A Real-Time Gesture Tracking and Recognition System Based on Particle Filtering and Ada-boosting Techniques

Chin-Shyurng Fahn[1], Chih-Wei Huang[1], and Hung-Kuang Chen[2]

[1] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.
csfahn@csie.ntust.edu.tw
[2] Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C.

Abstract. A real-time gesture tracking and recognition system based on particle filtering and Ada-Boosting techniques is presented in this paper. The particle filter, which is a flexible simulation-based method and suitable for non-linear tracking problems, is adopted to achieve hand tracking robustly. In order to avoid the influence of the other exposed skin parts of a human body and skin-colored objects in the background, our system further applies the motion information as a feature of the hand in addition to the skin color information. Compared with the conventional particle filters, our method leads to more efficient sampling and requires fewer particles. It results in lowering computational cost and saving much time for gesture recognition later. The gesture recognition uses the features derived from the wavelet transform, and employs an Ada-Boost algorithm which is excellent in facilitating the speed of convergence during the training. Hence, it is conducive to update new information and expand new gesture archives. The experimental results reveal our system is fast, accurate, and robust in hand tracking. Moreover, it has good performance in gesture recognition under complicated environments.

1 Introduction

With the fast development of computer technology in recent years, the interaction between humans and computers is becoming more and more important. Nowadays, the keyboard and mouse are standard equipments for people to interact with a computer. Nonetheless, this kind of interfaces restricts the communication between humans and computers to the manipulation of the devices. Alternatively, new Human Computer Interface (HCI) technology aiming at more rich, natural, and efficient ways of interactions between humans and computers appears in form of face tracking [1], hand tracking [2],[3], face recognition, gesture recognition [4], and so on, in these years. Of such new technology, gesture analysis has been established as one of the most important perceptual interfaces. As a key step of gesture analysis, the hand tracking, especially by monocular vision, is crucial to the success of the deployment of the vision-based user interface in new generation HCI systems.

C. Stephanidis (Ed.): Universal Access in HCI, Part II, HCII 2007, LNCS 4555, pp. 818–827, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A real-time gesture tracking and recognition system based on particle filtering and Ada-Boosting techniques is presented in this paper where a flow chart of this system is shown in Figure 1.

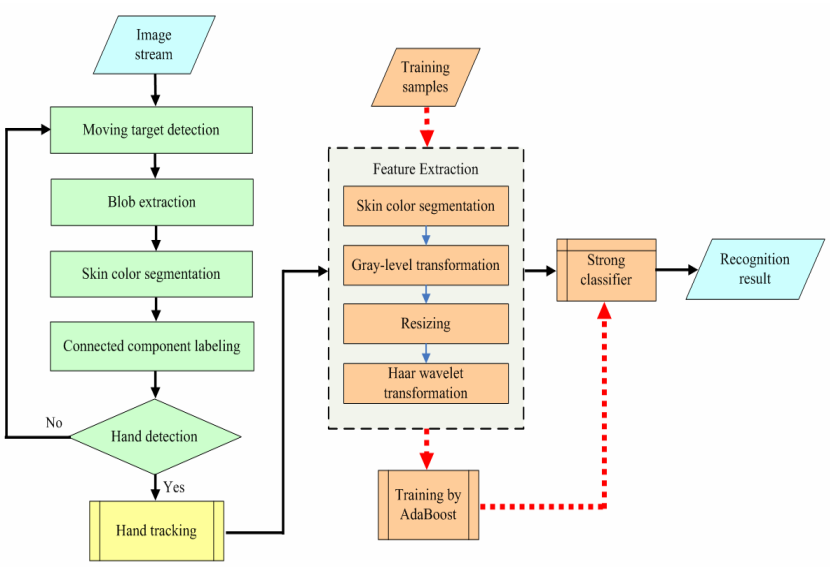

Fig. 1. The flow chart of our system

The particle filter, a flexible simulation-based method suitable for non-linear tracking problems, is adopted to achieve robust hand tracking [3], [5]. In order to avoid the influence of the other exposed skin parts of a human body as well as skin-colored objects in the background, our system makes use of the motion information as a feature of the hand in addition to the skin color information. Compared with the conventional particle filters, our method has a more efficient sampling scheme that requires fewer particles. It effectively reduces the computational cost of the tracking stage and implies more efficient gesture recognition.

The gesture recognition uses the features derived from the Harr wavelet transforms [6] and employs the Gentle Ada-Boost algorithm [7] which is excellent in facilitating the speed of convergence during the training. A great number of existing gesture recognition systems provides fine classification results. However, most of them need high-resolution image sequences and good segmentation of hand regions. Compared to these systems, our recognition system is able to achieve nice recognition results from more complex environments. Furthermore, the Ada-Boost algorithm for gesture recognition requires much less training time than the other machine learning methods such as the hidden Markov model and neural network to perform the same accuracy. Therefore, new gestures or training samples can be introduced or updated with much less efforts for various situations.

The discussion of hand tacking will be presented in Section 2 followed by the gesture recognition in Section 3. Some experimental results are shown in Section 4. Finally, the conclusions and remarks are given in Section 5.

2 Hand Tracking

In order to decrease the calculation time and improve the system performance, we can reduce the searching area while extracting hand regions by focusing on moving regions in a scene. If a moving region larger than a predefined size is detected, a color segmentation scheme to separate skin color regions from the moving region is conducted. Subsequently, we apply a connected component labeling method to acquire an exact area and determine wither it is a hand or not.

2.1 Hand Detection

The moving target detection of our proposed method is based on temporal differencing which is very adaptive for dynamic environments and need not establish background images. The moving target in a scene will be displayed as black pixels and segmented from the background by letting

$$NI_n(x,y)=\begin{cases}255 & if\ |I(x,y,n)-I(x,y,n-1)|\geq\varepsilon\\ 0 & \text{otherwise}\end{cases}$$

where n is the frame number, $I(x,y,n)$ is the pixel value of point (x,y) at the current frame, $I(x,y,n-1)$ is the pixel value at the previous frame, and ε is the threshold value acquired from experiments.

We subtract the current frame from the previous frame to generate the difference image using the motion analysis method proposed in [8]. This method is to compute the absolute value of the differences in the neighborhood (we use a mask) surrounding each pixel, then accumulates the difference value of each pixel in the mask. If the accumulated difference value exceeds a predetermined threshold, this pixel is assigned to a moving region.

With the motion analysis method, we can obtain the area of a moving target. To divide the hand region from the moving target, a skin color segmentation method is employed. First, we adopt the HSV color model to replace the RGB one. To aim at the extraction of skin color regions, we segment skin colors from others by H values which is ranged from 3 to 38. It is accomplished through the given upper and lower thresholds to discriminate skin color regions from non-skin ones. After segmenting the skin color region from an extracted moving target, we apply a connected component labeling method to obtain a more accurate area [9].

2.2 Tracking Scheme

The detection algorithm is used to find the complete hand region in the first frame; nonetheless, it is a time-consuming task. After detecting the hand region, we exploit the tracking process to surmount this problem. Our tracking method is activated after the initial location of a target is specified. On the other hand, the majority of the conventional methods employing detection together with tracking adopt a detect-then-track approach where the target is detected in the first frame and then turned to the tracker in the subsequent frames. A well-designed tracking scheme not only saves us a lot of calculation time but extracts the target from an image sequence precisely. To continuously understand the motion direction and trajectory of the target is the main

purpose of tracking. Our system utilizes the particle filtering technique that is modified from [3] and [5] to implement the tracking algorithm.

The particle filter is one of flexible simulation-based methods for sampling from a sequence of probability distributions. The strength of these methods lies in their simplicity, flexibility, and systematic treatment of nonlinearity. Moreover, it can deal with extreme non-Gaussian and multimodal noise distributions. The particle filter approximates the sequence of probability distributions of interest using a large set of random samples, named particles. These particles are propagated over time using simple Importance Sampling and re-sampling mechanisms. The iterative process of the particle filter used in our system is summarized in Figure 2.

Given a sample set $S_{t-1} = \left\{\left(s_{t-1}^{(i)}, P_{t-1}^{(i)}\right) | \ j = 1,2,\cdots,n\right\}$ at time step $t-1$, perform the following steps:

1. **Propagating:** propagate each sample $s_{t-1}^{(i)}$ with probability $P_{t-1}^{(i)}$ by a dynamic model to obtain the sample set S_t.
2. **Weighting:** weigh the samples using both color and motion cues as:
   1) Calculate the distance between each sample and the target by
   $$Dis = \sqrt{1 - \frac{M}{M_t}}$$
   where M is a linear combination of the measurement of color and motion cues; M_t is the size of the sample.
   2) Weigh each sample with
   $$\pi^{(i)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{Dis^2}{2\sigma^2}}$$
   where σ is the standard deviation of a Gaussian distribution.
3. **Selective re-sampling:** if $\pi^{(i)}$ exceeds or equals a threshold, select C_i even-weighted samples.
4. **Estimate:** obtain the state of the sample with the largest weight
   $$s_t^{(j)}, \ j = \arg\max_i \pi_t^{(i)}, i = 1,2,\cdots,n$$

Fig. 2. An iterative process of a particle filter

3 Gesture Recognition

We adopt the Ada-Boosting technique to train our samples whose sub-band information in the frequency domain is extracted from the Haar wavelet transform [6] as the features. The classification accuracy will be enhanced with the raise of the number of training samples, the iterations of boosting, and the suitability of features used for texture description. In order to improve the accuracy of classification, some preliminary processes are conducted before the wavelet transform.

During the tracking procedure, the human hand is represented by a rectangular window and the size of the window is changing with the size of the tracked hand. To reduce the influence of the background, the skin color segmentation method is again

applied to extracting the real hand region. The Grey-level transformation is instantly performed after the skin color segmentation. Then we normalize the rectangular windows by resizing them to a fixed size. The smallest size of a hand region is about pixels in our images; thus, we unify all the rectangular windows to this size. The Haar wavelet transform is performed on such a normalized rectangular window to obtain the second level sub-band LL2 as our hand features.

In this paper, we adopt the Gentle Ada-Boost algorithm [7] integrated with the Classification and Regression Trees (CARTs) as a weak classifier and utilize the GML Ada-Boost Matlab Toolbox provided by the Graphics and Media Laboratory [10] for training/constructing our strong classifier. A version of the Gentle Ada-Boost algorithm is given in Figure 3.

Given: training samples $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ for sample vector $x_i \in X$ and label $y_i \in \{-1, 1\}$, weak classification space H, $H_1(x) = 0$, and integer T specifying the number of iterations

Initialize the distribution: $D_1(i) = 1/m$

For $t = 1, 2, \ldots, T$

(1) Fit the regression function $h_t(x)$ by weighted least squares of y_i to x_i with distribution $D_t(i)$

(2) Update $H_{t+1}(x) = H_t(x) + h_t(x)$

(3) Update $D_{t+1}(i) = D_t(i)\exp(-y_i h_i(x_i))$ and renormalize $D_{t+1}(i)$ such that $\sum_{i=1}^{m} D_{t+1}(i) = 1$

Output the final strong classifier: $H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$

Fig. 3. The Gentle Ada-Boost algorithm

We train and construct a unique strong classifier for each hand gesture through the Ada-Boost algorithm. By applying the features that are extracted from the rectangular window to the strong classifier, we can obtain a weight of the final prediction. In order to raise the accuracy of classification, we contemplate the weights of 10 consecutive rectangular windows and calculate the sum of these weights. Comparing the sum of weights for various strong classifiers, we can choose the largest one, and if this sum is greater than a predefined threshold, we think that the user is making a particular hand gesture.

4 Experimental Results and Discussion

In this section, we will implement our proposed method and analyze the experimental results. In order to demonstrate the effectiveness of the algorithms, some video

sequences are tested. All the software of our real-time gesture tracking and recognition system is run on a personal computer using a Pentium 4 3.2 GHz CPU with 1 GB DDRAM. The developing environment includes the Borland C++ Builder 6, MATLAB 7.2, Windows XP Professional SP2 equipped with DirectX 9.0, and the image sequences with the resolution of 320 × 240 pixels. The throughput obtained is from 20 to 25 frames per second.

The accuracy measures used for gesture tracking are based on the number of tracks in the reference data set as well as the number of tracks reported by the system. The notations and definitions of measuring our system performance are mc = A'/A, mf = R'/A, and mm = 1-R/A, which respectively symbolize the fraction of correct tracks, the fraction of reported tracks that are false alarms in which they do not correspond to true tracks, and the miss rate that should be zero for an ideal system.

In order to simplify tracking, we assume that there are no other exposed skin body parts of the same hand which is to be tracked. Moreover, we make the movement of the right hand is faster than other exposed skin body parts, especially than the head. Furthermore, we broadly classify the testing image sequences into three types: general, complicated, and speedy samples and label them as Types A, B, and C, respectively.

4.1 Type A: General Samples

We regard a video sample as a general one if the background of this sample is uncomplicated, the motion of a tracking target is evident, the environment is simple and under sufficient illuminant. For each testing video, we repeatedly execute it 10 times and record the experimental results. There are 11 video samples considered as a simple situation. Table 1 records the miss rates as well as the worst, average, and the best accurate rates of tracking for these 11 samples. The average accuracy of this type is better than 96% and even reaches 100%. Notice that all the experimental data are taken down by calculating the average values for each video sample.

Table 1. The Tracking Accuracy of General Samples

Measurement	Sample										
	A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8	A-9	A-10	A-11
Avg. m_f (%)	0.6	2.5	2.0	0.2	2.6	0.0	2.2	0.0	2.1	2.7	0.0
Avg. m_m (%)	0.2	1.1	0.1	0.2	1.4	0.0	0.1	0.0	0.0	0.1	0.0
Worst m_c (%)	98.0	88.8	91.5	99.0	90.0	100	94.9	99.6	95.8	95.8	100
Avg. m_c (%)	99.2	96.2	97.9	99.7	96.1	100	97.7	99.9	97.9	97.2	100
Best m_c (%)	100	99.1	100	100	99.2	100	99.5	100	99.6	99.1	100

4.2 Type B: Complicated Samples

We consider a testing video as a complicated sample if too many factors interfere with the tracking result; for instance, the hand is disturbed by human faces or

skin-colored objects in the background, the hand region is too small, and the illuminant is not enough. We have 14 complicated samples and their experimental results are listed in Table 2.

Table 2. The Tracking Accuracy of Complicated Samples

Measurement	Sample													
	B-1	B-2	B-3	B-4	B-5	B-6	B-7	B-8	B-9	B-10	B-11	B-12	B-13	B-14
Avg. m_f (%)	3.1	5.6	3.9	1.8	5.4	6.9	4.0	2.4	5.2	2.8	5.6	1.5	5.2	2.5
Avg. m_m (%)	1.8	11.1	0.0	13.2	4.7	2.2	6.6	8.0	0.6	7.2	13.8	4.5	12.4	8.4
Worst m_c (%)	93.7	73.5	94.4	81.8	81.5	89.0	86.4	81.1	91.8	79.8	76.7	88.7	55.3	87.2
Avg. m_c (%)	95.0	83.4	96.1	84.9	90.0	90.9	89.5	89.6	94.2	90.0	80.6	94.0	82.4	89.1
Best m_c (%)	96.5	93.2	97.3	88.1	95.4	92.7	94.6	94.6	96.2	97.2	83.5	99.1	96.4	92.8

In this type of samples, the errors of tracking hands are usually caused by face disturbance. It is obvious that a human face is an integral skin color region and its size is often larger than that of a hand. Hence, when the hand moves near the face, the rectangular window tends to locate the target on the face region. But, if the hand continuously moves, the rectangular window constantly returns to the right position due to the effect of the motion information. Notice that the miss rate m_m becomes large in this type. It is because we set a threshold for the weight of the rectangular window, so our system will re-detect the hand region if the weight is less than the threshold. The re-detecting procedure could avoid unnecessary wrong tracks and diminish the false rate m_f, but a frequently re-detecting actions would lead to the increase of m_m. Consequently, this threshold should be determined properly.

4.3 Type C: Speedy Samples

It will raise the tracking difficulty if the hand moves quickly. In our system, the average frame number per second is from 20 to 25, and if the hand movement range is over 100 pixels length (around one third width of a scene) in 10 consecutive frames, we identify this sample as a speedy one.

In general, the tracking accuracy of speedy samples is lower than those of the other two types. The experimental results are shown in Table 3. When the position of a hand changes too much in a short period, it usually exceeds the re-sampling scope of a particle filter. We can see that the differences between the worst and the best correct rates m_c in Samples C-2, C-3, and C-6 are very large. It is because the hand overlaps the human face or skin-colored objects and suddenly moves away at a high speed, while the particle candidates are slow in reacting and miss the hand region easily. Our system can keep tracking well if any one of the particle candidates just hits on the hand region. On the contrary, if the particle candidates do not catch the correct target, it will either abandon the tracking target then try to detect the hand region once more or keep tracking on the wrong target until the hand moves near the re-sampling scope

of the particle filter. Attempting to enlarge the re-sampling range could slightly solve this problem. Nevertheless, the wider the re-sampling range is, the sparser the particle candidates are, and sometimes it would reduce the tracking precision.

Table 3. The Tracking Accuracy of Speedy Samples

Measurement	***Sample***						
	C-1	C-2	C-3	C-4	C-5	C-6	C-7
Avg. m_f (%)	1.0	10.0	4.0	10.8	11.0	8.8	2.0
Avg. m_m (%)	5.0	17.9	9.5	9.4	6.5	10.0	19.5
Worst m_c (%)	91.2	56.5	49.0	70.8	78.8	49.2	72.8
Avg. m_c (%)	94.0	72.2	86.6	79.9	82.5	81.3	78.6
Best m_c (%)	97.0	87.4	94.3	88.7	87.4	88.0	83.2

We compare the efficiency of the conventional [5], particle filter with mean shift [3], and our proposed methods. And the experimental results are recorded in Table 4.

Table 4. Comparison of the Three Hand Tracking Methods

Measurement	Method		
	Convention	Mean shift	Ours
Avg. execution time	47ms	25ms	17ms
Avg. frame number	21	39	58
Number of particles	100	20	20
Avg. accuracy (Type A)	91.7%	98.6%	98.3%
Avg. accuracy (Type B)	86.5%	91.0%	90.3%

Table 5 shows the average execution time of the mean shift method and our proposed method with different number of particles.

Table 5. Average Execution Time of the Mean Shift Method and Our Proposed Method

Number of particles	Avg. execution time (ms)	
	Mean shift	Ours
20	25	17
40	36	28
60	48	33

Our system can recognize three kinds of gestures: the palm, thumb, and victory gestures as Figure 4 illustrates. The positive and negative training samples are obtained manually from video sequences. Furthermore, in order to raise the variety of training samples, we also take the hand pictures as our samples directly. We totally have 2,100 training samples, and each kind of gestures includes approximately equal

number of samples. The evaluation method of gesture recognition is to measure the accuracy of classifying the three particular gestures. We execute the gesture recognition procedure for each testing video 5 times to acquire more objective results and record the estimative effects for different gestures.

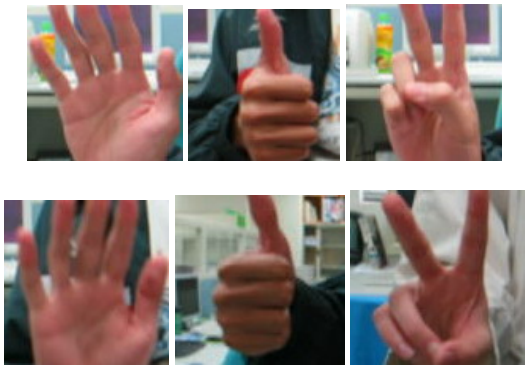

Fig. 4. Some samples of the palm, thumb, and victory gestures

Before we start training, a K-fold cross-validation procedure is applied. We adopt 5-fold cross-validation to estimate the accuracy of different system models in our experiments. From the model estimation, we find that the best performance is achieved while the Gentle Ada-Boost algorithm uses 24 CART splits and 500 iterations. In consideration of the tradeoff between accuracy and efficiency, we exploit the Gentle Ada-Boost algorithm with 24 CART splits and 500 iterations as our classification model.

The average recognition rates of the three kinds of gestures are shown in Table 6. As it can be seen from this table, the accuracy of the thumb gesture is much better than those of the other two. Comparing with the other two kinds of gestures, the shape of the thumb gesture is more stable and effortlessly to be recognized.

Table 6. Average Recognition Rates of the Three Kinds of Gestures

Accuracy	Gesture		
	Palm	Thumb	Victory
Average recognition rate (%)	80.9	96.9	81.4

5 Conclusions and Future Work

In this paper, we have presented a robust and efficient real-time gesture tracking and recognition system. The current system takes approximately 0.05-0.06 second a frame to complete all processes, including hand tracking and gesture recognition. During the tracking procedure, our method can use fewer particles to maintain exact hand tracking than the conventional particle filtering methods do, and results in low computational cost. Our tracking system is very robustly and precisely, which is not only suitable for tracking the hand behind a shelter or suffering from head

disturbance, but also allows the slight oscillation of a scene. By employing a predefined threshold in the re-sampling step, our tracking system overcomes the degeneracy problem, saves much calculation time, and enhances the tracking accuracy.

The above computer vision based gesture tracking and recognition system is still far from achieving an ideal state of intelligence and flexibility. Although we have already proposed an acceptable system model, several parts require further improvement. At present, the observation phase of our tracking algorithm only uses color and motion cues. Some other cues, for example, texture features and the relative location of hands, can also be utilized, so that we can differentiate two hands from a head. Besides, the gesture recognition results are not good enough yet. Future work tends to improve the recognition rate of dynamic gestures by incorporating more features such as the contour, orientation, speed, and trajectory information as well as training with more and larger variety of gesture samples.

References

1. Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal 2(2), 1–15 (1998)
2. Imagawa, K., Lu, S., Igi, S.: Color-based hands tracking system for sign language recognition. In: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 462–467 (1998)
3. Shan, C., Wei, Y., Tan, T., Ojardias, F.: Real time hand tracking by combining particle filtering and mean shift. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 669–674 (2004)
4. Liu, X., Fujimura, K.: Hand gesture recognition using depth data. In: Proceedings of the 6th IEEE Intern. Conf. on Automatic Face and Gesture Recognition, pp. 529–534 (2004)
5. Song, X.Q.: Real-time visual detection and tracking of multiple moving objects based on particle filtering techniques, Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC (2005)
6. Kumar, S., Kumar, D.K., Sharma, A., McLachlan, N.: Classification of visual hand movements using multiresolution wavelet images. In: Proceedings of the International Conference on Intelligent Sensing and Information Processing, pp. 373–378 (2004)
7. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. The. Annals of Statistics 28(2), 337–407 (2000)
8. Graf, H.P., Cosatto, E., Gibbon, D., Kocheisen, M., Petajan, E.: Multi-modal system for locating heads and faces. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 88–93 (1996)
9. Suzuki, K., Horiba, I., Sugie, N.: Linear-time connected-component labeling based on sequential local operations. Computer Vision and Image Understanding Archive 89(1), 1–23 (2003)
10. Vezhnevets, A.: GML Ada-Boost Matlab Toolbox, Technique Manual, Graphics and Media Laboratory, Computer Science Department, Moscow State University, Moscow, Russian Federation (2006) http://research.graphicon.ru/