IADIS International Journal on Computer Science and Information Systems
Vol. 7, No.2, pp. 1-17
ISSN: 1646-3692

VIDEO ACTION RECOGNITION BASED ON HIDDEN MARKOV MODEL COMBINED WITH PARTICLE SWARM

Haiyi Zhang . *Jodrey School of Computer Science, Acadia University, Canada*

ABSTRACT

This paper covers a novel learning algorithm for a HMM (Hidden Markov Model) based on PSO (Particle Swarm Optimization) is addressed, and a new video action recognition approach is proposed. Firstly, we extract the features of the action's trajectories from each of the target objects. Then, we compare them with the semantic event probability produced in the HMM. Secondly, we improve the HMM by modifying its learning algorithm parameters based on PSO. This has the advantage of changing the computational learning level of the HMM from a result that is locally optimal to one that is globally optimal; Meanwhile, it can avoid the common computational errors associated with data overflow. Finally, we identify the objective activity patterns using the Time Warping Method by matching event probability sequence. Practical data experiments show that the presented algorithm is efficient by reflecting the real activities and can influence how the problem is attacked. The comparative experiments also show that it has advantages over the Baum-Welch algorithm and other famous methods.

KEYWORDS

Action recognition, behavior modeling, HMM (Hidden Markov Model), event probability sequence, PSO (Particle Swarm Optimization)

1. INTRODUCTION

Action recognition technology plays an important role in multimedia information representation, searching, image processing, intelligent video surveillance, human-computer interaction and content based video information retrieval [Bobick96, Veeraraghavan05, Bashir07]. By extracting the features of a target object's movement and status, and by using machine learning and classification, computers can recognize activities and describe them in natural language. Current popular dynamic recognition methods includes are DSN (Dynamic Bayesian Network), NN (Neural Networks) and HMM (Hidden Markov Models). HMM uses parameters to describe statistical characteristics of random procedures. It contains two random

procedures: the Markov chain and normal random procedure. The Markov chain uses state transition probability to describe states transition. Normal random procedure uses observation probability to describe the relationship between description sequences and observation sequences.

1) X is the set of states, $X=\{S_1, S_2,..., S_N\}$, N is the total number of states, and q_t represents state at time t;
2) O represents the set of observable symbol, $O=\{V_1,V_2,...,V_M\}$, and M is the number of symbol that could observed from one state;
3) $A=\{a_{ij}\}$ is state transition probability distribution, $a_{ij}=P\{q_{t+1}=S_j \mid q_t=S_i\}$, $1\le i, j\le N$, and $a_{ij}\ge 0$ for any (i, j);
4) $B=\{b_j(k)\}$ represents observation probability of state j, the probability of state j exports specific observation symbolic, $b_j(k)=P\{O_t=V_k \mid q_t=S_j\}$, $1\le k\le M$;
5) $\pi=\{\pi_i\}$, $\pi_i=P\{q_1=S_i\}$ is the initialization state distribution.

There are three difficult problems in traditional HMM.

- Evaluation. Given observation sequence $O=O_1,O_2,\ldots,O_T$ and mode parameters $\lambda=(A, B, \pi)$, calculate probability $P(O|\lambda)$ of given observation sequence with model parameter λ.
- Decoding. Given observation sequence $O=O_1,O_2,\ldots,O_T$ and model parameters $\lambda=(A, B, \pi)$, find the best states sequence $Q^*=q_1^*,q_2^*,\ldots,q_T^*$ under a specific meaningful circumstance.
- Learning. Given a structure of HMM such as the N and M, find the model parameters $\lambda=(A, B, \pi)$ to maximize $P(O|\lambda)$ under the given observation sequence $O=O_1,O_2,\ldots,O_T$.

The key problem with HMM learning is finding the appropriate model parameters to maximize $P(O \mid \lambda)$ under given observation sequence. However, because of the limit of the training data set, there isn't a *best* method to evaluate λ. Therefore, due to inadequate training, the recognition rate of current methods is usually not very good.

The PSO (Particle Swarm Optimization) is a kind of evolutionary computation and uses swarm intelligence. By using PSO the evaluation function needn't to be derivable, and it could find the best solution rapidly.

In PSO, particle information is represented as N dimensional vector $X_i=(x_{i1}, x_{i2},\ldots,x_{in})$, velocity as $V_i=(v_{i1},v_{i2},\ldots,v_{in})$, x_pbest and x_gbest represent individual best value and global best value. Expressions (1) and (2) are update equations of velocity and position:

$$v_{id}(k+1)=w*v_{id}(k)+c_1*r_1*[x\_pbest_{id}(k)x_{id}(k)] +c_2*r_2*[x\_gbest_{gd}(k)-x_{id}(k)] \quad (1)$$

$$x_{id}(k+1)= x_{id}(k)+ v_{id}(k+1) \quad (2)$$

$$w=(w_{final}-w_{initial})(max\_gen-gen)^n/max\_gen^n+w_{initial} \quad (3)$$

w is weight coefficient, c_1 and c_2 are learning coefficients, r_1 and r_2 are random number between 0 and 1, $w_{initial}$ is initial weight coefficient, w_{final} is linear decent end weight coefficient, max_gen is the max times of iteration, and gen is current times of iteration

1.1 Motivations and Overview of Approach

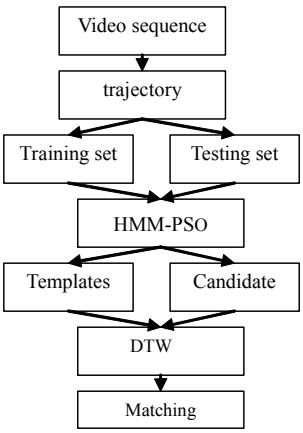

Figure 1. Action recognition based on event probability

This paper uses the continuous features of action trajectory; therefore continuous HMM is used in training parameters. Some theoretical researches are made and a new action recognition approach is proposed. The activities trajectory are concluded according to the objective motion information, and an improved HMM based modeling and classification methods, named HMM-PSO, is developed to be used in event probability sequence modeling and video recognition. In HMM-PSO, PSO algorithm is in charge of HMM's parameter training in order to enhances the performance of traditional HMM.

The variation of motion trajectory could reveal some important information about the target objects, such as directions, velocities, positions and so on. The motion trajectory provides lots of useful features for modeling. The significant changes of motion trajectory in time and space often indicate the changes of semantic events. So semantic events could be mined from motion trajectory, and activities could be described as event probability sequences.

We use event probability to model the behavior and take HMM-PSO to mine out the semantic event probability from the original trajectory. Then DTW (Dynamic Time Warping) is used to match the model, and finally get the recognition results. The processing framework is shown in Fig. 1. We have verified the performance and efficiency of HMM-PSO in behavioral action modeling and its learning abilities, and the benchmark testing and comparative experiments show that HMM-PSO is suitable to be used in behavior action modeling at different perspective views.

The advantages of HMM-PSO are as follows.

- HMM-PSO is an efficient modeling approach. Even in case of that there exists great differences between the trajectories for a same kind of behavior, it can also find out the inherent relationships and mine out the probability of the event happening. For example, Fig. 4 is the action trajectory of erasing a blackboard, and it is demonstrating a same kind of action. However, their action patterns are quite different. Fig. 2 and Fig. 3 are the event probability sequences concluded by HMM-PSO. We can see that HMM-PSO has the ability to retrieve the common attributes from the same kind of motion even though their action

models are quite different. The probability of an event happening will change according to the change in the motion trajectory, synchronously.

- To some extent, HMM-PSO can mine out information about connotative behavior from the superficies and indicate the activities by the probability of the event happening, which can be seen as recognition of video content and a demonstration by the semantic expression.
- HMM-PSO is better at resolving the generalized extremum problem during parameter training computation and it can lead to an excellent parameter model. For example, Fig. 4 (a) and Fig. 4 (b) indicate the objective function values obtained by HMM-PSO and HMM-Baum. It is very clear that HMM-PSO performs better than HMM-Baum, since HMM-PSO will give a better training on parameter λ so that $\sum\log(P(O\ |\lambda))$ is maximized. The measurement rules of HMM-PSO generate a better computational result. And more, Table 2 indicates the results obtained by HMM-PSO and HMM-Baum with different parameters. It is also very obvious that HMM-PSO can achieve bigger objective values, which illustrates that HMM-PSO has better performance in parameter optimization and can avoid falling into local optimal. We also find that HMM-PSO is better than the tradition Baum-Welch algorithm.

1.2 Contributions of the Paper

Our contributions are summarized as follows.

- We designed a method based upon PSO to optimize parameter training process of HMM, and the step-by-step process of the algorithm is also provided. Experiments show that HMM-PSO can avoid the problem of data overflow efficiently.
- We proposed a method of mining out event probability sequences from motion trajectories using HMM-PSO. The results indicate the inner happening probabilities of the events and have the ability to show their semantic meanings. Our presented method has the potential to build accurate models for complex problems.
- We designed and ran various simulations experiments and comparative experiments to test our theories and methods. All of the experiments support the opinion that HMM-PSO has advantages in action modeling and content recognition.

1.3 Organization of the Paper

Section II reviews related works. Section III details the active learning algorithms of HMM-PSO. Section IV describes action recognition. A large number of experimental and comparative results are provided and discussed in Section V, prior to a summary in Section VI.

2. RELATED WORKS

Bobick et al[Bhobick96] transformed target motion image sequences into MET （Motion-Energy Images) and MHI Motion-History Images), and calculated the similarity in degrees of the test samples with the templates by measuring the Mahalanobis distance between them.

Veeraraghavan et al [Veeraraghavan 05] used DTW, by computing the distance between two shape sequences and then identifying the activities or gaits.

Bashir et al [Bashir 07] presented an improved HMM based on target trajectory analysis, and proved that HMM performs better than GMM (Gaussian Mixture Model). Huang et al [Huang,07] developed a method by using the ACO (Ant Colony Optimization) to classify human postures trajectories, and then used the HMM to model their activities. Cuntoor et al [Cuntoor 08] used target trajectory event probability sequences for modeling and identified the activities with HMM. Uddin et al [Uddin 08] used the approach of independent component analysis to set up the models of action shapes, and used HMM for recognition.

There is also some academic research into this method of using a HMM. Pau and Chin et al [Pau 08] proposed a hierarchical HMM based on space, activities and temporal context. Hasan et al [Hasan 08] addressed a reconfigurable HMM to recognize human behavioral action in an environment of wireless sensors.

Du et al [Du 08] developed a novel approach by decomposing the action into multiple interactive stochastic processes, which reflect the relationships between the motion sequence details. Park and Aggarwal et al [Park 03] presented an approach of two-person interactions action recognition by a hierarchical BN (Bayesian network). Muncaster et al [Muncaster 07] presented a general model of *d*-layer DSN to deal with more complex recognition problems, and the deterministic annealing clustering method are used in each of the layers to detect the status automatically. Buccolieri et al [Buccolieri 05] used NN to identify human's postures by analyze the profile. Sumpter and Bulpitt [Sumpter 04] used self-organizing NN for action recognition. Zhu et al [Zhu06] developed a new motion descriptor based on optical flow histograms and used SVM(Support Vector Machine) as a classifier. Cao et al [Cao 04] presented a new strategy of representing the video motion by the filtered images and took SVM as images classifiers.

Action recognition based on motion trajectory has attracted many attentions these days. Rao et al [Rao 02] set up action behavior models according to the changes on 2-D trajectory curvatures, and also verified that the methods are not sensitive to the angle of views. Bashir et al [Bashir07] presented a method of deciding the segmentation points by hypothesis testing approaches combined with the changing of velocity and acceleration on the curve, and then indexed the video by the Euclidean distance between the split trajectories and the results of matching algorithm. Two kinds of view-angle-independent feature representation methods are addressed in [Bashir06], which are CDF (Center Distance Function) and the CSS (Curvature Scale Space), and HMM are used in behavior identification and classifier. Furthermore, Bashir et al [Bashir07-2] also used GMM and HMM in action recognition. Cuntoor et al [Cuntoor 05] improved the HMM based recognition method by considering that the event status would change along with the changes of features. Thus, the event probabilities were denoted by the migration in the implicit layer and the pattern of motion which could be described by the event probabilities sequences.

Some further academic researches on the method of event probability description, action behavior recognition, view-angle-independent identification method, the robustness of parameters in HMM, and the practical applications are also been discussed in [Cuntoor 05] by Cuntoor et al.

3. ACTION LEARNING OF IMROVED HMM (HMM-PSO)

In this paper, a new target optimization function log ($P(O|\ \lambda)$) is proposed to avoid data overflow in computation of feed-forward probability, described as follows.

$$\max \log(\prod_{k=1}^{K} P(O^k \mid \lambda)) = \max \sum_{k=1}^{K} \log P(O^k \mid \lambda) = \max \sum_{k=1}^{K} \log(\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}^k)\beta_{t+1}(j)) \quad (4)$$

$$St.\ \forall 1 \le i, j \le N, 1 \le m \le M, \pi_i \ne 1, a_{ij} \ne 1, \pi_i \ge 0$$

$$b_j(O_{t+1}^k) = \sum_{m=1}^{M} c_{jm} \mathrm{N}(O_{t+1}^k, \mu_{jm}, \Sigma_{jm}), 1 \le j \le N \qquad a_{ij} \ge 0, c_{jm} \ge 0, \Sigma_{jm} \ge 0,$$

$$\sum_j \pi_j = 1, \sum_{i,j} a_{ij} = 1, \sum_{j,m} c_{jm} = 1$$

HMM in this paper uses an ergodic model. a_{ij} in the state transition probability matrix is positive and the dimension is N^2. Besides, suppose the dimension of observation vector is D, then the parameters to be optimized $\lambda=(A, B, \pi)$ could be the particle in PSO, so $X=\lambda=(\{a_{ij}\}, \{c_{jm}, \mu_{jm}, \Sigma_{jm}\}, \{\pi_i\})=(x_1, \ldots, x_{N*N}, x_{N*N+1}, \ldots, x_{N*N+3N*M*D}, x_{N*N+3N*M*D+1}, \ldots, x_{N*N+3N*M*D+N})$, the particles' dimension are:

$N^2 + 3N * M * D + N$.

The first N2 dimensions represent the state transition matrix A, the last N dimensions represent the initial probability distribution π, and the other dimensions represent the parameters of observed probability distribution.

In this paper, the particles are restricted by a few constrains. If one particle violates the rule, it will be mapped into the feasibility space. The algorithm is as follows.

Pseudo-code of PSO Feasibility Space Verification Algorithm (FV):

Step1: If the particle position vector X satisfies the constraint condition, then finish.

Step 2: If $x_i<X_{min}$, then $x_i= X_{min} - x_i$; and if $x_i>X_{max}$, then $x_i= X_{max}$.

Step3: If X satisfies the constraints, then finish.

Step 4: If $\sum_{i=N*k+1}^{N(k+1)} x_i = 0$, then $x_i=1/N$, and $k=\{0,\ldots,N-1\}$.

Step 5: Mapping the parameters x in c_{jm} and π_i in the same way.

Step6: If X still violates the constraints, then verify X by normalizing it. Define X as the position of the particle that violates the constraints, and X^* as the verified position:

Step 6.1: For state transition matrix A, that is the first dimensions: $x_i^* = x_i \Big/ \sum_{i=N*k+1}^{N(k+1)} x_i$, $1 \le i \le N$, $k=\{0,\ldots,N-1\}$;

Step 6.2: For dimensions representing height coefficients c, that is the dimensions from $N^2 + 1$ to $N^2 + N * M * D$: $x_i^* = x_i \Big/ \sum_{i=N^2+1}^{N^2+N*M*D} x_i$, $N^2+1 \le i \le N^2+N*M*D$;

Step 6.3: For dimensions representing covariance matrix, that is the dimensions from $N^2 + N * M * D + 1$ to $N^2 + 2N * M * D$ as follows:

$x_i^* = \mid x_i \mid$, $N^2+N*M*D+1 \le i \le N^2+2N*M*D$;

Step 6.4: For dimensions representing initial distribution, that is the last N dimensions:

$$x_i^* = x_i \Big/ \sum_{i=N^2+N*M*D+1}^{N^2+N*M*D+N} x_i \ , \ N^2+3N*M*D+1 \le i \le N^2+3N*M*D+N.$$

Pseudo-code of HMM parameter optimization based on PSO (HMM-PSO):

Step1: Initialize the population size, position x_i and velocity v_i ($Vmin \le v_i \le Vmax$), *x_pbest*, and *x_gbest*;

Step2: If the termination conditions are satisfied, go to Step3, else do as follows:

Step 2: update each particle's velocity and position according to (1) and (2)

Step 2.2: judge whether the current particle satisfies the constraint conditions, if not, then use FV to map it;

Step 2.3: evaluate each particle by target function $f(x_i)$ according to (4).

If $f(x_i) > x\_pbest_i$, then $x\_pbest_i = f(x_i)$;

If $\max(f(x_i)) > x\_gbest_i$, then $x\_gbest_i = \max(f(x_i))$.

Step3: The *x_gbest*'s position is the recommended HMM's parameters.

4. ACTION RECOGNITION BASED ON HMM-PSO

4.1 Trajectory Feature Extraction

This paper uses 2-D motion trajectory, and chooses two features of motion trajectory. The first feature is shape, namely the geometry derived from the pattern of the motion and the direction. The second feature is velocity, which describes the dynamics of the object's trajectory. References [Cuntoor 05, Cuntoor 08] choose the geometry information to describe activities such as position, curvature. But in this paper, both static and dynamic information are considered. Motion trajectory features are described by position, acceleration and curvature, namely $s(t) = [r(t)\ v(t)\ a(t)\ k(t)]$.

Motion trajectory is described as

$$r(t) = [x(t)\ y(t)\ t],\ 1 \le t \le n \qquad (5)$$

and n is frame number of video sequence.

So the velocity and acceleration of the target object is

$$\text{velocity vector: } v(t) = [x'(t)\ y'(t)\ 1] \qquad (6)$$

$$\text{acceleration vector: } a(t) = [x''(t)\ y''(t)\ 0] \qquad (7)$$

This paper uses $k(t)$ to describe the key feature of motion.

$$k(t) = |r'(t) \times r''(t)| / \|r'(t)\|^3 \qquad (8)$$

$r'(t)$, $r''(t)$ and $\|r'(t)\|$ is speed, acceleration and velocity.

Operator X means cross product of the vector. Substitute (6), (7) into (8)

$$k(t) = \frac{\sqrt{y''(t)^2 + x''(t)^2 + (x'(t)y''(t) - x''(t)y'(t))^2}}{(\sqrt{y'(t)^2 + x'(t)^2 + 1})^3} \qquad (9)$$

4.2 Event Probability Sequence Computation

Event probability sequences could quantify important features properly and adapt to data changes dynamically. There are two steps to compute event probability sequences for each action in this paper. The first step is to train the HMM using given motion trajectory sample, and the second step is to compute event probability sequences of given motion trajectory.

$O=\{O_1,O_2,...,O_T\}$ represents motion trajectory of target object, namely the observation sequence, and O_t is the feature vector of the trajectory in t frame, namely $O_t=[r(t)\ v(t)\ a(t)\ k(t)\ ]$. $Q=\{q_1,q_2,...,\ q_T\}$ represents hidden states in HMM, $q_t \in \{S_1,...,S_N\}$. The number of possible hidden state sequences is N^T for the given observation sequence O. And among those possible sequences, the sequence that has the largest probability is the best sequence, according to traditional HMM definition.

Reference [Cuntoor 08] points out that the state transition probability indicates the extent of motion variation and the probability of occurrence of a specific event. This paper builds on this idea. The semantic event probability is found by the computation of the maximum state's transition probability. The computational complexity is (N^2 - N).

According to the definition of HMM, the transition probability between state i and j is described as

$$\eta_t^{(1)}(i,j)=P\{\ q_t=S_i,\ \ q_{t+1}=S_j|O,\ \lambda\ \},\ 1\le i,j\le N \qquad (10)$$

$\eta_t^{(1)}(i,j)$ could be computed by the HMM learning algorithm.

(10) shows state transition in one single frame, and could represent the probability of occurrence. But low level state transition, such as i→j, may lose some important motion features. So high level state transition, like i→i→j→j, is used to represent stable state transition. The formula is described as

$$\eta_t^{(2)}(i,j)=P\{q_{t-1}=S_i,\ \ q_t=S_i,\ q_{t+1}=S_j,\ q_{t+2}=S_j|O,\ \lambda\ \},\ 1\le i,j\le N$$

The following formula could be deduced from the above one:

$$\begin{aligned}\eta_t^{(2)}(i,j) &= P(q_{t-1}=s_i,q_t=s_i,q_{t+1}=s_j,q_{t+2}=s_j,O\mid\lambda)\big/P(O\mid\lambda)\\ &= P(q_{t-1}=s_i,o_1^{t-1},q_t=s_i,q_{t+1}=s_j,q_{t+2}=s_j,o_t^T\mid\lambda)\big/P(O\mid\lambda)\end{aligned} \qquad (11)$$

Reference [21] has proven that (11) could be simplified as:

$$\eta_t^{(2)}(i,j)=\alpha_{t-1}(i)a_{ii}b_i(o_t)a_{ij}b_j(o_{t+1})a_{jj}b_j(o_{t+2})\beta_{t+2}(j)\big/P(O\mid\lambda) \qquad (12)$$

$\alpha_t(i)$ and $\beta_t(j)$ are forward probability and backward probability, namely $\alpha_t(i)=P(O_1, O_2, ..., O_t, q_t=S_i|\ \lambda)$, $\beta_t(j)=P(O_{t+1}, O_{t+2, ..., } O_T \mid q_t=S_j, \lambda)$.

Similarly, suppose state transition sequence is:

$$\underbrace{i\to i\to\cdots\to i}_{p\text{ frames}}\to\underbrace{j\to j\to\cdots\to j}_{p\text{ frames}}$$

Then the computation formula of event probability is:

$\eta_t^{(p)}(i,j)=P\{q_{t-p}=S_i, q_{t-p+1}=S_i, \ldots, q_t=S_i, q_{t+1}=S_j, q_{t+2}=S_j, \ldots, q_{t+p+1}=S_j | O, \lambda\}, 1\le i, j\le N, p=2,3,\ldots,P$ (13)

In (13), data may overflow in $\alpha_{t-p+1}(i)$ and $\beta_{t+p}(j)$ with the passage of time. We deduce from the above one:

$$\eta_t^{(p)}(i,j)=\prod_{s=t-p+2}^{t+p-1} c_s \tilde{\alpha}_{t-p+1}(i) a_{ii}^{p-1} b_i(O_{t-p+2}) \cdots b_i(O_t) a_{ii}^{P-1} b_j(O_{t+1}) b_j(O_{t+2}) \cdots b_j(O_{t+p}) a_{jj}^{p-1} \tilde{\beta}_{t+p}(j) \quad (14)$$

$\tilde{\alpha}_{t-p+1}(i)$ and $\tilde{\beta}_{t+p}(j)$ are scaled forward and backward probabilities :

$$\tilde{\alpha}_t(i)=\left[\prod_{s=1}^{t}(1/\sum_{i=1}^{N}\alpha_s(i))\right]\alpha_t(i) \quad \tilde{\beta}_t(j)=\left[\prod_{s=t+1}^{T}(1/\sum_{i=1}^{N}\alpha_s(i))\right]\beta_{t+1}(j)$$

In each frame, there are N^2 transition situations between state i and j, and for different states i and j, there are N^2 -N transition situations. The probability of an event could be the state transition with the maximum probability, which could be defined as :

$$e_t^{(p)}(k,l)=\max_{i\neq j}\eta_t^{(p)}(i,j) \quad (15)$$

$(k, l)=arg\ \max_{i\neq j}\eta_t^{(p)}(i, j)$, k and l represent the state before and after the occurrence of the event.

An event could be defined with event probability $e_t^{(p)}(k,l)$, scale parameter p and state before and after the event (k, l).

This method has three advantages:

(1) It is simple and robust because it uses the transition of hidden states to capture information about action.

(2) It is efficient to describe motions because event probability sequence is computed from state transition probability.

(3) No manual classification is needed even for a complex action.

4.3 Dynamic Time Wrapping (DTW)

Suppose there are two feature sequences: the template feature sequence $A=a_1,a_2,\ldots,a_i,\ldots,a_I$, and the feature sequence to be tested $B=b_1,b_2,\ldots,b_j,\ldots,b_J$. The relationship between A and B is described as follows : $f=c(1), c(2),\ldots, c(k),\ldots, c(k)$. $c(k)=(i(k), j(k))$, represents the comparison of $j(k)$ frame in sequence B and $i(k)$ frame in sequence A. $c(k)$ could be seen as a point in plane i-j, and it moves with parameter k leaving a curve, which is called matched path.

Suppose $d(i(k), j(k))$ is the local matched distance between the i-th frame in template sequence and the j-th frame in to-be-tested frame. The target of DTW is to find the average matching distance which satisfies both sequences.

$$D(A,B) = \min \sum_{k=1}^{K} d(c(k))w(k) \Big/ \sum_{k=1}^{K} w(k) \quad (16)$$

Suppose path f satisfies the formula above, and is the matching path of sequence A and B. So the distance of f is the matching distance between A and B. $w(k)$ is the weight coefficient of matched point $c(k)$.

There are several constrains in the matching process:

(1) f should be monotonic, namely $i(k\text{-}1) \leq i(k), j(k\text{-}1) \leq j(k)$;
(2) the wrapping function should be monotonic, namely $i(k)\text{-}i(k\text{-}1) \leq 1, j(k)\text{-}j(k\text{-}1) \leq 1$;
(3) the wrapping function must be matched in both the two endpoints of sequence A and B, namely $i(1)=1, j(1)=1, i(K)=I, j(K)=J$;
(4) define the width of the wrapping window r, and $| i(k)\text{-} j(k)| \leq r$.

Suppose there are R motion trajectories and K activities. For given p, there are at least K event probability sequences with R events. One HMM may correspond with more trajectory templates, but each trajectory template only corresponds with one event probability sequence.

R motion trajectories to be tested could result in R candidates of event probability sequence. O_0, the sequence to be tested, could result in K candidates of event probability sequence $Ec = (ec_1, ec_2, \ldots, ec_K)$. Compute the average distance $D(C_i, ec_j)$ according to (17) between each candidate event probability sequence and a specific action. Choose the minimum average distance as the best result, and determine the pattern of the sequence to be tested.

$$D(C_i, ec_j) = \sum_{e_k \in C_i} D(e_k, ec_j) \Big/ N_{C_i} \quad (17)$$

In the formula above, C_i represents the pattern of the i-th action. N_{Ci} represents the number of event probability sequences generated from C_i. e_k represents the event probability sequence belonging to C_i. $D(e_k, ec_j)$ represents the distance of the path computed from two event probability sequences.

Choose the best event probability sequence to be tested using formula (18), and determine the pattern of the sequence to be tested.

$$Opt^* = \arg\min D(C_i, ec_j) \quad (18)$$

5. EXPERIMENTS

We develop a two part experiment: 1) motion modeling analysis; 2) recognition capability analysis. Our testing data is collected from UCF Human Action Data Set of Central Florida University and ASL (Australia Sign Language) of UCI-KDD. We run our programs under the environment of 2.66GHZ CPU, 512M RAM and Visual C++ 6.0.

We use continuous vectors as trajectory characters and run a continuous HMM based on sGD (single-Gaussian Distribution) to model the action, so that we can guarantee the correctness in describing the singles produced by HMM. At the same moment, in the case of training multiple samples and avoiding of data overflow problem during the computing, we use multiple observing sequences and the changing scale method in Baum-Welch parameter estimation, named HMM-Baum. During the parameter training process of HMM-PSO, the

parameter learning can be considered as a constrained optimization problem, which can be demonstrated by $\log(P(O|\lambda)$.

5.1 Behavior Action Modeling with UCF

UCF collects video data from daily behaviors by seven actors shooting at different positions, in which there are mainly seven kinds of simple activities shown in Table 1.

Let Hidden state number N=7, event granularity p=5; $Maxgen$=500, c_1=1.8, c_2=1.8, $w_{initial}$=0.4, w_{final}=0.9, and state transition probability A and initial probability π are produce randomly. The mean value vector and the covariance matrix of the sGD probability density function are obtained by a K-means classifier (i.e. if the observing sequences have been classified into a same kind of trajectory pattern; they will be dynamically classified by the K-mean method further, and the number of the state of HMM is exactly the number of the classes).

5.2 Recognition Performance of HMM-PSO with UCF

We calculate action event probability and match the event probabilities to the behavior pattern by using DTW. The comparative experiment between our presented method to the method by Rao et al show that HMM-PSO performs better in video content recognition.

Form the results of Table 3, it can be found that the *recognition rate* of HMM-PSO is higher, especially in the cases of OpenDoor and PourWater. Moreover, we believe that if we have sufficient training samples, the overall performance of HMM-PSO can be improved further.

At first, we determine the implied status parameter N based on some experienced computation. Then, we specify parameter p by the method of comparative calculation. Fig. 5 shows the average recognition rates, with the solid and dashed lines indicating the results from HMM-Baum and HMM-PSO, respectively. It can be noted that we obtain the best result at p=5. Thus, we let p=5 in our following experiments. Fig. 5 also indicates that HMM-PSO *performs better at identifying the action's pattern*.

5.3 Behavior Action Modeling with ASL

ASL described the motion of human beings and its sample size is 6650, and each of the samples contains multiple attributes. In out experiments, we extract the 2D trajectories of hand motion and try to identify the pattern of gesture.

Let N=12, p=8, Maxgen=500, cl=1.8, c2=1.8, winitial=0.4 and wfinal=0.9. The size of samples for each gesture motion is 69, and the size of learning sample and testing sample are all about 60% of the total sample size. The other parameters keep same to those in Section 4 A.

We will present another advantage of HMM-PSO: its good ability to model independently on perspective. We will show that HMM-PSO can provide accurate models and that it is not subject to the impact of perspective.

The trajectories of the gesture motion of Symbol “alive” at different perspective +60 degrees, zero and -60 degrees are presented in Fig. 4, respectively, and their event probability sequences by HMM-PSO are demonstrated in Fig. 7. It can be seen that, though the

trajectories for the same kinds of motion at different perspective views are quite different, the model produced by HMM-PSO and the event probability sequences given by HMM-PSO for a certain kinds of motion are similar.

5.4 Recognition Performance of HMM-PSO with ASL

We use HMM-Baum and HMM-PSO in action modeling for multiple kinds of motions. Table 4 demonstrates that the performance of recognition rate of HMM-PSO is better since than that of HMM-Baum, because the parameters' capabilities of HMM-PSO are better.

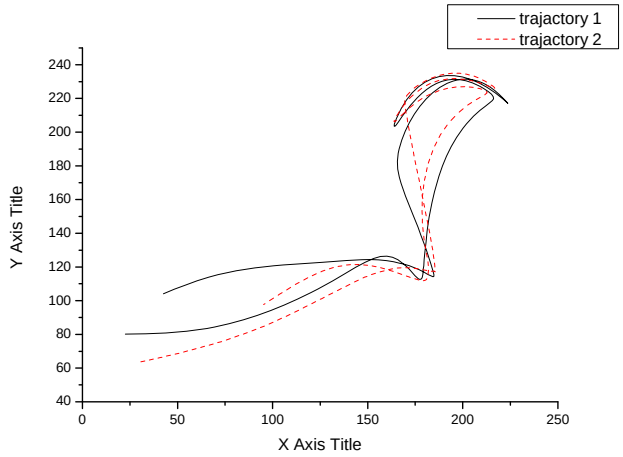

Figure 1. Motion trajectory (pick up chalk rub – erase board – lay down chalk rub)

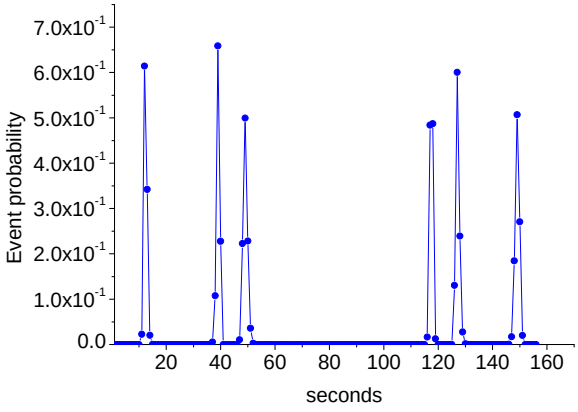

Figure 2. Event probability sequence of the trajectory 1 in Fig.1

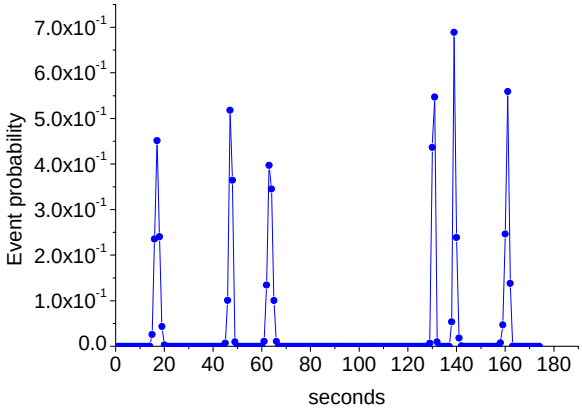

Figure 3. event probability sequence of the trajectory 2 in Fig.1

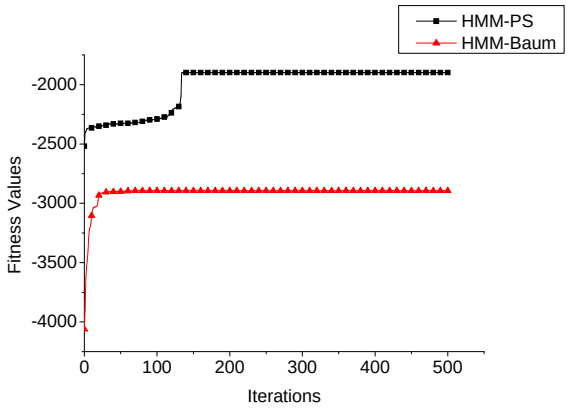

Figure 4 (a). Pour water

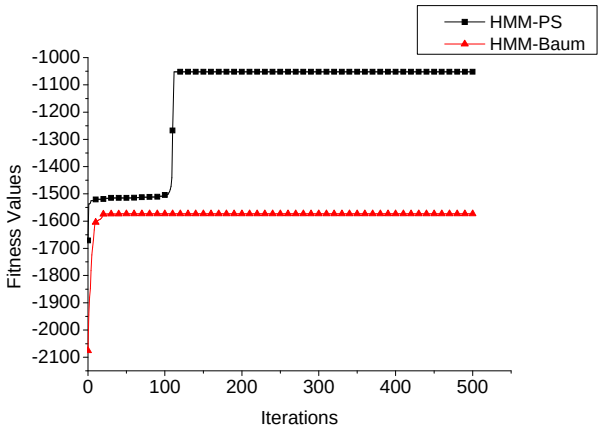

Figure 4 (b). Close door

Figure 4. Value of the objective function by HMM-PSO and Baum-Welch

Table 1. UCF Data Set

Data Set	CloseDoor	EraseBoard	OpenDoor	PickUp&PutDown	PickUp	PourWater	PutDown
Samples	4	4	18	8	21	3	17
LearningSamples	3	3	12	5	15	2	12
TrainingSamples	3	3	12	5	12	2	10

Table 2. Parameters obtained by HMM-PSO and HMM-Baum

ActionClass	HMM-PS parameterlearning model			HMM-Baum parameter learning model		
	N=6	*N*=7	*N*=8	*N*=6	*N*=7	*N*=8
1	-1296.35	-1264.05	-1268.11	-1573.65	-1541.34	-1540.90
2	-4062.51	-3816.89	-3835.35	-5622.41	-5360.44	-5391.11
3	-8873.06	-8612.50	-8603.85	-14105.02	-13756.98	-13568.65
4	-2954.40	-2886.69	-2808.43	-3945.10	-3876.58	-3760.03
5	-6445.22	-6466.55	-6254.55	-9006.14	-8826.49	-8589.22
6	-2015.59	-1874.06	-1811.61	-2893.43	-2871.16	-2733.61
7	-10198.69	-9794.90	-9749.91	-14741.83	-14107.60	-13963.74

Table 3. Recognition rate of HMM-PS and Rao et al [16] for UCF

Recognition rate	CloseDoor	EraseBoard	OpenDoor	PickUp&PutDown	PickUp	PourWater	PutDown
HMM_PS	73.15%	73.15%	100.0%	57.50%	62.96%	62.96%	76.37%
Rao et al [16]	52.96%	75.32%	49.87%	55.78%	/	32.65%	/

Table 4. Recognition rates by HMM-PSO and HMM-Baum for gesture movement action

RecognitionRate	Active class: samples						Average of
(*Rrecog.*)	2:138	4:276	8:552	16: 1104	29: 2001	38: 2622	*Rrecg.*
HMM-PS	82.3%	80.6%	71.6%	59.8%	48.5%	40.4%	63.87%
HMM-Baum	79.7%	74.2%	70.4%	58.9%	46.2%	38.1%	61.25%

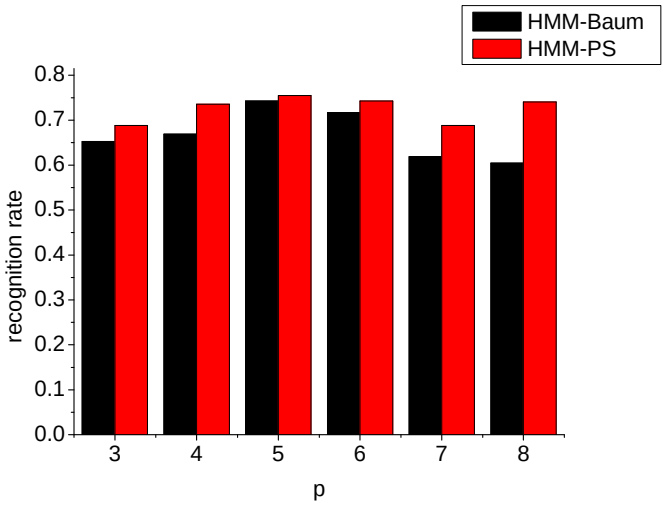

Figure 5. Average rates at different event granularity parameters

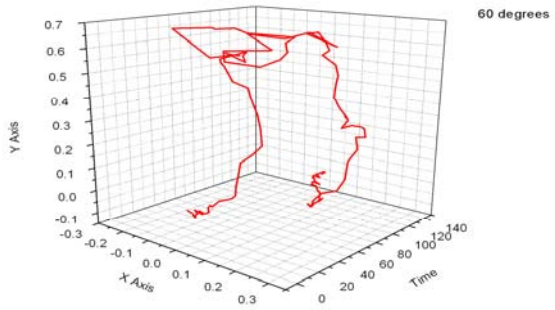

Figure 6. (a) Perspective at 60 degrees counter clockwise

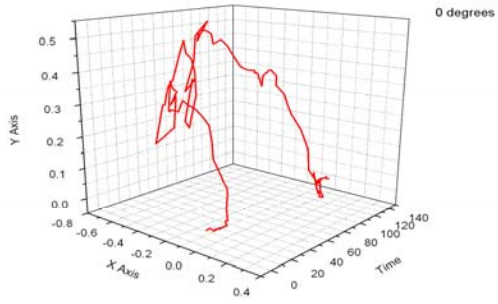

Figure 6. (b) Perspective with no rotating

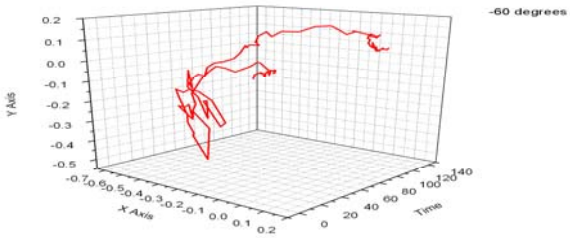

Figure 6. (c) Perspective at 60 degrees clockwise

Figure 6. Gesture motion trajectories at different perspectives of Symbol “alive”

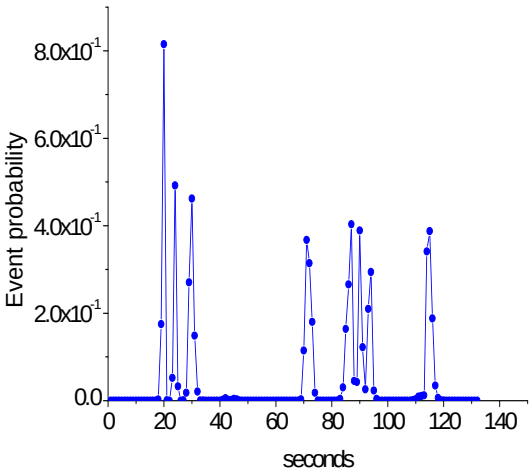

Figure 7. (a) Perspective at 60 degrees counter clockwise

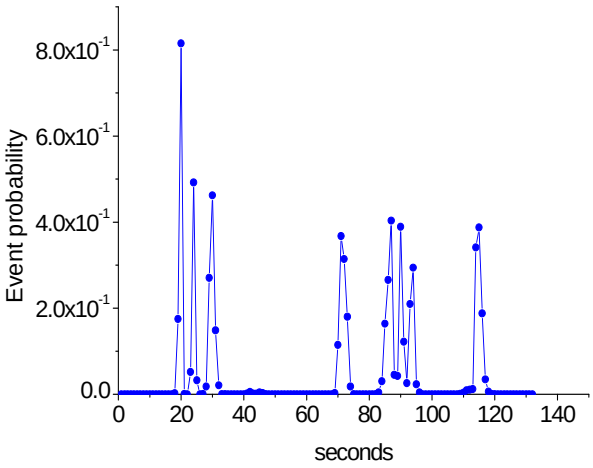

Figure 7. (b) Perspective with no rotation

Figure 7. Event probability sequence of Symbol “alive

6. SUMMARY AND FUTURE WORKS

In this paper, we developed an approach using HMM-PSO for video recognition. We proposed an action modeling method based on event probabilities, and meanwhile, presented an approach to optimize the parameters of HMM. Benchmark data experiments, as well as large number of comparative experiments with other popular methods verify that our presented methods have lots of advantages.

We also know that, though HMM-PSO’s performance is enhanced, the general *recognition rate* still has space to be improved, especially for larger scale active class problems. After academic research and lots of data experiments, we’ve gotten to know that there are two main reasons for HMM-PSO’s deficiency for big size problems. The first reason is that we take the single Gauss density function to indicate observing probability, but we cannot guarantee its efficiency when describing complex motions. The second reason is that we use the DTW method to calculate the event probability, but we are not sure whether it can correctly reflect the distance of the classes between the learning samples and the testing samples.

Our future work will focus on following points. First, we will try to do more practical data experiments, especially on large data set problem, to evaluate our method, model and algorithm in whole scale. Secondly, we will do some academic research on how to improve the recognition rate. Thirdly, we will further discuss how to evaluate and enhance the recognition accuracy. And finally, we will try to apply our presented approaches in more practical applications.

REFERENCES

Journal

Veeraraghavan, A, Chowdhury, A. K. and Chellappa, R. 2005.Matching shape sequences in video with applications in human movement analysis, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1896-1909, Dec.

Bashir, F. I., Khokhar, A. A.and Schonfeld, D. 2007. Object trajectory-based activity classification and recognition using hidden markov models, *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912-1919, Jul.

Cuntoor, N. P. ,B. Yegnanarayana, and Chellappa,R. 2008. Activity modeling using event probability sequences, *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 594-607, Apr.

Pau, C. C. and Chin, D. L.2008. A daily behavior enabled hidden Markov model for human behavior understanding, *Pattern Recognition*, vol. 41, no. 5, pp. 1572–1580, May.

C Sumpter, N. and Bulpitt, A. 2004. Learning spatio-temporal patterns for predicting object behaviour, *Image and Vision Computing*, vol. 18, no. 9, pp. 697-704, Jun.

ao, D. Masoudot, W. and D. Boley, D. 2004. Online motion classification using support vector machines,in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, pp. 2291-2296.

Rao, C. Yilmaz,A. and M. Shah, M. View-2002. Invariant representation and recognition of actions,"*International Journal of Computer Vision*, vol. 50, no. 2, pp. 203-226, Nov.

Bashir, F. I. , Khokhar, A.A. and Schonfeld, D. 2007. Real-time motion trajectory-based indexing and retrieval of video sequences, *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 59-65, Jan.

Bashir, F. I. , Khokhar,A.A. and Schonfeld, D. 2006. View-invariant motion trajectory-based activity classification and recognition, *Multimedia Systems*, vol. 12, no. 1, pp. 45-54, Aug.

Bashir, F. I., Khokhar,A. A. and Schonfeld, D. 2007 (2)Object trajectory-based activity classification and recognition using hidden markov models, *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912-1919, Jul.

Cuntoor, N. P., Yegnanarayana, B. and Chellappa, 2005. Interpretation of state sequences in HMM for activity representation, in *Proc. IEEE Conf. Acoustic Speech and Signal Processing*, Philadelphia, PA, R. pp. 709-712.

Cuntoor, N. P. , Yegnanarayana, B and Chellappa, R. 2008. Activity modeling using event probability sequences, *IEEE Trans. Image Process*, vol. 17, no. 4, pp. 594-607, Apr.

Conference paper or contributed volume

Author, year, paper title. *Proceedings title (in italics)*. City, country, inclusive pages.

Beck, K. and Ralph, J., 1994. Patterns Generates Architectures. *Proceedings of European Conference of Object-Oriented Programming.* Bologna, Italy, pp. 139-149. Bobick, A. and Davis, J. Real-time recognition of activity using temporal templates, in *Proc. 3rd IEEE Workshop Applications of Computer Vision*, Sarasota, FL, 1996, pp. 39-42.

Huang, W. , Zhang, J. and Liu,Z. , 2007. Activity recognition based on hidden Markov models,in *Proc. 2nd Int. Conf. Knowledge Science, Engineering Management*, Melbourne, pp. 532-537.

Uddin, M. Z.,Lee,J. J. and Kim, T. S., 2008. Shape-based human activity recognition using independent component analysis and hidden markov model, in *Proc. 21st Int. Conf. Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Wroclaw, pp. 245-254.

Hasan, M. K.H. A. Rubaiyeat, H. A., Lee, Y.K. and Lee, S., 2008. A reconfigurable HMM for activity recognition, in *Proc. 10th Int. Conf. Advanced Communication Technology*, Phoenix, pp. 843-846.

Du, Y. T., Chen, F., Xu, W. L., and Zhang, W. D. , 2008. Activity recognition through multi-scale motion detail analysis, *NEUROCOMPUTING*, vol. 71, no. 16-18, pp. 3561-3574, Oct.

Park, S. and J. K. Aggarwal, J.K., 2003. Recognition of two person interactions using a hierarchical Bayesian network, in *Proc. 1st ACM SIGMM Int. Workshop Video Surveillance*, Berkeley, CA, pp. 65-76.

Muncaster ,F. and Ma, Y. Q. , 2007. Activity recognition using dynamic Bayesian networks with automatic state selection, in *Proc. IEEE Workshop Motion and Video Computing*, Austin, TX, pp. 30-30.

Buccolieri, F. , Distante, C. and A. Leone,A., 2005. Human posture recognition using active contours and radial basis function neural network, in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, Como, pp. 213-218.

Zhu, G. F. , Xu, C. S., Gao, W. and Huang, Q. M. , 2006. Action recognition in broadcast tennis video using optical flow and support vector machine, in *Proc. Workshop Human-Computer Interaction (HCI)*, Graz, pp. 89-98.