3D Hand Motion Tracking Using Improved Hidden Markov Model of Behavior

Wei Gai
School of Information Science and Engineering, University of Jinan
Shandong Provincial Key Laboratory of Network based Intelligent Computing
Shandong, 250022, China
Email: gaiwei1987@126.com

Zhiquan Feng*, Bo Yang, Yan Lin and Xianhui Song
School of Information Science and Engineering, University of Jinan
Shandong Provincial Key Laboratory of Network based Intelligent Computing
Shandong, 250022, China
Email: fzqwww@263.net

***Abstract*—We present a novel approach for 3D hand motion tracking of stage characteristics as observed in cognitive psychology. A Hidden Markov Model acquired in human-computer interaction is used to represent the high-level structure of gestures reduce the dimensionality of the search space. The estimation of hand gestures is handled by combing the model and particle filter. The sampling is introduced PERM instead of standard re-sampling to compensate the error caused by the observation model. The simulated experiment demonstrates significant improvements in tracking speed and robustness over comparison methods.**

***Index Terms*—Hidden Markov Model, cognitive psychology, PERM, human hand tracking**

I. INTRODUCTION

Hand is a kind of multi-joint and non-rigid object, and hand tracking is a typically nonlinear and non-Gaussian filtering problem. Many researchers home and abroad have already made many different methods for this problem, which can attribute to appearance-based and model-based approach generally [1-4]. The first one [5-8] is mainly based on characterization that can be divided into region-based, active contour-based and other image feature-based tracking methods. The other [9-14] adopts usually fixed object models.

At present, the most common approach is particle filter (PF), and it's a random sampling filter approach that is another realization [15-16] of recursive Bayesian filter developed after the mid-90s of last century. Its main idea is to describe the probability distribution of random samples called "particle", and then approximate the actual probability distribution by adjusting the size of particle weights and the location of samples. However, this method needs to sample a large number of particles. For example, even if assuming it only uses ten variables, and each variable takes twenty discrete values, this method needs $3.2*10^8$ samples at least, which makes real-time tracking become hopelessly out of reach. Therefore, reducing the numbers of particles and time cost become one of the cores of particle filter.

Through learning cognitive psychology [17], we exploit the fact that hand motion has phase feature. When using particle filter, if each variable of gesture at each stage is still sampled, the time cost would increase greatly. In order to solve the problem, we use the method of machine learning [18-20], and analyze a specific virtual assembly system to decompose the state space, and then establish the Hidden Markov Model. For different states, we sample selectively instead of completely. Meanwhile, we chose PERM sampling [9, 21] instead of standard sampling that may compensate the error caused by the observation model.

II. HIDDEN MARKOV MODEL

Hidden Markov Model [22] (HMM) was first proposed in 1957. It was successfully applied to the acoustic modeling in the 1980s. In resent years, HMM is applied to the analysis of the volatility of financial markets, economic budget, neurobiology, biological genetics, etc. Recently, HMM is mainly utilized in engineering field, such as image processing, voice synthetic, seismic survey and so on. It achieved scientific significance and value of the important results [23].

Hidden Markov Model is an important method to solve the problem of high dimensional feature space search. Through the model, we may obtain the relationship between high dimensional feature set and gestures. It's a doubly stochastic process. The first stochastic layer is the underlying first-order Markov process; the second is the set of output probabilities for each state. The essence [24] of HMM is to quantify the time (observation) sequence to a small amount of discrete states, and there is a transition probability between any two of the states (including themselves).

The basic elements of Hidden Markov Model as follow [25].

Manuscript received Mar. 25, 2010; revised Apr. 10, 2011.
project number: 60973093, 60873089.
* corresponding author.

doi:10.4304/jcp.7.3.730-735

- The number of states N. S is a set of states, $S = \{S_1, S_2, ... , S_N\}$, and the state at time t is denoted as
  $q_t \in S$, $1 \le t \le T$.
  T is the length of observation sequence.
- The number of observation symbols M. V is the set of all observed symbols,
  $V = \{v_1, v_2, ..., v_M\}$.
- The matrix of state transition probability A:
  $A = \{a_{ij}\}$,
  $a_{ij} = P\{q_{t+1} = S_j \mid q_t = S_i\}$,
  $1 \le i, j \le N$,
  $a_{ij} \ge 0$, $\sum_{j=1}^{N} a_{ij} = 1$.
- The output probability matrix B of observed variables under the condition of state Sj. $B = \{b_i(v), 1 \le i \le N, v \in V\}$.
- The probability distribution in initial state π : $\pi = \{\pi_i, 1 \le i \le N\}$.

III. THE TRACKING ALGORITHM

Firstly, we establish hand motion behavior model, which is Hidden Markov Model, acquired in human-computer interaction. Secondly, we predict the gesture at current time, and we can propagate particles only in plausible directions. Lastly, we use the PERM method to re-sample. For sampling particles, what we need to do is to compare the similarity with actual gesture, and only cope with the samples of maximum and minimum similarity.

The main frame of our algorithm is as follows (Fig.1).

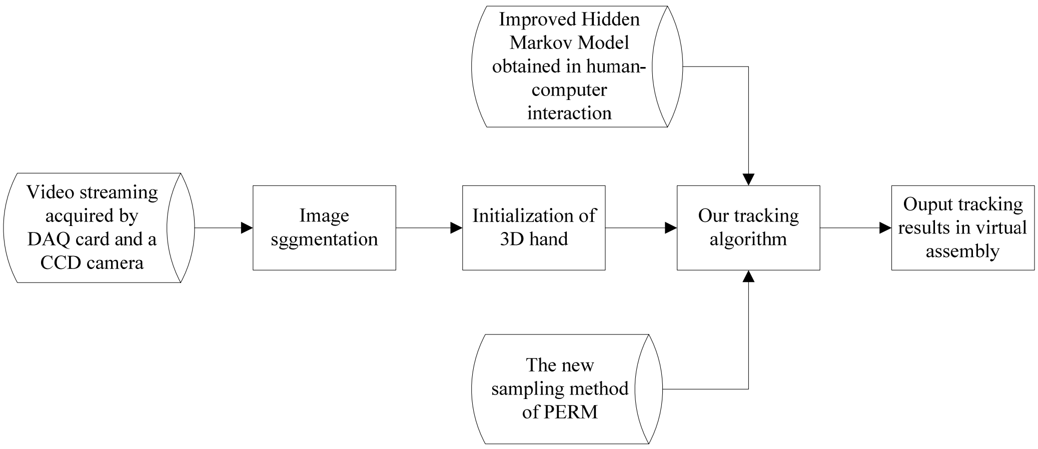


Figure 1. The global framework of our algorithm

A. Cognitive-behavioral model of hand motion

Through observation and analysis of a virtual assembly system, we factor out the three basic operations used frequently.

- Select part;
- Grab part;
- Move (or assembly) part.

In this system, a gesture is regarded as a process unit, and the process of hand motion is divided into three states: the state of selecting part, the state of grabbing part, the state of moving part.

B. Training the behavioral model

According to the cognitive-behavioral model described in part A, we train the model. The details as follow.

Step1: Obtain observation sequence O using data glove and position tracker.

Step2: Initialize the Hidden Markov Model $\lambda = (N, M, \pi, A, B)$.

Step3: Initialize the parameters again with Viterbi algorithm.

Step4: Calculate parameters with Baum-Welch algorithm, and get the new Hidden Markov Model

$\lambda' = (\pi', A', B')$.

Step5: Calculate the probability of observing sequence $P(O \mid \lambda')$ in the model λ' .

If

$| P(O \mid \lambda') - P(O \mid \lambda) | < threshold$,

the training completes.

Otherwise, jump Step4.

C. Predicting according to the model above

The predictions of HMM and observed sequence of current image are independent of each other, which easily leads to large forecast errors. To tackle the problem of gesture predicting errors, the algorithm proposed is that we establish the relationship between the two which can improve the tracking accuracy. The concrete prediction steps as follows.

Step1: Initialization. The model $\lambda = (N, M, \pi, A, B)$ acquired in part B, the time variable $t=1$.

Step2: Predict the gesture state n at time t.

Step3: Calculate the gesture X_t' corresponding to the probability b_{nt}.

Step4: Calculate the similarity d_t between X_t' and the current image. If d_t < *threshold*, jump Step7. Otherwise, jump Step5.

Step5: In the interval *(t-3, t-1)* and *(t+1, t+3)*, select the largest probability b_{nt}, and get the gesture X_t' associated with it. Calculate the similarity between X_t' and the current image. If d_t < *threshold*, jump Step7. Otherwise, jump Step6.

Step6: In the range *t-3* to *t+3*, b_{nt} and the gesture X_t' corresponding to b_{nt} is weighted respectively, and computed the sum that is the predictive value X_t'.

Step7: Sample particles according to the dynamic system model and X_t', and calculate weights of the particles on the basis of the observation likelihood models. Assuming sampling number *SampleNum*, the set of weights $\omega = \{ \omega_i, 1 \leqslant i \leqslant SampleNum\}$.

Step8: Re-sample with PERM.

If $\omega_i < \omega_{min}$, select the particle depending on the probability *a*, and its weight $\omega_i = 2 * \omega_i$.

If $\omega_i > \omega_{max}$, the particle is sampled *K* times, and their weights $\omega_i = \omega_i / K$.

Step9: Calculate output gesture according to these particles and their weights.

D. PERM method

Although standard re-sampling method can restrain particle degradation problems, this method takes weights as standard, which may lose the particles of small weights, and copy the particles of large weights many times.

To solve this problem, we introduce PERM method, which is the most efficient approach for solving protein folding problem. PERM is a “growth” algorithm, in which all the monomers, except for the first two ones, are placed one by one on the regular square lattice until all the *N* (length of the chain) Monomers are placed [26].

We can describe the algorithm by giving the strategy of how to choose an action to place the n^{th} monomer, in which the first *n*-1 monomers have been placed. The details are as follows [26]:

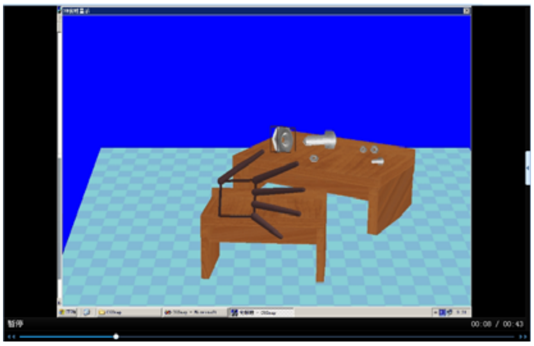
- 1) Calculate the qualities of all possible valid action α of placing the n^{th} monomer.
- 2) Calculate the predicted weight *Wn* and thresholds *Wn+*, *Wn-*.
- 3) Compare *Wn* with *Wn+* and *Wn-*, and then choose an action.
- 4) If $Wn \in [\, Wn-, Wn+ \,]$, choose an action α at a some probability.
- 5) If *Wn*< *Wn-*, the partial chain is pruned at probability 0.5, otherwise we can choose an action α at a some probability, then double the weight *Wn*.
- 6) If *Wn*> *Wn+*, we can choose a set of actions $A = \{ \alpha_{v1}, \alpha_{v2}, \ldots, \alpha_{vk}\}$ consisting of *k* mutually different continuations at a probability and implement them independently, then calculate each branch’s corresponding weight.

According to this principle, the PERM method can adjust automatically all the weights. The result is that it produces neither too large weights, nor too small weights to effectively suppress the extreme point of the error.

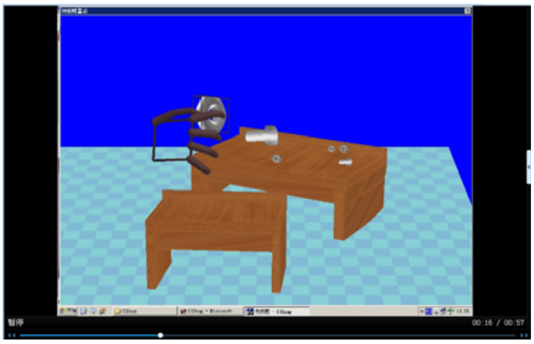
IV. Experimental Results

We have realized our algorithm using VC++, a PC with CPU for Inter(R) Core(TM), frequency for 2.66GHz and 4GB memory, a CCD camera and a DAQ card. The training data used is obtained by a single user. The experiments of virtual assembly are conducted by different users.

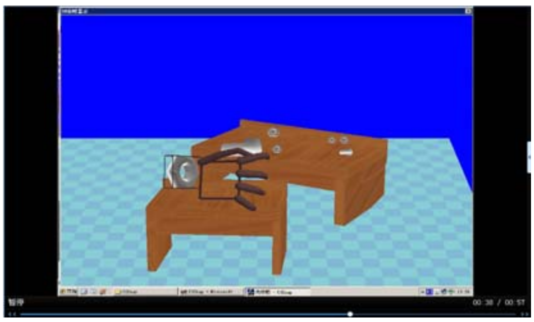
Some of the virtual assembly procedures are shown in Fig.2.



(a) Assembly scene



(b) One of moving processes



(c) One of assembly processes

Figure 2. Virtual assembly processes

Experiment1: Track with a behavior model and without the model.

According to the behavior model that has been trained, our algorithm will spread the particles in a plausible direction, which can reduce the numbers of sampling particles and the time consuming. We compare the performance of our algorithm with that of a simple SIR particle filter (PF). The results are presented in Fig.3.

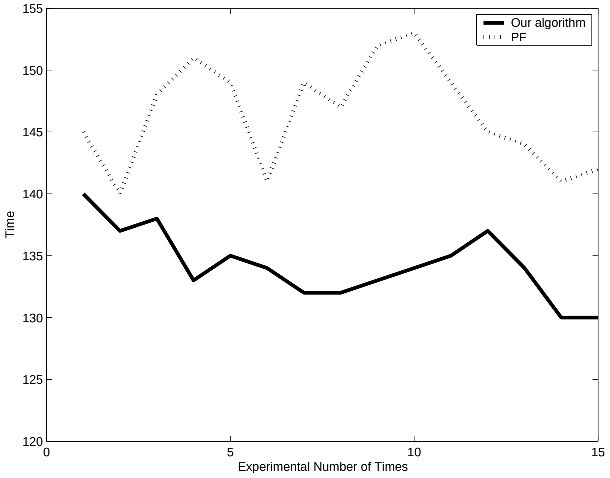


Figure 3. The average time consuming of every frame

Fig.3 is shown that our algorithm is less time consuming than PF under the condition of the similar accuracy. This is due to the fact that each variable of hand is sampled in PF algorithm, while the variables varied greatly are sampled in our algorithm.

Experinment2: Track with and without PERM method.

To evaluate the effect of the PERM method, we track hand motion using our algorithm and the Hidden Markov Model algorithm without PERM respectively.

Define accuracy:

$$a = e^{-\lambda HD}$$

In the formula, *HD* is a Hausdorff distance between the projection of tracking results on image plane and actual observation, λ is an empirical value. In this experiment, λ=0.01.

What this formula means is that the smaller the value *HD*, the greater the precision *a*.

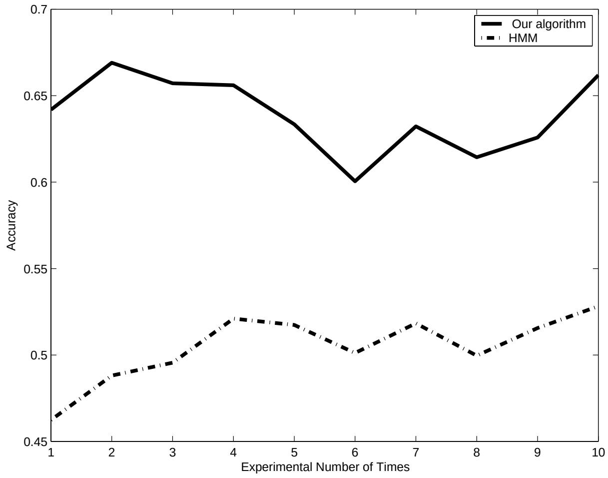


Figure 4. Tracking accuracy

In Fig.4, we can conclude that the algorithm introduced PERM provides better accuracy than the HMM method that is not introduced PERM.

Experiment3: To further illustrate the effectiveness of PERM method, that is, how the method chooses sample particles.

TABLE I. THE WEIGHT CHANGE IN PERM METHOD

The number of particle	1	2	3	4	5	6	7	8	9	10
Weight	0.14	0.40	0.20	0.12	0.02	0.01	0.03	0.05	0.01	0.02
The results (accept or not)	Accept	Accept, and continue to sampling	Accept	Accept	Not accept	Not accept	Accept, and weight doubled	Accept, and weight doubled	Not accept	Not accept

In Table1, “The results (accept or not)” means sampling results, which means that the particle will be accepted or not accepted by PERM, and its weight will be adjusted or not. “Accept” means that the particle is selected and its weight is kept unchanged. “Not accept” means that the particle doesn’t meet the requirement, and is not selected. “Accept, and weight doubled” means that the proportion of particles increases and its weight becomes doubled. “Accept, and continue to sampling” means that we should spread particles on the basis of this particle.

The principle of PERM is to choose particles depending on the weights of the particles.

If the weight $\omega \in [\omega_{min}, \omega_{max}]$, it can select the particle, and its weights is invariable.

If the weight $\omega < \omega_{min}$, it may choose the particles in accordance with the probability *a*, and the weight $\omega = 2 * \omega$.

If the weight $\omega > \omega_{max}$, it can continue to sample particles K times, and the weight of each particle $\omega = \omega / K$.

In Table1, we can see that all particles have been given attention. The benefit of this method is both to avoid the particles of small weights loss, but also inhibit the particles of large weights from copying repeatedly, which is different from and superior to the standard re-sampling. The comparison result can be seen from Fig.5 in which we compare our algorithm with PF of standard re-sampling.

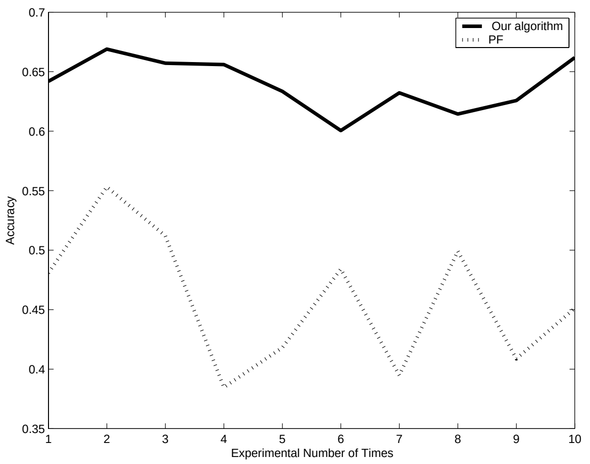


Figure 5. The comparison result between our algorithm and PF

Experiment4: Combined with the above analysis, compare above the three algorithms to evaluate the real-time of our algorithm (Fig.6).

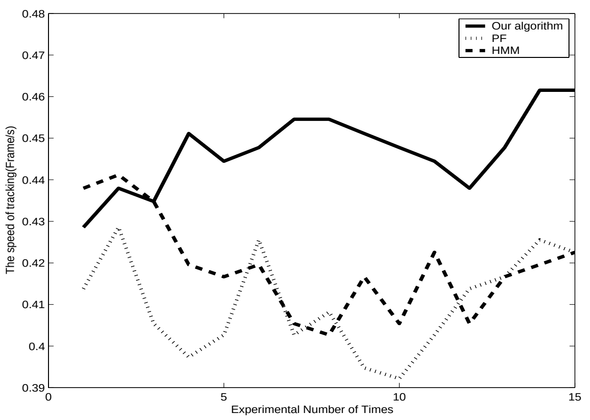


Figure 6. The real-time comparison

In the above machine configuration, we track the hand motion through frame image data obtained from the data acquisition card.

In Fig.6, we can see that the tracking speed of our algorithm is fastest, and its average speed is 0.447 frame/s. The tracking speed of PF algorithm is slowest. It can be seen that our algorithm is the best real-time.

V. CONCLUSIONS

A major highlight of this paper is that hand motion behavior is divided into stages from the perspective of cognitive psychology. We build a system model for each stage, and sample particles selectively to reduce the numbers of particles and save time consumption of the algorithm. At the same time, we introduce a new sampling method PERM, which is the most efficient approach for solving protein folding problem, to avoid particle degradation, but also effectively inhibit the error between the observation model and the true target distribution. Our algorithm is suitable for real-time application, and improves the tracking accuracy.

However, when we track the motion out of the training models, the tracking errors may be larger. In future, our research will focus on detecting tracking failure from time to time and updating system model, in order to enhance the stability of our algorithm.

ACKNOWLEDGMENT

This paper is supported by National Natural Science Foundation of China (No.60973093, No. 60873089), Natural Science Foundation for Distinguished Youth Scholar of Shandong Province (No.JQ200820), Key Project of Natural Science Foundation of Shandong Province(2006G03), Science and Technology Plan of Shandong Province Education Department (J07YJ18).

REFERENCES

[1] Aggarwal JK, Cai Q, “Human motion analysis: A review,” Computer Vision and Image Understanding, vol. 73, pp. 428–440, March 1999.

[2] Jun J, Wang L, and Singh S, “Video analysis of human dynamics – A survey,” Real Time Imaging, vol. 9, pp. 321–346, May 2003.

[3] Wang L, Hu WM, and Tan TN, “A survey of visual analysis of human motion,” Chinese Journal of Computers, vol. 25, pp. 225–237, March 2002.

[4] Robert T. Collins, Mean-shift Blob, “Tracking through Scale Space,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR ‘03), Madison, 2003, pp. 234-240.

[5] Isard M, Blake A, “A mixed-state CONDENSATION tracker with automatic model-switching,” Proceedings of the Sixth International on Computer Vision, Bombay, 1998, pp. 107-112.

[6] Wang Xi-Ying, Zhang Xi-Wen, and Dai Guo-Zhong, “An approach to tracking deformable hand gesture for real-time interaction,” Journal of Software, vol. 18, pp. 2423–2433, October 2007.

[7] Erol A, Bebis G, Nicolescu M, Boyle, R. D., and Twombly, X., “Vision-based hand pose estimation: A review,” Computer Vision and Image Understanding, vol. 108, pp. 52–73, Jan. 2007.

[8] Kolsch M., Turk M., “Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration,” Computer Vision and Pattern Recognition Workshop, pp. 158–168, 2004.

[9] Chen Rui, Liu Guoyi, Zhao Guoying, Zhang Jun, and Li Hua, “3D Human Motion Tracking Based on Sequential Monte Carlo Method,” Journal of computer aided design and computer graphics, vol. 17, pp. 85–92, Jan. 2005.

[10] MacCormick J, Isard M. Partitioned Sampling, Aarticulated Objects, and Interface-Quality Hand Tracking. In Proc. ECCV, Dublin, 2: 13-19, 2000.

[11] JP Wachs, H. Stem, Y. Edan. Real-Time Hand Gesture Telerobotic System Using the Fuzzy C-Means Clustering Algorithm. World Automation Congress(WAC), Orlando, Florida, USA. 403-410,2002.

[12] Feng Zhiquan, Yang Bo, Li Yi, Wang Zhonghua, and Zheng Yangwei, “Research on Human Hand Tracking Aiming at Improving Its Accurateness,” Journal of Computer Research and Development, vol. 45, pp. 1239–1248, July 2008.

[13] Feng Zhiquan, Yang Bo, Li Yi, Zheng Yanwei, and Zhang Shaobai, “Research on Hand Gestures Tracking Based on Particle Filtering Aiming at Optimizing Time Cost,” Acta electronica sinica, vol. 37, pp. 1989–1995, Sep. 2009.

[14] Feng Zhiquan, Yang Bo, Zheng Yanwei, Tang Haokui, Xu Tao, and Li Yi, “Design and Research on Particle Filtering for Human Hand 3D Tracking,” Journal of Software, Supplement, 2008, 19: 87-95.

[15] Gordon N J, Salmond D J, and Smith A F M, “A novel approach to nonlinear/ non-Gaussian Bayesian state estimation,” IEEE Proceedings on Radar and Signal Processing, vol. 140, pp. 107–113, Feb. 1993.
[16] Djuri’c P M, Kotecha J H, Zhang Jiangqui, etal, “Particle filtering,” IEEE Signal Processing, vol. 20, pp. 19–38, May 2003.
[17] LIANG Ling-jian. Modern Cognitive Psychology[M]. Shanghai: Shanghai Education Press, 2006.
[18] Schlenzig, J., Hunter, E. and Jain, R. Recursive Identification of Gesture Inputers using Hidden Markov Models, Proc. Second Ann. Conf. on App1. of Comp. Vision, 1994, pp. 187-194.
[19] Yamato, J., Ohya, J., and Ishii, K.. Recognizing Human Action in Time-Sequential Images using Hidden Markov Models, Proc.(IEEE International Conference on Computer Vision)ICCV, IEEE Press, Kerkrya, Greece, 1992, pp. 379-385.
[20] T. Kurita and S. Hayamizu. Gesture recognition using HLAC features of PARCOR images and HMM based recogniser. Proc. of the Int. Conf. on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 422-427.
[21] Li Xiaomei, “Improved PERM Algorithm of Protein Folding Simulations,” Computer Science, vol. 34, pp. 197–199, 2007.
[22] Zhu Ming, Guo Chunsheng, “Hidden Markov Model and Its Application and Progress,” Computer Systems and Applications, vol. 19, pp. 255–259, July 2010.
[23] Duan Hongmei, Wang Jun, Ma Lianghe, and Xu Ran, “The Application of HMM in Speech Recognition,” Journal of Mathematics for Technology, vol. 18, pp. 16–20, June 2002.
[24] H. Lee, J. Kim, “An HMM-Based Threshold Model Approach for Gesture Recognition,” IEEE Trans. TPAMI, vol. 21, pp. 961–973, Oct. 1999.
[25] Wang Zhitang, Cai Linbo, “Hidden Markov Model(HMM) and Its Application,” Journal of Human University of Science and Engineering, vol. 30, pp. 42–44, April 2009.
[26] Huang Wenqi, Lv Zhipeng, “Personification algorithm for protein folding problem: Improvement in PERM,” Chinese Science Bulletin, vol. 49, pp. 2092–2096,2004.

Wei Gai, born in 1987, Shandong Province, China. She is an enrolled postgraduate of university of Jinan. And research interests on human-computer interaction and moving human tracking.

Zhiquan Feng, born in 1964, Sichuan Province, China. He is a Ph.D, associate professor, and supervisior. His main research interests include image processing, moving human hand tracking, and human-computer interaction.

Bo Yang, born in 1965. He is a professor and vice-president of University of Jinan. He is the Director of the Provincial Key Laboratory for Network-based Intelligent Computing, and Member of the Technical Committee of Intelligent Control of Chinese Association of Automation. His main research interests include computer networks, artificial intelligence, machine learning, knowledge discovery, and data mining.

Yan Lin, born in 1989, enrolled postgraduate. Her main research interests in moving hand tracking and hunman-computer interaction.

XianHui Song, born in 1986, enrolled postgraduate. Her main research interests on human-computer interaction and moving human tracking.