visual Hand Gesture Segmentation Using Three-Phase Model Tracking Technique for Real-Time Gesture Interpretation System

Tsung-Han Tsai and Chung-Yuan Lin

Department of electrical engineering
National central university
Taiwan, R.O.C

Received August 2011; revised April 2012

ABSTRACT. *The task of automatic gesture segmentation is highly challenging due to the computational burden, the presence of unpredictable body motion and ambiguous nongesture hand motion. In this paper, a new approach is developed using Hausdorff-based model tracking technique for the application of real-time human-computer interaction. This paper proposed a three-phase model tracking approach. It consists of two main stages; one is the motion history analysis to classify dynamic gesture into preparation, retraction and nucleus state based on temporal relationship. The other is model tracking, which tracks signer model and object model with different constraint based on the classified state. Finally, gesture model is extracted based on matching object model and signer model and the hand gesture region is segmented from the gesture model. Experiments are performed to test the robustness of gesture segmentation under various hand scale and complex background. The segmentation error rate and computational complexity are also analyzed to demonstrate the proposed three-phase model tracking approach can be applicable to real-time human-computer interaction system.*

Keywords: Hausdorff distance, motion analysis, model track, hand gesture segmentation, humancomputer interaction

1. **INTRODUCTION.** The use of hand gestures provides an attractive to cumbersome interface devices for human-computer interaction (HCI). In particular, visual interpretation of hand gestures can help in achieving the ease and naturalness desired for HCI. This motivated a number of researchers concerned with the computer visionbased analysis and interpretation of hand gestures [1]. An important application of machine version, therefore, is to extent the interface between man and machine, allowing machine to directly perceive what its operator is doing. The ability to follow a hand moving in the space and to recognize a particular motion as a meaningful gesture is an essential step in intelligent system design and natural human-machine interaction.

To extract and recognize gestures, researches have used various techniques ranging from hidden Markov modeling (HMM), artificial neural networks (ANN), dynamic timing warping (DTW) and others techniques reviewed in [2][3][4][5]. However, these approaches require the signer to wear specialized gloves or when the background color is restricted. It reveals that these systems do not provide excellent means of human-computer interaction. To overcome the limitations of the individual cues, fusion of cues is explored for face localization in video, but not fully exploited for hand localization [1].

This paper concentrates on the problem of automatically separating hand location from the gesture sequence. Pure vision-based solutions usually rely on skin color and/or motion information to detect hands [6]-[8]. However, approaches based on predefined skin color models suffer from sensitivity with respect to changing illumination conditions. Motion-based hand segmentation approaches rely on the assumption that features important for gesture will be associated with motion. However, a usable HCI related applications require on-line procedure and real-time performance. Thus, reducing the desirable appearance based features or models of the gestures to decrease computational complexity have to be considered. This paper proposes an efficient model, named signer model, which can be automatically constructed without prior information. This idea comes from the degree of boundaries variation when the signer performs gesture. The boundary of the body will have less variation than the boundary of hand/arm parts. Fig 1(b) and (c) show an example on the degree of boundaries variation. Thus one can establish a signer model, which comprises model points in the signer except the hand/arm points, by tracking two different boundaries. Fig 1 shows the concept of the signer model construction. The signer model is an efficient feature because one can extract the hand/arm edge points by matching edge points between signer model and object model from the inputted sequence.

This paper develops a three-phase model tracking approach (TPMT). It consists of two main components. The first component, motion history analysis, classifies inputted gesture frame into three states, said as preparation, retraction and nucleus state, based on temporal relationship. The second component tracks signer model and object model with different constraint based on the classified state. Gesture model is then extracted based on matching object model and signer model. Finally, hand gesture regions are segmented from the gesture model.

2. MOTION HISTORY ANALYSIS. Human gestures are the dynamic process. It is important to consider the temporal characteristics of gestures. This may help in temporal segmentation of gestures from other unintentional hand/arm movements. In terms of our general definition of hand gestures, this is equivalent to determine the gesture interval. [1] calls this interval a gesture state. It has established that three phases make a gesture: preparation, nucleus, and retraction. Fig. 2 shows an example that dynamical representation of a gesture is separated into frames associated with these three states respectively. The preparation state consists of a preparatory movement that sets the hand in motion from some resting position. In the retraction state, the hand approach consists of a position for presenting a gesture. Finally, the nucleus of a gesture has some definite form and enhanced dynamic qualities. This involves classifying each inputted frame based on the performed gesture into preparation, retraction and nucleus state.

2.1. Concept. This paper uses a period of motion, named motion history, to identify gesture state in each inputted frame. Specifically, the proposed TPMT uses the statistic of motion information $m(t)$ and $r(u, v)$ of the consecutive frame to denote three gestures state on-line. $m(t)$ is noted as the summation of the t -th difference image in temporal domain

$$m(t) = \sum_{x \in I} d_t(x) \quad (1)$$

where $d_t(x)$ presents the difference image in position x of the frame I at time install t . $r(u, v)$ is denoted as the difference of motion information within a period of time install u to v

$$r(u, v) = m(u) - m(v) \quad (2)$$

A gesture sequence may contain several distinct and important gestures. In our observation, the nucleus state can be identified by local minimum of $r(u, v)$, due to the characteristics of held gestures to emphasize their important. The preparation state can be identified by the increased $r(u, v)$, and the retraction state can be identified by the decreased $r(u, v)$.

2.2. On-line motion history analysis. Since the hand/arm will have larger motion than the resting parts in the gesture sequence, the local minima of motion information can be obtained by observing differences of the motion information around time install t that converge to zero. Unfortunately, the nucleus state does not always take place at every local minima of motion information in reality, because signers will not completely stop their hand/arm when emphasizes the important of the gesture in reality. Besides, other factors such as illumination and noise may also affect the nucleus state judgment using the local minimum of motion information.

To overcome this problem, an additional state, named fleet nucleus state, is introduced. Fleet nucleus state deals the noise, illumination and body motion. It needs a counter to determine the period of the nucleus state when local minima have been detected. A nucleus state will be brief when it is caused by noise, illumination and body motion. Otherwise, a nucleus state will keep long when it is caused by actual gesture appearance. This paper proposes a finite state machine with the difference of the motion information, and judges the gesture state in each time install t . The four-state transition diagram is shown in Fig 3, where each state stands for a gesture state and comprises respective trigger conditions

- 1) Preparation state: in this state, a transformation to retraction state will only be triggered. This implies that the hand motion will start decreasing. Thus, one can judge whether the retraction state can be triggered by examining the ratio $r(t, t - 1)$. If the ratio $r(t, t - 1)$ becomes negative, then a retraction state is triggered.
- 2) Retraction state: in this state, a transformation to fleet nucleus state will only be triggered. This implies that the hand motion is at local minima. Thus, one can judge whether the nucleus state will be triggered by examining the ratio $r(t, t - 1)$. If the ratio $r(t, t - 1)$ becomes positive, a fleet nucleus state is triggered. While fleet nucleus state is triggered, *swing* and *base* parameter have to be determined. *Base* is determined as $m(t - 1)$ and *swing* is determined as follow: if $m(t - 1)$ is smaller than $T_{context}$ and $\max(r(t, t - 2), r(t - 1, t - 3))$ is smaller than $m(t - 1)$, then *swing* is determined as $\max(r(t, t - 2), r(t - 1, t - 3))$; else if $m(t - 1)$ is smaller than $T_{context}$ and $\max(r(t, t - 2), r(t - 1, t - 3))$ is larger than $m(t - 1)$, then *swing* is determined as $m(t - 1)$; else *swing* is determined as $\min(r(t, t - 2), r(t - 1, t - 3))$. These two parameters determine an oscillation range of motion information under incomplete stillness nucleus gesture during nucleus and fleet nucleus state.
- 3) Fleet nucleus state: in this state, three transformations will be triggered. The first one is a transformation to retraction state, if $m(t - 1)$ is larger than $base + swing$ or $m(t - 1)$ is smaller than $base - swing$, and ratio $r(t, t - 1)$ is positive. The second one is a transformation to preparation state, if $m(t - 1)$ is larger than $base + swing$ or $m(t - 1)$ is smaller than $base - swing$, and ratio $r(t, t - 1)$ is negative. These two cases imply that the triggered local minimum of motion information is caused by noise, illumination. The third one is a transformation to nucleus state, if the fleet nucleus state keeps triggered over four consecutive frames. This case implies that the meaningful gesture is in appearance
- 4) Nucleus state: in this state, a transformation to preparation state will only be triggered. This implies that the hand motion will start increasing. One can judge whether the

preparation state will be triggered by satisfying one of following two rules. One is the *continuous positive period* over four consecutive frames. The *continuous positive period* is a quantity presenting how many *continuous positive ratios* appeared in a period of frames. The *continuous positive ratio* is a sample frame where ratios satisfy that $r(t, t-1)$, $r(t-1, t-2)$, $r(t-2, t-3)$ are positive and $r(t, t-1) > r(t-1, t-2) > r(t-2, t-3)$. The other is $m(t-1)$ is larger than *base + swing* or $m(t-1)$ is smaller than *base - swing*.

3. MODEL TRACKING. The signer model and object model are tracked and matched to each other after gesture states have been obtained by motion history analysis. However, gray scale images are not normally suitable for template matching because they are too sensitive and too heavy in computational load. Instead, it is common to use binary edge images with fewer computations. The edge points of the models are not restricted to object boundaries, but they can also be in the interior such as eyes and mouth in a face. In TPMT, the edge images are obtained by the Canny operator [9]. Two different edge images are applied to TPMT, one is current edge model obtained by Canny operator at the current frame I_t . The other is object model obtained by Canny operator at difference image d_t . Moreover, a robust matching method must be able to handle binary models that are undergoing translation, rotation, and changes in shape. This excludes basic template matching where the new position is determined by the highest correlation between the model and subsequent frames. The Hough transform has been successfully applied to the detection of arbitrary shaped binary edge points [14]. However, it comes at a high computational cost, especially for a multidimensional Hough accumulator space that includes translation, rotation, and scaling. The Hausdorff approach, which matches the interested binary model against subsequent frames by minimizing the Hausdorff distance, is adopted. This approach is computationally efficient and robust to noise and changes in shape.

3.1. Signer model tracking phase. While edge plays an important role in extracting both the physical change of the surface of signer and the surface of gesture in the real scene, segmentation of the gesture in sequence suffers to separate moving edge belonging to hand/arm from great deal of moving edges. One of the simple ways is to construct a signer model SE which consists of body/head edge points except hand/arm edge points. The constructed signer model is used to compare with the object model in the nucleus state. Since the signer model consists of only body/head edge points, the hand/arm edge points can therefore be separated as those that deviates the signer model. Accordingly, the extracted hand/arm edge points are the edges corresponding to the gesture. Therefore, the key point in our algorithm is to obtain the edge points in object model corresponding to the edge points in the signer model.

The signer might rotate or change its shape as it signs through the video sequence. As a consequence the corresponding edge points in the signer model must be updated during the preparation state and the retraction state. More precisely, the model is actually not updated, but a new model is derived by selecting an appropriate set of edge points from the object model of the current difference image. Considering two frames and their difference images in the preparation state and the retraction state respectively as the example shown in Fig. 4, the pixels associated with considerable difference value are hand/arm region and the boundaries of head and body. This implies that the object model of the previous difference image contains an important cue for choosing the set of edge points forming the new signer model. The edges associated with the head/body of signer tend to enforce temporal continuity since the motion of head/body is small while a person is gesturing in

the preparation state and the retraction state. On the other hand, the hand/arm edges are much more discontinuous than the head/body edges because the motion of hand/arm is very large during these two states, as shown in Fig. 4.

As the matching block in Fig. 5, the object model of the difference image in the previous preparation state is matched against the object model x of the difference image in the current retraction state. Matching is performed on edge points because it is computationally efficient and insensitive to the changes in illumination. It is also a natural choice given that binary models represent signer in this paper. Notes that e is the representative of the signer for the previous difference image, and our goal in signer model tracking phase is to find its position in the current difference image. To this end, the Hausdorff distance is chosen as matching criterion [10]. It is a simple but powerful measure for comparing binary images or positions. It is very robust when signers are partially occluded, rotated, or changed with their shape. The Hausdorff distance $H(O, R)$ between the objective model O and the reference model R is illustrated in Fig. 6 and given by

$$H(O, R) = \max \{h(O, R), h(R, O)\} \quad (3)$$

with

$$h(O, R) = \max_{o \in O} \min_{r \in R} \|o - r\| \quad (4)$$

and

$$h(R, O) = \max_{r \in R} \min_{o \in O} \|r - o\| \quad (5)$$

Thus, for every model point $o \in O$, the distance to the nearest edge pixel $r \in R$ is calculated, and the maximum value is assigned to $h(O, R)$. Similarly, for each edge point $r \in R$, the distance to the nearest model point $o \in O$ is computed and $h(R, O)$ is set to the maximum distance. Finally, the Hausdorff distance is the larger of the two ones, and every point in the objective model O must be within the obtained Hausdorff distance of some point in the reference model R .

A shortcoming of the definition in (4) and (5) is the large impact that outlying model points will lead to a large Hausdorff distance, even if all other points perfectly are matched. Therefore, it is preferable to use the generalized Hausdorff distance [10], as it does not suffer from this problem. Instead of using the maximum value in (4), the distances are sorted in ascending order and the k -th value is chosen. Similarly, the l -th value of the ordered distances is selected in (5). With the parameters k and l , we can choose how many model points are the near edge points and vice versa. The best match is now found by minimizing the Hausdorff distance between the objective model O and the reference model R for all translations of the reference model relative to the objective model. The smallest distance indicates the new position of the tracked edge point. Several suggestions for efficient implementation are presented in [10]. The main idea assumes that the Hausdorff distance is smaller than a specific threshold so that bad matches can be ruled out early. Obviously, matches can only be found if the Hausdorff distance is indeed smaller than the specific threshold.

The signer model update process is illustrated in Fig. 5. We first apply motion history analysis to find retraction and preparation states. When preparation state is determined, the first difference image in that state is applied to Canny operator to obtain difference edges and keep the resulted difference edges DE_p as the object model for the preparation state. Similarly, the same procedure is applied for the retraction state when it is determined, and its resulted difference edges DE_r are the object model for the retraction state. While both two difference edges are determined, we then construct a signer model

by tracking these two moving edges using generalized Hausdorff distance, i.e.

$$SE = \left\{ o \in DE_r \mid \min_{r \in DE_p} \|o - r\| \leq T_s \right\} \quad (6)$$

where T_s is the threshold value for the signer model. For the next gesture, the same procedure is used to update the signer model.

3.2. Object model tracking phase. This paper models a gesture as preparation, retraction and nucleus state in turn, since a gesture is a series of postures over a time span connected by motion. During the preparation state and the retraction state, the signer model is constructed by tracking both difference edges in the preparation state and the retraction states. After reconstructing signer model, the gesture model can be extracted by matching the object model and signer model in the nucleus state.

The object model tracking procedure is active when the nucleus state is determined. As the block diagram shown in Fig. 5, we first use Canny operator to extract the current edge E_t from the current frame I_t and the difference edge DE_t from difference image. The set of ME_t presents the object model in the t -th frame. The points in ME_t are edges of the moving objects. If DE_t denotes the set of all pixels belonged to the moving edge from the difference image, the object model generated by moving edge is given by selecting all edge pixels within a small distance T_c of DE and denoted as ME^{moving} , i.e.

$$ME_t^{moving} = \left\{ o \in E_t \mid \min_{r \in DE_t} \|o - r\| \leq T_c \right\} \quad (7)$$

In addition, a previous frames moving edges are referred to detect still edges in the current frame. These still edges are belonged to moving edges and denoted as ME^{still} i.e.

$$ME_t^{still} = \left\{ o \in E_t \mid \min_{r \in ME_{t-1}} \|o - r\| \leq T_s \right\} \quad (8)$$

The final edge model ME_t is for current frame. It is expressed by combining the ME^{moving} and ME^{still}

$$ME_t = \{ ME_t^{moving} \cup ME_t^{still} \} \quad (9)$$

3.3. Gesture model tracking phase. The detected ME_t consists of two parts. One is the edges belonged to gesture and the other is edges belonged to body/head of the signer. In order to extract gesture region, the gesture edges and the body edges need to be separated. Based on the SE model constructed before, the body edges in the ME_t are separated by selecting all edge pixels within a small distance T_b of SE , i.e.

$$BE_t = \left\{ o \in ME_t \mid \min_{r \in SE} \|o - r\| \leq T_b \right\} \quad (10)$$

Then we can obtain the gesture model GE by

$$GE_t = \{ o \in ME_t \mid o \notin BE_t \} \quad (11)$$

With the GE_t be detected, the gesture is ready to segment. The procedure [13] that extracts relevant regions to edges is employed to extract the gesture region by the GE_t . The horizontal candidates are declared as the region inside the first and last edge points in each row and the vertical candidates for each column. After finding both horizontal and vertical gesture candidates, intersection region through logical AND operation are further processed by morphological operations [11].

4. EXPERIMENTAL RESULTS. We use a video database of i6-Gesture database [12] and MH sequence. All image sequences in i6-Gesture database have been recorded by two different view, hand view and body views. Sequence MH is a home-made gesture sequence. Fig. 7 shows the value of motion information for each frame of a sequence in i6-Gesture database, and the first frame of each state detected by motion history analysis. The segmented gesture from the first frame of the nucleus state is also obtained. Note that the motion information of the segmented frame close to the local minimum is the most representative gesture than others in preparation and retraction state. The motion information of the hand view shown in left figure is more stable than the body view, where the only local minimum falls in nucleus state. The body view has more than one local minimum, where one is in nucleus state and the other caused by illumination change is in preparation state. Despite illumination change happened, the proposed motion history analysis classifies each inputted frame correctly. Besides, the computation load is also reduced, since our algorithm only segments gesture during the nucleus state. Fig. 8 shows the value of motion information for each frame of MH sequence, and the first frames of two nucleus states and a fleet nucleus state detected by motion history analysis. The fleet nucleus state involves that the transition from the current gesture to the next gesture. Despite a local minimum of motion information is detected, the proposed motion history analysis can determine it as a transitional gesture and avoid it to be considered as the representative gesture.

Fig 9 shows three gestures extraction results on MH sequence. The original frames associated with each extracted gesture are shown in first column. The current edge image of each original frame, the difference edges determined as gesture edges by the signer model, and the extracted gestures are shown in second, third, and fourth column respectively. In the initial two gestures, the signer model contains disturbances which contain some part of edges of moving head. In the third gesture, the signer model contains only few edges of moving head. This is because the TPMT can track the head/body edges while excluding the hand/arm edges. Therefore, the signer model will become robust after a short initial period. Also note that the gesture images are successfully extracted despite that the difference edges for gesture are disturbed. This is because most of edges of head are discontinuous, the formed intersection regions are small and removed by morphological operations.

Numerical evaluation is usually done in terms of the number of *false negatives* (the number of gesture pixels that were missed) and *false positives* (the number of other pixels that were marked as gesture). The ground truth is achieved by manually labeling some frames from the video sequence. The gesture extraction results of Lab1_1, Lab1_2, Class_1, Class_2, Lab2_1, and Lab2_2 are shown in first, second, third, fourth, fifth, and sixth row respectively in Fig. 10. The frame size of hand view is 320×240 , the frame size of body view 352×288 . Each gesture comprises a different background. The first four columns show frames contained in video sequence, the fifth column shows segmented results of the fourth column. The numerical evaluation results are given in Table 1. The error rate of frame is tested by $FN + FP/framesize$. It is seen that, the average error rate is below 2%.

All experiments are performed on a Pentium IV 1.8 GHz PC. The run time performance of the proposed method is about 0.03msec per frame for 320×240 frame size and 0.028msec per frame for 352×288 frame size. This performance is applicable on real-time HCI applications. Run time analysis show that Canny operator takes nearly 66% of the whole run time, and Gaussian smooth is the major contributor. Model tracking takes 21% of the whole run time. The motion history analysis takes the least computation load, which

takes 7% of the whole run time. This makes the method well-suited to systems with require real-time processing requirement.

5. **CONCLUSION.** This paper proposes a TPMP approach to automatically construct three different models without prior information. The TPMP approach consists of two main components. The motion history analysis classifies inputted gesture frame into preparation, retraction and nucleus state based on a history of motion information. The model tracking component tracks signer model and object model with different constraint based on the classified state. When the nucleus state is detected, the object model and the signer model are matched to generate gesture model and segmented hand region from gesture model. The experiment results show that our proposed algorithm can perform well in the real scene. The run time analysis demonstrates that the proposed algorithm is usable on real time HCI applications.

REFERENCES

- [1] V. I. Pavlovic, R. Sharma and T. S. Huang, Visual interpretation of hand gestures for humancomputer interaction: A review, *IEEE Trans. Pattern Analy and Mach*, vol. 19, pp. 677-695, 1997.
- [2] Y. Wu and T. S. Huang, Vision-based gesture recognition: a review, *Proc. of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, vol. 1739, pp. 103-115, 1999.
- [3] A. Corradini and H. M. Gross, Camera-based gesture recognition for robot control, *Proc. of the IEEE International Conference on Neural Networks*, vol. IV, pp. 133-138, 2000.
- [4] Y. Yuan, Y. Liu, and K. Barner, Tactile gesture recognition for people with disabilities, *Proc. of the IEEE 30th International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 461-464, 2005.
- [5] R. Yang and S. Sarkar, Gesture Recognition using Hidden Markov Models from Fragmented Observations, *Proc. of the IEEE 19th Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 766-773, 2006.
- [6] M. H. Yang, N. Ahuja, and M. Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition, *IEEE Trans. Pattern Analy and Mach*, vol. 24, pp. 1601-1074, 2002.
- [7] R. Grzeszczuk, G. Bradski, M. H. Chu and J. Y. Bouguet, Stereo based gesture recognition invariant to 3D pose and lighting, *Proc. of the IEEE 13th Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 826-833, 2000.
- [8] N. Habili, C. C. Lim and A. Moini, Segmentation of the face and hands in sign Language video sequences using color and motion cues, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 8, pp. 1086-1097, 2004.
- [9] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679-698, 1986.
- [10] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, Comparing images using the Hausdorff distance, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850-863, 1993.
- [11] L. Vincent, Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms, *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 176-201, 1993.
- [12] <http://wasserstoff.informatik.rwth-aachen.de/dreuw/database.html>
- [13] C. Kim and J. N. Hwang, Fast and automatic video object segmentation and tracking for content-based applications, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 122-129, 2002.
- [14] M. R. Dobie and P. H. Lewis, Object tracking in multimedia systems, *Proc. of the IEE 4th International Conference on Image Processing and Its Applications*, pp. 41-44, 1992.

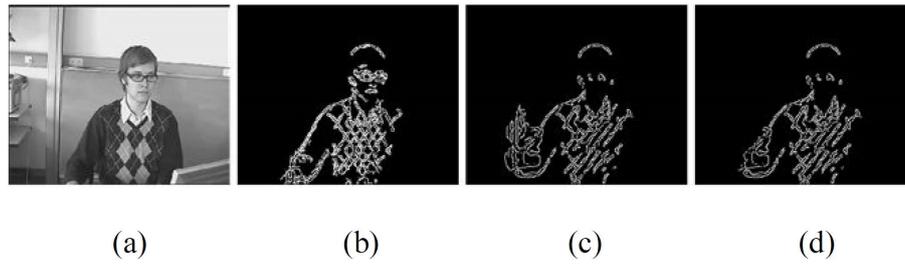


FIGURE 1. Two-dimensional binary signer model construction. (a) signer image. (b), (c) moving edge points derived from different time install. (d) the constructed signer model points using (b) and (c).

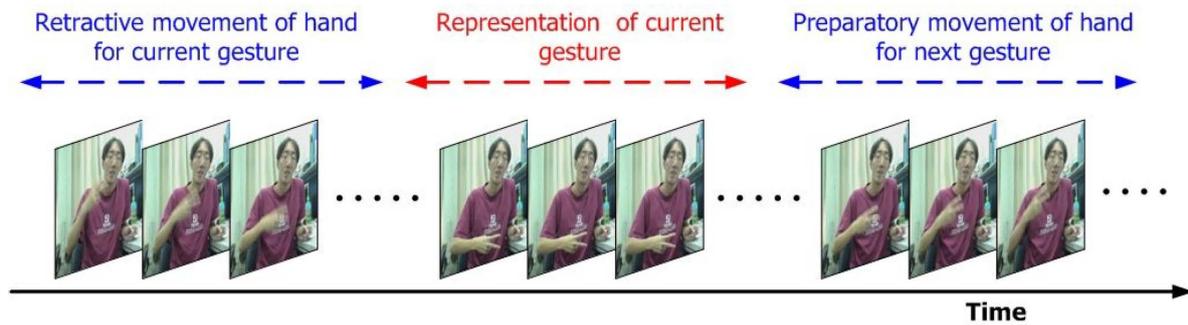


FIGURE 2. An example of dynamical representation of a gesture.

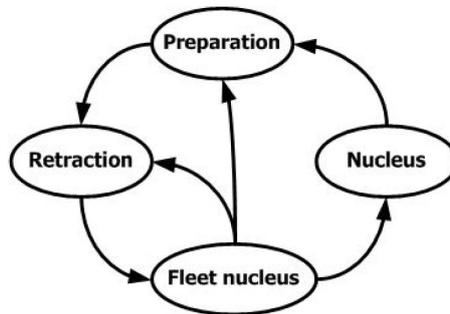


FIGURE 3. A four gesture states transition diagram, that model the gesture state as preparation, retraction, fleet nucleus and nucleus state.



FIGURE 4. Two frames and its difference image in the preparation state and the retraction state.

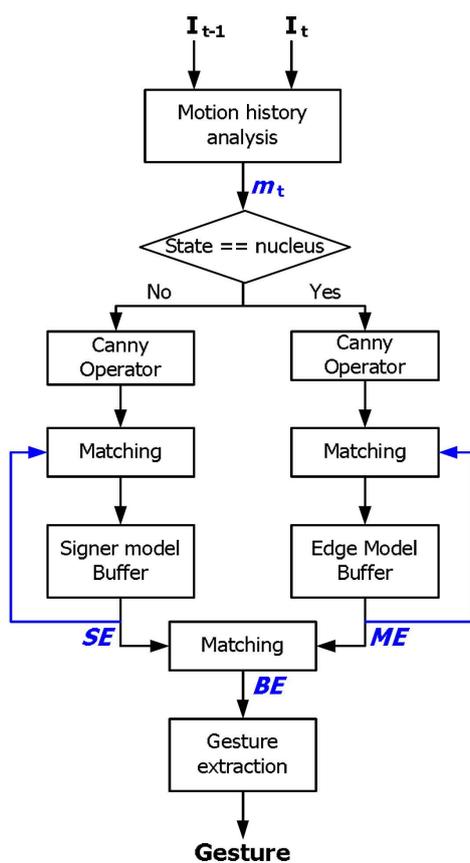


FIGURE 5. Block diagram of the proposed gesture extraction algorithm.

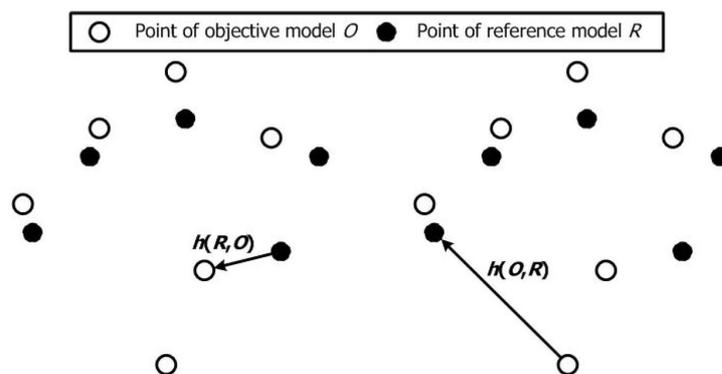


FIGURE 6. Definition of Hausdorff distance.

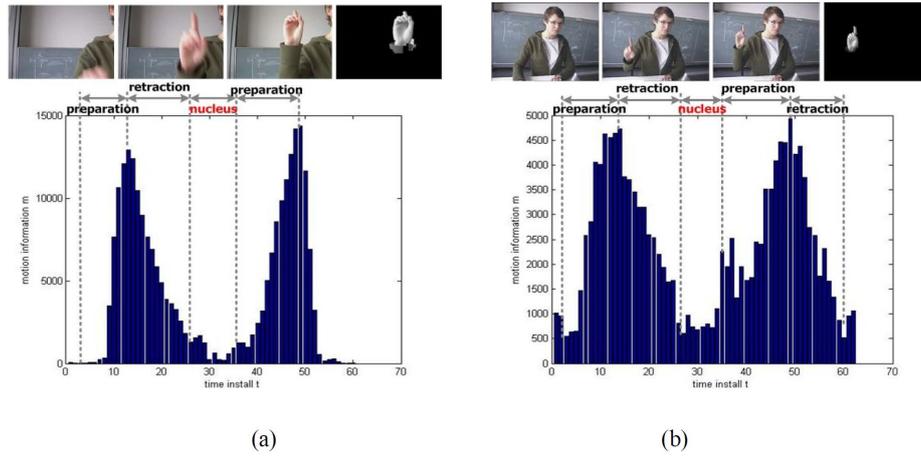


FIGURE 7. The histogram of the motion information for each frame. The first frame corresponding to each state and the extracted gesture of the first frame of the nucleus state is shown in upper row.

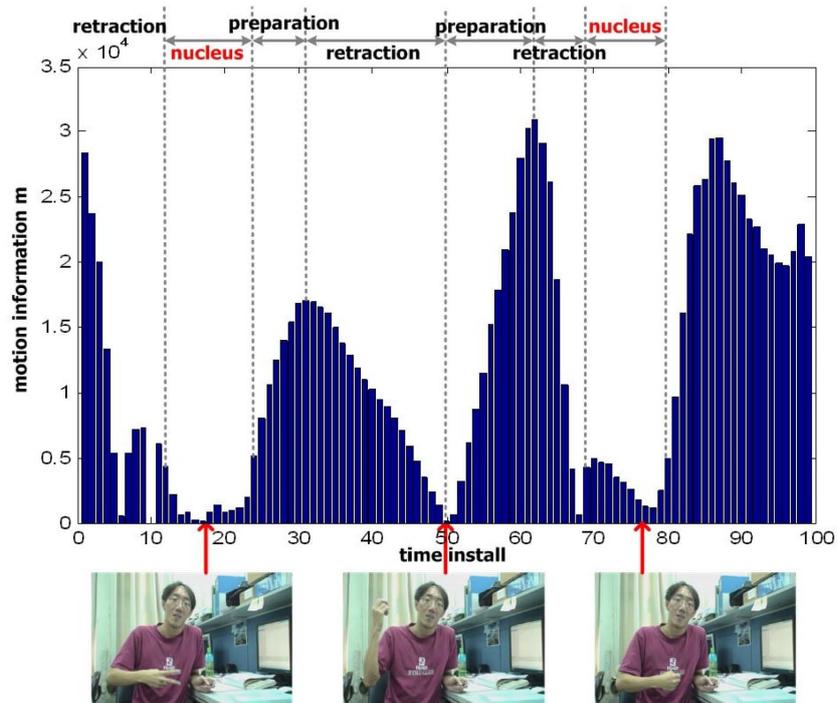


FIGURE 8. The history of the motion information for each frame in HM sequence.



FIGURE 9. HM sequence and its gesture extraction result. The first column shows three frames in nucleus state. The current edge image, the difference edges for gesture, and the extracted gestures are shown in second, third, and fourth column respectively.



FIGURE 10. Image sequences in i6-Gesture database and its gesture extraction results.

TABLE 1
Numerical evaluation results of the method presented in this paper.

Test Sequence	FN	FP	FN+FP	Error rate
Lab1_1	320	495	815	1.06%
Lab1_2	113	107	220	0.21%
Class_1	649	951	1600	2.08%
Class_2	350	118	468	0.46%
Lab2_1	621	172	793	1.03%
Lab2_2	174	300	474	0.46%
HM	805	386	1191	1.55%