Typical Sequences Extraction and Recognition*

Gengyu Ma and Xueyin Lin

Key lab on pervasive Computing, Ministry of Education, Computer Science Department
Tsinghua University, Beijing, China, Postcode 100084
Tim99@mails.tsinghua.edu.cn, lxy-dcs@mail.tsinghua.edu.cn

Abstract. This paper presented a temporal sequence analyzing method, aiming at the extraction of typical sequences from an unlabeled dataset. The extraction procedure is based on HMM training and hierarchical separation of WTOM (Weighted Transition Occurring Matrix). During the extraction, HMMs are built each for a kind of typical sequence. Then Threshold Model is used to segment and recognize continuous sequence. The method has been tested on unsupervised event analysis in video surveillance and model learning of athlete actions.

1 Introduction

Recently research on developing new technology of Human Computer Interactive (HCI) has been a hot topic in information technique area, and understanding the human actions, such as facial expression, gesture and body movement etc., automatically by computer has attracted more attention than before. Since human activities are usually the dynamic signal sequences, and different categories of time sequences express different meaning, time sequence clustering is usually the first step for recognizing and understanding human activity. For example, if we want to know the gesture's meaning, its category should be recognized first. In this paper a novel method of extracting typical sequences from a mixture set of sequences is presented. A typical sequence means one kind of sequence which frequently occurs in more and less similar manner, such as typical gestures in human communication and typical activities of a badminton player. Extracting typical sequences automatically can help computer to understand the events and environments. Typical sequences analysis can also be used to detect atypical sequences, which is a hot topic in video surveillance field. It is evident that typical sequence extraction is the problem of data clustering. Therefore the criterion of similarity evaluation between different time sequences should be defined and the clustering strategy should be developed.

A sequence is firstly a dataset of observation follows some distribution. [11] uses cross entropy to measure the distance between two dataset, and [17] uses principal angle to measure the distance. In [17], the method is used to analyze the sequence of people movement and sequence of head tracking images. But this kind of methods

* This research is supported partially by the National Science Foundation of China (No.69975009) and National Key Basic Research and Development Program of China (No.2002CB312101)

N. Sebe et al. (Eds.): HCI/ECCV 2004, LNCS 3058, pp. 60-71, 2004.

ignores the temporal information in the sequences, and just treats the sequence as discrete observations.

Another kind of methods treats sequences as continuous observations. DTW (dynamic time warping) is a method widely used. It measures the distance between two sequences in the clustering method of [2]. But it is only applicable under the condition of small time warping, such as hand writings or curves.

In many complex fields such as speech recognition, HMM has a better performance than DTW [3]. HMM is a stochastic model which describes the probabilities of hidden state transition and observation [4]. Training of HMM, however, needs large amount of data. So it can not be directly used on a single short sequences, such as motion trajectories or image sequences.

HMM can measure the similarity of a sequence respective to a set of sequences. One framework of model building based on HMM adopts the K-mean strategy: First an initial clustering is built, and an HMM is trained for each cluster. Then the cluster membership of each sequence is reassigned based on which cluster's HMM can produce the largest probability. The training and reassigning steps continue until the system parameter converges. The problem of such a framework is that the final result is sensitive to the initial grouping. Most of the researches, adopting such a framework, concentrate on developing a strategy for getting a good initial clustering. Oates, etc. [2], for instance, used DTW in the initial step, and Smyth [5] just randomized the initial clustering.

In our clustering method, however, HMM is used in another way. Similar to [6][7], entropy minimization criterion is adopted for model building. It is well known that entropy can be used to measure the certainty of a model, a compacter model has less uncertainty, and hence a smaller entropy. Entropy minimization is also a theoretical explanation of MDL (minimum description length), which prefer the model that can represent the data with shorter coding length [12, 13]. In [6, 7], entropy minimization is achieved by trimming transitions with small probabilities. In our method, however, entropy reduction is mainly based on separating sequences into clusters. A hierarchical splitting framework is adopted to reduce the interaction between different kinds of sequences. The model building procedure proceeds recursively as follows: A HMM model is trained based on the whole dataset first. Then TOM (Transition Occurring Matrixes) and weighted TOM – WTOM, two features defined in section 2, are calculated for each sequence. Then Normalized Cut [10] is used to split the whole dataset into two clusters. Meanwhile, the entropy of the whole system is reduced. The splitting procedure will continue until all the data in each cluster share the same TOM, and each cluster gains a unique HMM model. A continuous sequence can be segmented into typical and atypical sequences automatically.

The remainder of this paper is organized as follows. The formula of the entropy of an HMM and the structure optimization principle is addressed in section 2. The hierarchical clustering procedure is discussed in section 3. The continuous sequence segmentation and recognition method is presented in section 4. The experiment result, including the clustering of hand gesture, surveillance video, and athlete activity are in section 5. Section 6 is the conclusion.

2 Structure Optimizing Rule

It is well known that entropy can be used to measure the amount of "disorder" of a model, defined as

$$H(p) = \int_{X\in\Omega} -p(X)\ln p(X)dX \tag{1}$$

Where X lies in space Ω, and its distribution is p(X).

2.1 Entropy of a HMM

The parameter of HMM includes three parts: transition matrix $A=(a_{ij})=(P(q_{t+1}=s_j|q_t=s_i))$, which is the transition probabilities among hidden states, prior distribution $\boldsymbol{\pi}=(\pi_i)=(P(q_1=s_i))$, which is the distribution of hidden state at the begin of the sequence, and observation distribution $B=(b_i)=(P(o_t|q_t=s_i))$. Given the HMM parameter, the uncertainty is embedded in its hidden state and observation sequence, so the entropy of it can be calculated as follows.

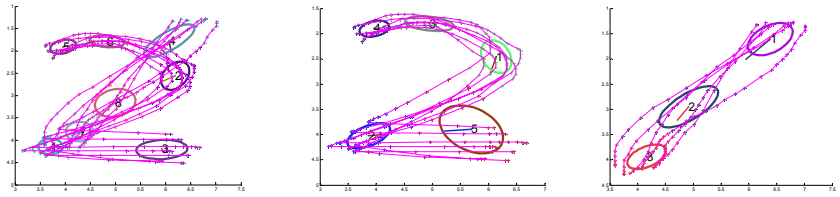
$$\begin{aligned}
&H(P(O,Q))\\
&=-\sum_{q_t\in\{s_1,\ldots,s_L\}}\int_{o_t\in\Omega}P(O,Q)\log P(O,Q)dO\\
&=-\sum_{i=1}^{N}\pi_i\log\pi_i-\sum_{i=1}^{N}\sum_{j=1}^{N}\left(a_{ij}\log a_{ij}\sum_{t=1}^{L}P(q_t=s_i)\right)\\
&-\sum_{t=1}^{L}\sum_{i=1}^{N}\int_{o\in\Omega}P(q_t=s_i)P(o\,|\,q_t=s_i)\log P(o\,|\,q_t=s_i)do\\
&\approx-\sum_{i=1}^{N}\pi_i\log\pi_i-\sum_{i=1}^{N}\sum_{j=1}^{N}\left(t_i a_{ij}\log a_{ij}\right)\\
&-\sum_{i=1}^{N}\left(t_i\int_{o\in\Omega}P(o\,|\,q_t=s_i)\log P(o\,|\,q_t=s_i)do\right)\\
&=H(\pi)+\sum_{i=1}^{N}t_i H(P(q_{t+1}\,|\,q_t=s_i))+\sum_{i=1}^{N}t_i H(P(o\,|\,q_t=s_i))
\end{aligned} \tag{2}$$

The entropy of a HMM is composed of the entropy of its initial distribution, the weighted sum of the entropies of transition probabilities, and the weighted sum of the entropies of observation distributions.

Actually, the entropy of the traditional trained HMM model does not reach the minimum value. In fact, traditional Baum-Welch training method only trains the transition and observation probabilities under given state number and assigned model structure. So the HMM after training is only one of the local minimums. In our method, we train not only the transition probabilities between them, but the hidden state number as well. We train the whole HMM structure, defined as the hidden state number and the transitions between them, according to some rules explained in next section. As a result, the entropy can be reduced towards the global optimum.

2.2 Structure Optimizing Rule

It is well known that HMM is a first order Markov process and hence can not precisely describe complicated situations. Actually high order Markov property can be expressed by dividing a single Markov model into several simpler ones.



(a) (b)

Fig. 1. Express high order Markov process by separated models. (a) trained together; (b) trained separately

For example, in Fig.1, hand writing sequences "1" and "2" are trained together to built a HMM. In the model, hidden state 4 can both transfer to 2 or 8, because the model didn't know where the sequence ends. Evidently this structure does not reflect the possible state transitions properly. Some information lost in this model. But using two models, this information can be represented, Fig.1(b).

In fact, transition 4→2 conditional depend with the transition 1→3. If a sequence includes 4→2, it must also include 1→3. Therefore uncertainties in the transition of hidden states can be further reduced and hence the entropy. In model (b), the complicated transition path has been explicitly expressed by two branches, and uncertainty is just the prior distribution. In short words, in HMM, the first order Markov assumption can not represent the relations between transitions. Our way of overcoming this is separating the models as shown in (b). Each kind of stroke order has a unique model, so the uncertain transition is removed.

Based on these observations we introduce the structure optimizing rule:

Rule 1: If a transition does not occur in all samples, separate the data into two clusters according to the occurrence of this transition.

To implement this algorithm, we introduce the TOM (Transition Occurring Matrix). For a sequence, if a hidden state transition from i to j occurs, the entry (i,j) in TOM is 1, else it is 0. So rule 1 can be explained as: **all sequences in a cluster have the same TOM.**

In Fig.1(a), there are three kinds of TOMs.

$$\begin{pmatrix} 1 & 1 & & & \\ & 1 & & & \\ & & 1 & & \\ & 1 & 1 & & \\ & & & 1 & 1 \\ & & & & 1 \\ 1 & & & 1 & \\ & & & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & 1 & & & 1 \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ & & & & 1 \\ 1 & & & 1 & \\ & & & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ & & & & 1 \\ 1 & & & 1 & \\ & & & 1 & 1 \end{pmatrix}$$

But in Fig1.(b), each cluster has a single TOM.

2.3 Structure Optimization and Entropy

In order to explain the situation mentioned above more clearly, a general situation is shown in Fig.2, in which some sequences pass the hidden states 2-3, while others pass 2-4, and they all pass states other than 2,3, and 4 ('others' in the Fig.). If the model of

4(a) is split into two models as shown in Fig.2(b), the entropy of either models can be calculated as follows.

Suppose that the prior probabilities of these two types of sequence are (P_1,P_2), the average durations at state 2 are (T_1,T_2), and all parameters besides state 2 are the same for these two kinds of sequences. Remembering that for s=1,2,5,…,N, $t_s=P_1t_{s*}+P_2t_{s'}$, and $t_3=t_{3*}$, $t_4=t_{4'}$ are always satisfied, so the entropies due to the observation are always the same. The only difference of the entropy is due to state 2 and the prior distribution.

In (a), entropy related to the initial distribution is 0, in (b) it is $H([P_1,P_2])= P_1\ln P_1 + P_2\ln P_2$.

The parameters in model (a) can be estimated as

$$a_{23}=\frac{P_1}{P_1\cdot T_1+P_2\cdot T_2},\ a_{24}=\frac{P_1}{P_1\cdot T_1+P_2\cdot T_2},\ \text{and } a_{22}=1-a_{23}-a_{24}.$$

The entropy of the transitions at state 2 in (a) is

$H(P(q_{t+1}|q_t=2)) = H([a_{22}, a_{23}, a_{24}])$.

And the expected occurring time t_2 is $t_2=P_1T_1+P_2T_2$.

In (b) this part of entropy is

$$H(P(q_{t+1}\mid q_t=2^*))=\frac{1}{T_1}\ln\frac{1}{T_1}+\frac{T_1-1}{T_1}\ln\frac{T_1-1}{T_1}$$

$$H(P(q_{t+1}\mid q_t=2'))=\frac{1}{T_2}\ln\frac{1}{T_2}+\frac{T_2-1}{T_2}\ln\frac{T_2-1}{T_2}$$

And the corresponding occurring times are T_1 and T_2 as assumed above.

So the entropies to be compared are:

$$H(a)=(P_1T_1+P_2T_2)*\ln(H(P(q_{t+1}|q_t=2)));$$

$$H(b)=P_1\ln P_1+P_2\ln P_2+T_1P_1H(P(q_{t+1}|q_t=2^*)) +T_2P_2H(P(q_{t+1}|q_t=2'))$$

The functions have 3 variables: T_1, T_2 and P_1. In Fig.3, P_1 is set to 0.5, and the functions are drawn with respect to T_1 and T_2.

For different P_1, the entropy of model (b) is always smaller than or equal to that of model (a). In fact, H(a)=H(b) when $T_1=T_2$. Please notice that in this section only the entropy due to the hidden state transition is concerned, so even the observation distribution is not retrained, this structure modification will reduce the entropy.

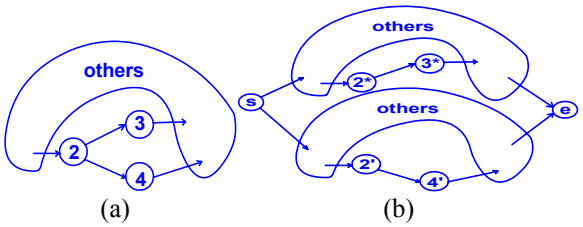
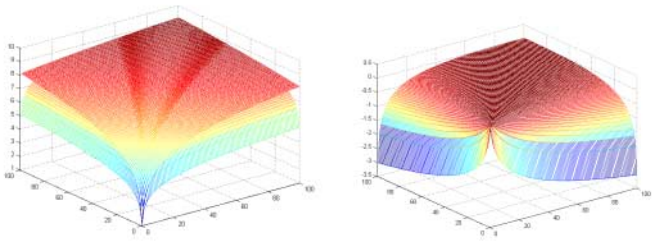


Fig. 2. (a) HMM before modification; (b) HMM after separation



(a) (b)

Fig. 3. (a) The entropies of HMM a and b; (b) H(b)-H(a); Axes x and y are T1 and T2

3 Hierarchical Separations

In our method a hierarchical framework is employed and the mentioned structure optimizing rule is used to split the whole dataset into clusters recursively. The reason of using a hierarchical framework will be explained in section 3.1. Besides, another matrix called weighted Transition Occurring Matrix (WTOM) is defined. The introduced weight is used to evaluate the Gaussianity of each hidden state to make the hierarchical separation procedure more robust.

3.1 Hierarchical Separation

The entire clustering can not directly be based only on the TOMs calculated from the HMM trained from the whole dataset. Since the whole dataset is a mixture of several different kinds of sequences, the cross-interactions between different data and noises prevent this HMM to represent each kind of data well. The handwriting “1” and “2” for example, in the original model, there are totally three kinds of TOMs.

The main advantage of using a hierarchical separation framework is that the corresponding HMM can be updated recursively. By hierarchical separation, the parent dataset are divided into clusters. Data sharing greater similarities are usually assigned within the same cluster. In the process of hierarchical separation, data in each cluster become simpler, their HMM structures and calculated WTOMs become more reliable. Near the leaves of the hierarchical tree, even one difference on TOMs can distinguish two clusters. Using a hierarchical framework, the separation process of handwriting “1” and “2” is as follows. Given the three kinds of TOMs, the TOMs are split into two classes. The difference between TOM 1 and TOM 2 is 4, difference between TOM 2 and TOM 3 is 2, and the difference between TOM 3 and TOM 1 is 4. So the best separation is 1 vs. 23. After this separation, all data are separated into two clusters. For each cluster, a HMM is trained, thus the model in Fig.1(b) is got. In each cluster, all data have the same TOM, so the hierarchical separation ends. Till now, the condition in optimizing rule is fulfilled, so the entropy is minimized.

The hierarchical clustering process is shown in Fig.4.

In the hierarchical separation procedure, a hierarchical tree is built. Each node in the tree corresponds to a cluster and its HMM in the separation process. The root node represents all the training data, and the HMM trained on all the data is called global

HMM. Each node besides the leaves has two child nodes, which are the separation result of this node. In each leaf of the tree, all data have the same TOM. All the clusters embedded in the leaves make up the separation result of the training data.

In the separation procedure TOM is used as the feature of each sequence. Since each TOM is really a pattern, Normalized Cut [10] method can be employed to divide the whole set into two subsets. NCut is a graph cut method. Given a similarity matrix, NCut method finds the separation result with the maximized normalized interclass similarity. In this application, the similarity matrix can be calculated conveniently from the TOMs (and WTOM, which will be introduced later).

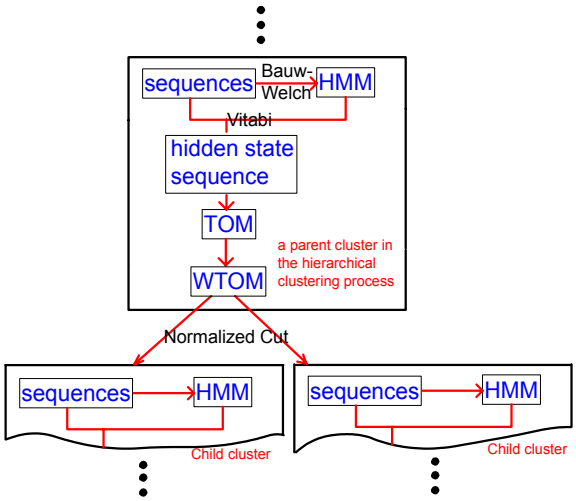


Fig. 4. The hierarchical clustering process

3.2 TOM and WTOM

As mentioned before, TOM is the feature of a sequence that describes its hidden states and their transitions. Since TOMs are calculated from the corresponding HMM, HMM training should be performed first. In this step, the method presented by [16] is employed, which automatically determines the hidden states number of a HMM according to entropy minimization criterion.

Then TOMs are calculated by the Vitabi algorithm. Although TOMs can be used in separation directly, using weighted TOM can make the result more robust. The weight is adopted to judge "whether the transition is good", or "whether the data in the training sample support the distribution". If so the weight value of the corresponding transition will be higher. The weighting value is measured by the evaluation of the Gaussianity of the two states relative to the transition.

Where h_i is the entropy of training data and g_i is the expected entropy of the ideal Gaussian distribution.

For a certain hidden state i, if its observation follows a Gaussian distribution, its entropy will be bigger, so the distribution is more greatly supported by the data. In the WTOM, a state with bigger entropy will has a greater weight value in the computation of similarity matrix.

In the separation process, TOM is used as the stopping criterion, and WTOM is used to calculate similarity matrix used in the NCut algorithm.

$$\begin{aligned} \mathrm{WTOM} &= (t_{ij} w_{ij}) = \left(t_{ij} \exp(h_i - g_i + h_j - g_j)\right) \\ h_i &= -\int P(x) \log P(x) dx \approx -\frac{1}{N} \sum_{t: s_t = i} \log P(o_t \mid \theta_i) \\ g_i &= H(N(\mu, \Sigma)) = \frac{d}{2} + \log((2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}) \end{aligned} \tag{3}$$

4 Typical Sequence Extraction and Recognition

From the hierarchical separation tree the typical sequences in the training dataset can be obtained. Typical sequence is defined as the kind of sequence that occurs frequently. In the separation result of the training data, a cluster with many samples corresponds to a typical sequence. In the separation procedure, not only typical sequences are found out, for each cluster, the corresponding HMM can also be built. Therefore it can be used to recognize typical sequences from a continuous input sequence.

A continuous sequence usually is a mixture of some typical sequences and some atypical sequences connected together. Therefore some method of recognizing atypical sequence should be included. In our method the entire model is composed of some typical sequence models and a Threshold Model [18]. Each of the first N models is trained by the data in a cluster, and corresponds to a typical sequence. The threshold model corresponds to the atypical sequences. In [18], the Threshold Model is composed of all the hidden states in typical models, their observation probabilities are preserved, but the transitions are set equally, $a_{ij}=a_{ik}=(1-a_{ii})/(N-1)$, $j \neq i$ and $k \neq i$, where N is the number of hidden states. This configuration means no restriction on the transitions between hidden states, so atypical sequence will has a higher probability than that of typical models. In this paper, the atypical sequences are already in the training dataset. So the global model can be directly used as Threshold Model.

While a sequence is given as input, the Vitabi method will find the best suitable hidden state path of the input observation, thus also tells the most probable arrangement of typical sequences and atypical sequences.

5 Experiment Results

The method presented in this paper is a universal method which can be applied on the analysis of many temporal sequences. The hierarchical separation method has been tested on handwriting digits, hand gesture image sequences, human movement, and other sequences. In this section only the experiments on surveillance video and athlete action is demonstrated here for page limitation.

5.1 Typical Event Extraction

The first example demonstrates the usage of our model building method on event extraction from the surveillance video of the entrance lobby of our laboratory, captured by a still surveillance camera. The scene is shown in Fig.5. The entrance is on the left side and a room can be seen inside the door with a phone on the table near it. The lab is beneath the image. The root HMM is shown in Fig.6, and the individual event models are shown in Fig.7, (a) - (i) respectively, with their corresponding meanings. Some abnormal situations are shown in (j) as they seldom occur and can not be recognized as a meaningful cluster.

Sometimes, the atypical sequences should be paid more attention than typical sequences, because atypical sequences are not common events. In this dataset, there are 6 atypical sequences: room → phone → room, room → computer → room, lab → computer → lab, entrance → phone → entry, and two people chatting in the lobby and then go out. The computer runs the surveillance system. And the entry → phone → entry may be produced by a stranger who is not working here. By tradition method, sequences about the computer can be detected because they all passed the ‘computer' state. But the entry → phone → entry sequence is not such lucky, because both the transitions ‘entry → phone' and ‘phone → entry' have high probabilities.

Fig. 5. Surveillance Scene

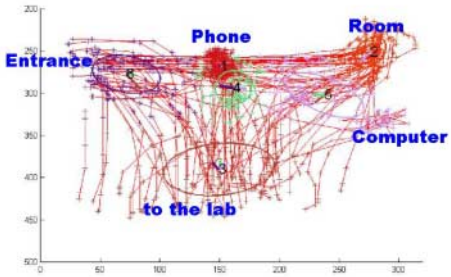


Fig. 6. Parent HMM at the root node of the hierarchical separation tree

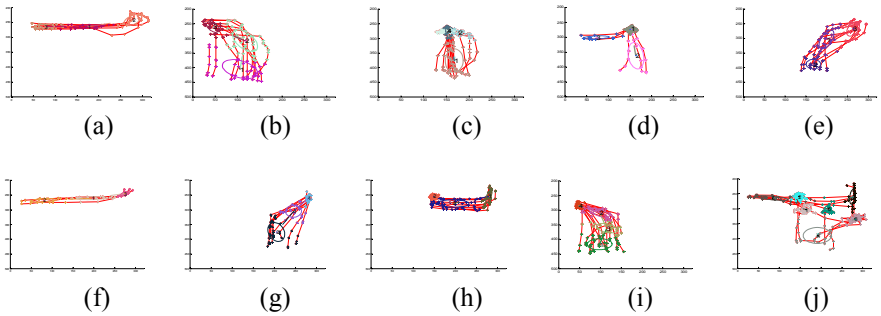


Fig. 7. Extraction result (a) entrance → room; (b) entrance → laboratory; (c) lab → phone → lab (d); lab → phone → entrance; (e) room → lab; (f) room → entrance; (g) lab → room; (h) room → phone → room; (i) lab → entrance; (j) others

5.2 Badminton Player Action Extraction and Recognition

We have used our method for badminton video analysis to extract player's typical actions. In this experiment a badminton video is pre-processed first, so that the player's shape can be segmented from the background. Fig.10 is some action sequences, composed of small player images. Each image was first resized to a normal size; then mapped to a low dimensional space by projecting to a 3D space spanned by the first 3 eigenvectors by means of PCA. Since there are shadows in some of the images, only the upper 2/3 part of an image is concerned. Thus the video is described by some 3D continuous vector sequences. Each sequence corresponds to a shoot in the video, and maybe consists of one or more actions. These long sequences are cut at the points where the athlete shape changes drastically, so that each segment corresponds to a single action.

After the above preprocessing, these action sequences make up the training sample of our sequence clustering algorithm. The training result is shown in Fig.9, each sequence of points represents a typical action. Different kinds of actions are drawn in different color. Each thick line is drawn manually to express the trends of a corresponding typical sequence. To be clear, 9 typical sequences are divided into two groups drawn in two images separately. The x and y axis are the first 2 components in PCA. The corresponding image sequences are shown in Fig.10. From the experimental results it can be seen that, some extracted typical actions, such as smashing, leaning left and leaning right, have explicit semantic meaning. However, there are also some actions that are not so typical to human's opinion. They are usually related to the pre-processing noise. For instance, sequence 1 includes the noise produced by another player's feet; and sequence 4 dues the wrong segmentation.

Finally, we got 9 meaningful actions and integrate them with a threshold model to recognize a continuous input sequence. Fig.9 is the segmentation result. The first frame of each sequence is shown in the figure, the last frame of a sequence just similar to the first frame of the next sequence. The type of each sequence is labeled above each sequence. Some trivial actions are recognized as atypical sequences, labeled as '0'. Six typical sequences are extracted from the video, with one "smashing start" and one "smashing end", two "lean right begin", and two "lean right end" actions. The corresponding frame number is indicated beneath of that image. Actually atypical sequences occupied the main part of the entire video, corresponding to the situation when the player pays attention to his opponent.

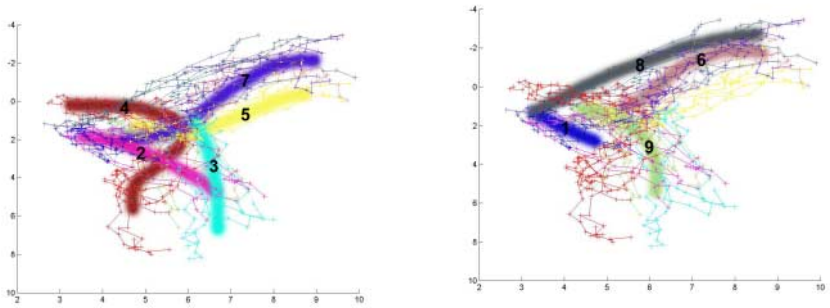


Fig. 8. Typical sequences in the badminton athlete action sequences. (in eigen space)

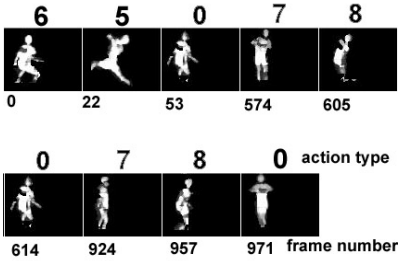


Fig. 9. The segmentation result of another continuous action sequence

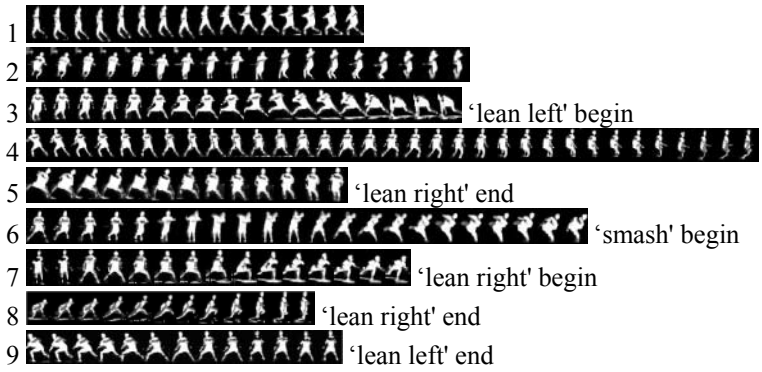


Fig. 10. Typical actions of badminton athlete

6 Conclusion

We proposed a new method to extract typical sequences non-surveillantly. The extraction procedure is a HMM based hierarchical separation procedure. After HMM is trained, TOMs and WTOMs are calculated and used as the features in separation. TOM gives a criterion on when the hierarchical separation process will stop, and WTOM makes the method more robust. Finally the model of the entire system is built by combining the typical models and a Threshold Model. This model is successfully used in the segmentation and recognition of continuous sequences.

References

[1] Berkhin, Pavel.: Survey of Clustering Data Mining Techniques. In: http://citeseer.nj.nec.com/berkhin02survey.html.

[2] Oates, Tim, Laura Firoiu, Paul Cohen.: Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. IJCAI, Working Notes, (1999) 17-21.

[3] Frank Höppner.: Time Series Abstraction Methods - a Survey. Proceedings of GI Jahrestagung Informatik, Workshop on Know- ledge Discovery in Databases, (2002) 777-786,

[4] Awrence R. Rabiner.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, no. 2, Feb. (1989) 257-285
[5] P. Smyth.: Clustering Sequences with Hidden Markov Models, Advances in Neural Information Processing, MC Mozer, MI Jordan, T. Petsche, Eds. Cambridge, MA, MIT Press, (1997) 648-654
[6] Matthew Brand.: Pattern Discovery via Entropy Minimization. Uncertainty 99 (AI & Statistics) (1999)
[7] Matthew Brand.: An Entropic Estimator for Structure Discovery. NIPS, (1998) 723-729
[8] Matthew Brand, Aaron Hertzmann.: Style machines. SIGGRAPH (2000)
[9] Aapo Hyvarinen and Erki Oja.: Independent Component Analysis: a Tutorial. http://www. cis.hut.fi/~aapo /papers/IJCNN99_tutorialweb/ (1999)
[10] J. Shi and J. Malik.: Normalized Cuts and Image Segmentation. Computer Vision and Pattern Recognition, (1997) 731-738
[11] E. Gokcay, J. Principe.: Information Theoretic Clustering, PAMI, Feb. (2002)
[12] Stine. RA.: Model Selection Using Information Theory and the MDL Principle. http://www-stat.wharton.upenn.edu/~bob/research/smr.pdf
[13] P.M.B. Vitanyi, M. Li.: Ideal MDL and its Relation to Bayesianism. Proc. ISIS: Information, Statistics and Induction in Science World Scientific, (1996) 282-291
[14] Anne Lorette, Xavier Descombes, Josiane Zerubia.: Fully Unsupervised Fuzzy Clustering with Entropy Criterion. ICPR, (2000)
[15] C. Li, G. Biswas.: Improving Clustering with Hidden Markov Models Using Bayesian Model Selection. International Conference on Systems, Man, and Cybernetics, vol. 1, (2000) 194-199
[16] Yoram Singer, Manfred K. Warmth.: Training algorithms for Hidden Markov Models using entropy based distance functions. NIPS, (1996) 641-647
[17] Lior Wolf, Amnon Shashua.: Kernel Principal Angles for Classification Machines with Applications to Image Sequence Interpretation.
[18] Hyeon-Kyu Lee, Jin H.Kim.: An HMM Based Threshold Model Approach for Gesture Recognition, PAMI, Oct. (1999)