

PAPER

Motion Pattern Study and Analysis from Video Monitoring Trajectory

Kai KANG^{†,††}, *Nonmember*, Weibin LIU^{†,††a)}, *Member*, and Weiwei XING^{†††}, *Nonmember*

SUMMARY This paper introduces an unsupervised method for motion pattern learning and abnormality detection from video surveillance. In the preprocessing steps, trajectories are segmented based on their locations, and the sub-trajectories are represented as codebooks. Under our framework, Hidden Markov Models (HMMs) are used to characterize the motion pattern feature of the trajectory groups. The state of trajectory is represented by a HMM and has a probability distribution over the possible output sub-trajectories. Bayesian Information Criterion (BIC) is introduced to measure the similarity between groups. Based on the pairwise similarity scores, an affinity matrix is constructed which indicates the distance between different trajectory groups. An Adaptable Dynamic Hierarchical Clustering (ADHC) tree is proposed to gradually merge the most similar groups and form the trajectory motion patterns, which implements a simpler and more tractable dynamical clustering procedure in updating the clustering results with lower time complexity and avoids the traditional overfitting problem. By using the HMM models generated for the obtained trajectory motion patterns, we may recognize motion patterns and detect anomalies by computing the likelihood of the given trajectory, where a maximum likelihood for HMM indicates a pattern, and a small one below a threshold suggests an anomaly. Experiments are performed on EIFPD trajectory datasets from a structureless scene, where pedestrians choose their walking paths randomly. The experimental results show that our method can accurately learn motion patterns and detect anomalies with better performance.

key words: *visual monitoring, trajectory clustering, Hidden Markov Models, hierarchical clustering, abnormal detection*

1. Introduction

Recently, video surveillance has a wide range of applications in our daily life [1]. Wherever we go, a busy train station, a parking lot or a shopping mall, there are hundreds of cameras recording our behaviors. With the development of the video surveillance system, more and more videos have been produced [2], while most visual surveillances rely on a human operator. No matter how vigilant the operators are, a significant number of potentially important events might be overlooked because of unavoidable information overload. So, the reliability of automated detection systems has become a primary issue [3]. The goal of our work is to learn activities and interactions in a complicated

and crowded scene [4].

Our aim is to provide visual surveillance systems with an activity-based scene model that supports behavioral understanding and analysis of the observed activity. We expect a visual surveillance system to discover typical types of activities (e.g., pedestrian makes a U-turn) in these scenes and then provide a summary of them; to detect abnormal activities, e.g., pedestrians with high-speed or strangle behaviors; to learn interactions modeled as distribution over global topics; and to predict behavior in the same group. These applications have been paid more and more attention in the human-computer interaction and virtual reality fields.

Learning motion pattern from trajectories has attracted the attention of computer vision researchers. Most of the existing learning methods are based on trajectory space attribute. Some authors presented a similarity measure for automatically identifying important spatial structure of traffic intersection using trajectories of vehicles [5]. Others clustered foreground pixels using a fast accurate fuzzy K-means algorithm. The cluster centroids of foreground pixels ensure that each cluster centroid is associated with a moving object in the scene [6]. The methods mentioned above have a common defect, which is that the trajectory clustering cannot be represented as a motion pattern which obeys uniform distribution.

To solve these problems, we must determine how to model activities and interactions in crowded scenes. In this paper we propose a Hidden Markov Model (HMM) method for learning motion pattern, as shown in Fig. 1.

- Trajectories are translated into smooth and continuous curve, and then segmented into sub-trajectories with the same length, the sub-trajectories are then quantized to a codebook;
- Every trajectory group can be modeled by a HMM, which represents the relationship between observations and states. The states of trajectory are hidden which are connected by state transition probabilities;
- An affinity matrix can be constructed to represent dissimilarity measures between different models. The trajectories or trajectory groups with smaller distance are merged and an Adaptable Dynamic Hierarchical Clustering (ADHC) tree will be built by loop;
- Based on the method of HMM and ADHC, trajectory motion patterns can be learned automatically. Using the affinity matrix above, we can detect unusual events by comparing the conformity scores.

Manuscript received October 24, 2013.

Manuscript revised January 25, 2014.

[†]The authors are with Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China.

^{††}The authors are also with Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China.

^{†††}The author is with School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China.

a) E-mail: wbliu@bjtu.edu.cn

DOI: 10.1587/transinf.E97.D.1574

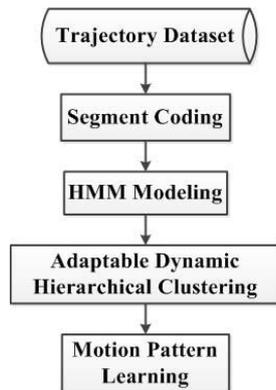


Fig. 1 Overview of motion pattern learning system.

The rest of the paper is organized as follows. Section 2 surveys related works on trajectory representation and HMM analysis. Section 3 explains the formation of HMM and affinity matrices, based on which we can construct the ADHC tree and detect the abnormal events. Section 4 discusses the experimental results of learning motion pattern and detecting anomaly from a Forum scene. In Sect. 5, we present a brief summary and conclusion with an outline of future research in this area.

2. Related Works

Over the last few years, a number of trajectory clustering techniques have been developed and applied to various system models for transient stability assessment. B. Morris & M. Trivedi [7] had enumerated a series of trajectory-based activity analysis for visual surveillance. They not only presented a survey of trajectory-based activity analyses for visual surveillance, but also described techniques that used trajectory data to define a general set of activities that were applicable to a wide range of scenes and environments.

The most direct way for trajectory cluster is based on spatial information. Imran N. Junejo [8] applied Hausdorff distance to compare different trajectories and calculated the edge weights of the similarity matrix. S. Atev [4] used standard K-means and its soft variant fuzzy C-means (FCM) as their techniques, they also presented a similarity measure which was suitable for use with spectral clustering in problems. However, they required all trajectories be normalized to a fixed length. N. Sumpter [9] and W. Hu [10] presented a novel approach to learning spatial-temporal patterns of objects in image sequences, using a neural network paradigm to predict future behavior. W. Hu [11] also introduced the fuzzy Self-Organizing Neural Network based method and make the learning process much more efficient. C. Stauffer [12] and X. Wang [13] put forward Co-Occurrence Decomposition, where trajectories were viewed as a bag of words where similar bags contain similar words. A co-occurrence matrix was formed from training data and decomposed to build document subjects (routes).

David Biliotti [14] developed hierarchical clustering method and provided tree structure for clustering dif-

ferent resolutions. There were two main hierarchical clustering variants, agglomerative [15] and divisive [16], which defined similarity relationships between trajectories in a tree-like structure following a bottom-up or top-down procedure. Using the concept of minimum average distance between machines oscillations exhibiting a common behavior, Carlos Juarez [17] proposed an online a hierarchical clustered structure of the system that can be used for on-line determination of multi-machine dynamic equivalents. Fan Jiang [18] referred to a process of dynamic hierarchical clustering (DHC), which differed from typical agglomerative hierarchical clustering in that the data reclassification process and the model retraining process were incorporated into each clustering step.

M. Bennewitz [19] presented a visual activity analysis module based on vehicle tracking, and encoded the spatiotemporal motion characteristics of these motion paths by a hidden Markov Model (HMM). L. Kratz [20] improved his framework by constructing a coupled HMM that models the spatial relationship of motion patterns surrounding each video region. He used this method to model the stationary structure of motion patterns in the video and identified a typical event as statistical anomalies. Faisal I. Bashir [21] presented a framework to estimate the multivariate probability density function based on PCA coefficients of the sub-trajectories using Gaussian mixture models (GMMs). N. Suzuki [22] employed HMM to model time-series features of human positions and then a similarity matrix of HMM mutual distances was formed. F. Porikli [23] and T. Xiang [24] also employed affinity matrix and eigenvector decomposition, but one for detection of usual events and another for dimension reduction on the N dimensional feature space. H. Zhou [25] combined multiple video streams in the inference level, with a Coupled Hidden Markov Model (CHMM), N. Oliver et al. [26] compared HMMs with CHMMs for modeling interactions like following and meeting, and showed that CHMMs are more efficient and accurate. F. Jiang [27] used a dissimilarity measure between time series based on the Bayesian information criterion (BIC), which confirmed to be more accurate than usual dissimilarity process.

Our approach has reported some preliminary findings to trajectory clustering using HMM and ADHC. In following sections, we will present our model-based recognition system that model trajectory group by HMM, cluster trajectories by ADHC and detect abnormal by Affinity Matrix.

3. Motion Pattern Learning

A trajectory is a time series of coordinates representing the motion path of an object over its life-time, i.e. number of frames that object exists. The trajectories can be represented as $T = \{T_1, \dots, T_P\}$, where P is the number of trajectories. For the j^{th} trajectory, we adopt the notation $T_j = \{(x_1, y_1, t_1), \dots, (x_Q, y_Q, t_Q)\}$, where the Q represents the number of points, and t is time sequence.

In this work, we transfer the trajectory group feature

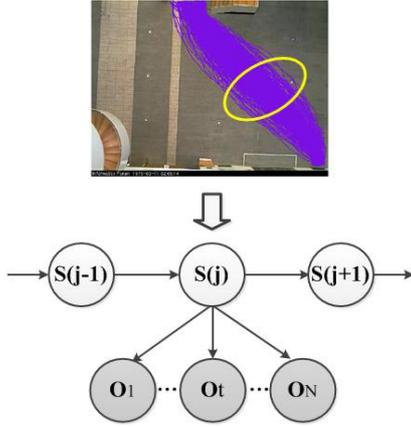


Fig. 2 Relationship between trajectory group and HMM. Every trajectory or trajectory group can be modeled as a HMM. The sub-trajectories group in yellow circle can be represented as a state which emits a series of observable outputs. The states transformed from one to another form a chain of HMM.

sequences into a parameter space λ which is characterized by a set of Hidden Markov Model (HMM) parameters [23]. The HMM is a probabilistic model composed of a number of interconnected states $S(j)$, each of which emits several observable outputs O_t . Every HMM is characterized by two probability distributions: the transition distribution over states and the output distribution over the output observations. The observable outputs $\{O_1, \dots, O_N\}$ will be encoded as a codebook. Since the states transition cannot be observed indirectly through the sequence of output symbols, and the states are said to be hidden. Figure 2 shows the relationship between trajectory group and HMM.

3.1 Trajectory Preprocessing

The first issue is how to solve the problem of point leakage. Since the trajectory sequence is ranked in chronological order, we can check the time series to see whether there is a point leakage. If there are L leaked points $\{1, \dots, L\}$, the i^{th} leaked point can be represented as the Different Value of the start point and the end point:

$$P_{leaked}(i) = P_{start} + \frac{i}{L+1}(P_{end} - P_{start}) \quad (1)$$

To get observable outputs of HMM, a continuous trajectory sequence T_j is automatically segmented into K sub-trajectory segments. The data of each sub-trajectory are concatenated into a dimensionality reduction vector and the set of sub-trajectories are processed for standardization subspace. F.I. Bashir [21] had represented a method of segmenting the trajectories at perceptually significant points of change, the spatial curvature of a 2-D curve. But the trajectories in our paper are structureless and they are too irregular to use the curvature-based segment.

To build a codebook, we have to get a uniform sampling of continuous trajectory. The Uniform Cubic B-splines has been proved to be a simple and effective method to obtain a continuous trajectory. The term

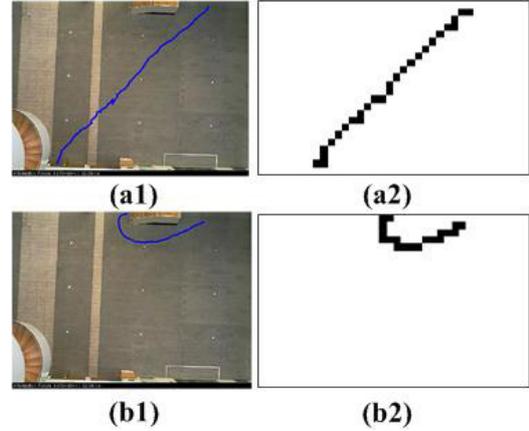


Fig. 3 Building codebooks of trajectories. (a1) and (b1) are original trajectories; (a2) and (b2) represent the codebook corresponding to them. For long trajectories as (a1), the character can be retained; for detailed trajectories as (b1), the particulars can be obvious.

“B-spline” is coined by Isaac Jacob Schoenberg and is short for basis spline [28]. Given $m+n+1$ points $\{P_0, \dots, P_{m+n+1}\}$ and the $m+1$ B-spline curves of degree n can be defined as:

$$S_{i,n}(t) = \sum_{k=0}^n P_{i+k} \cdot b_{k,n}(t) \quad (2)$$

where $t \in [0, 1]$, $i = 1, \dots, m$. The $b_{k,n}(t)$ is a B-spline basis function of degree n . Every B-spline curve can be confirmed by $n+1$ points. The basis B-splines of degree n can be defined as:

$$b_{k,n}(t) = \frac{1}{n!} \sum_{j=0}^{n-k} (-1)^j C_{n+1}^j (t+n-k-j)^n \quad (3)$$

where $t \in [0, 1]$ and $i = 1, \dots, m$. The Uniform Cubic B-spline is the most commonly used form of B-spline. We get basis B-splines from Formula (3) when $n = 3$, the blending function can easily be calculated:

$$S_{i,3}(t) = \frac{1}{6} \cdot \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix} \quad (4)$$

where $t \in [0, 1]$ and $i = 1, \dots, m$.

To quantize position, the scene (640×480) is divided into cells of size (20×20). The motion of a moving pixel is uniformly sampled using Uniform Cubic B-splines and quantized into a codebook as shown in Fig. 3. We set the sampling number as 30. The trajectory character and detail can be represented clearly from the codebook in Fig. 3.

3.2 Hidden Markov Model

We represent the trajectory group pattern as a Hidden

Markov Model (HMM). The hidden states capture the transitive properties of the successive coordinates of the spatiotemporal trajectory, and the sub-trajectories which can be observed as distribution of different states are presented as output symbols. The states sequence that maximizes the probability is the corresponding model for the considering trajectory. We propose to use the HMMs for trajectory clustering and recognition applications.

In our work, the first parameter specified for the HMM is the number of states. For each group of trajectories represented by a specific HMM, we set the number of states as the number of sub-trajectories for that class.

When the number of states is fixed, the complete set of HMM parameters describing the model can be given as the following:

$$\lambda = \{A, B, \pi, M, N\} \quad (5)$$

where A denotes the state transition probability matrix, which is a matrix of size $M \times M$. Output matrix B denotes the output probability between state and observation, which is a matrix of size $M \times N$. Vector π is the probability of the given sub-trajectory state which is the first sub-trajectory among all the trajectories. M and N represent the states number and observations number respectively.

The HMM structure of a given trajectory groups is shown in Fig. 2. Every clear circle represents a random variable of a hidden state, and every shaded circle represents the observation. Each state has a probability distribution over the possible outputs. Each chain corresponds to one trajectory group and the following observations correspond to the sub-trajectories in this group. Every chain can be represented as a HMM.

Once the number of states is decided, the HMM parameter in Formula (5) will be estimated. For a given trajectory, we split it into M sub-trajectories. The state variable q_t corresponding to the t^{th} sub-trajectory is tacked from one of M values $q_t \in \{S_1, \dots, S_M\}$. Since we assume a Markov process, so the next probability distribution of q_{t+1} depends only on q_t . This is described by the above states transition probability matrix A whose element $a_{ij} = Pr(q_{t+1} = S_j | q_t = S_i)$ represents the probability that q_{t+1} corresponding to state S_j transmitted from q_t corresponding to S_i . The initial states probabilities are denoted as π_i , which represent the probability that q_1 equals to S_i . The observational data O_t for each state of the HMM corresponds to the state of the instant at t^{th} sub-trajectory, which is denoted by $b_j(o_t) = Pr(o_t | q_t = S_j)$. The structure of our HMM can be showed clearly in Fig. 2.

The parameters of the HMM can be initialized to random values, and then final parameter estimated by the forward-backward procedure [27]. To train the model maximizing the probability of the observation, we can choose a proper $\lambda = \{A, B, \pi, M, N\}$ such that $Pr[O|\lambda]$ can be maximized locally using an iterative algorithm such as Baum-Welch method [29]. Finally, we can get a set of reasonable parameters for HMM. This procedure is called the maximum likelihood estimate of HMM.

$$\begin{aligned} Pr[O|\lambda] &= \sum_M Pr[O|S, \lambda] \cdot Pr[S|\lambda] \\ &= \sum_{q_1 \dots q_M} \pi_{q_1} \cdot b_{q_1}(O_1) \cdot a_{q_1 q_2} \\ &\quad \cdot b_{q_2}(O_2) \cdot \dots \cdot a_{q_{M-1} q_M} \cdot b_{q_M}(O_M) \end{aligned} \quad (6)$$

3.3 Affinity Matrix

To represent similarity between different groups, investigators are used to analyze the average distance between the groups represented by the individual classifiers. Similarity of different HMMs can be measured using the cross likelihood ratio (CLR). Let i and j be two feature sequences, modeled by two HMMs λ_i and λ_j respectively. The dissimilarity between i and j can be defined as [18]:

$$d(i, j) = |L_i + L_j - L_i^j - L_j^i| \quad (7)$$

where L_i and L_j denote the likelihoods of i and j being generated by their own fitted models, i.e., $Pr(i|\lambda_i)$ and $Pr(j|\lambda_j)$, while L_i^j and L_j^i denote the cross likelihoods, $Pr(j|\lambda_i)$ and $Pr(i|\lambda_j)$ respectively. This method is appropriate when states and observations are simple, but this is not fit for our work.

To solve this problem, a dissimilarity measure between time series based on the Bayesian Information Criterion (BIC) was used in our work [27], where BIC is a statistical criterion for model selection.

$$BIC = -\log L + \frac{C}{2} \log N \quad (8)$$

where L is the likelihood of the estimated model as above, N is the number of observations, and C is the number of model parameters. For any two estimated models, the one with smaller value of BIC is the optimal. Our work is to compute the dissimilarity of different groups using BIC. To make work clearly, every trajectory in our database is modeled with the same structure by an HMM, that is with equal number of parameter C_k , and N HMMs are trained for N trajectory groups.

When trajectory group i and j are merged and modeled together by one HMM. The difference of BIC values for the two models i and j is equal to [18]:

$$\begin{aligned} d(i, j) &= BIC(ij, \dots) - BIC(i, j, \dots) \\ &= \log L_i + \log L_j - \log L_{ij} \\ &\quad - \frac{1}{2} C_K \log N \end{aligned} \quad (9)$$

where (i, j, \dots) denotes any two trajectories i, j compared to all the other trajectories, and L_t is the likelihood of trajectory t generated by the HMM trained by itself λ_t , $Pr(t|\lambda_t)$. (ij, \dots) denotes the trajectory group (merged i and j) compared to all the other trajectories, and L_{ij} is the likelihood of the new trajectory group ij generated by its HMM, $Pr(ij|\lambda_{ij})$. $d(i, j)$ measures the dissimilarity of two trajectory groups, and L refers to the likelihood for all trajectories in this group. If two trajectory groups are identical, the dissimilarity distance $d(i, j)$ will have a minimum value. The

smaller $d(i, j)$ is the greater tendency for i and j to merge.

For each feature, the $K \times K$ affinity matrix $D = \{d(i, j)\}$ provides a new representation of dissimilarity for the training dataset. To class database, agglomerative hierarchical clustering using BIC-based dissimilarity measure can be performed.

3.4 Adaptable Dynamic Hierarchical Clustering

Starting with any given individual trajectory, all possible pairs of trajectories are merged. To minimize the variability of trajectories within a cluster and maximize the variability between clusters, the Hierarchical Clustering is proved to be a convenient algorithm for our clustering.

The model-based similarity measure has an over fitting problem when given few training samples as mentioned above. This problem is true for our first several steps of the hierarchical clustering: when a cluster contains only a few trajectories (or starting from one trajectory), the trained HMM tends to be over fitted, and the dissimilarity measures between them are quite unreliable, which result in clustering errors. These errors will accumulative to the latter clustering steps.

To address this problem, we can update the clustering results at each merging step. In other words, when a new HMM is trained based on trajectory merging, all the trajectories in the database should be reclassified. In this way, some trajectories which have been incorrectly clustered at previous steps should be associated to other new HMMs. And then all the HMMs are retrained according to the updated trajectory classes. This updating process also corrects errors for later clustering steps. When clusters have gathered more samples, the trained HMMs become more reliable. This process above can be referred as Dynamic Hierarchical Clustering (DHC) [18].

Considering the original method of DHC cannot process the large quantities of trajectories effectively, we propose Adaptable Dynamic Hierarchical Clustering (ADHC), which is illustrated in Fig. 4. To simplify the hierarchical clustering process, an adaptable merging threshold was proposed. The threshold is determined by the current HMM parameter, which can be adjusted automatically. When the dissimilarity of pairwise trajectory groups is smaller than threshold, they will be merged. In this way, several trajectories can be clustered within one step, so the computation for hierarchical clustering will be greatly reduced.

$$d_{th} = \exp\left\{-\frac{L_{max} - L_{min}}{L_{max} + L_{min}}\right\}/\sigma \quad (10)$$

where L_{max} and L_{min} represent the maximum and minimum likelihood of this HMM, and σ is the accommodation coefficient.

The adaptable dynamic hierarchical clustering process is represented as bellow:

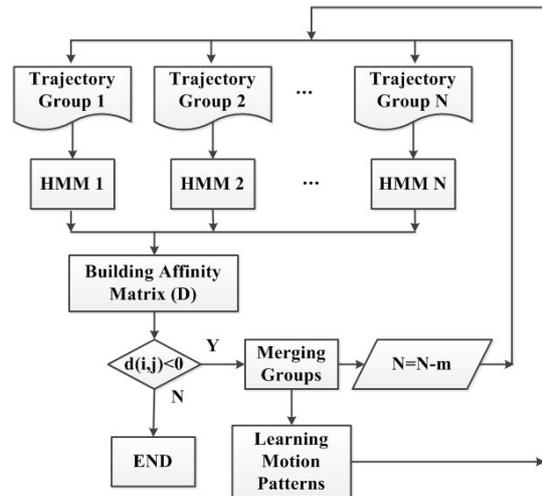


Fig. 4 The framework of Adaptable Dynamic Hierarchical Clustering Tree. The result of this algorithm is a tree-like hierarchic cluster in which the individual items are located in the leaf nodes and the tree branches represent clusters with two or more objects.

Table 1 Comparison of three methods in Dataset 1.

	Ordinary HC	DHC in Jiang's method	ADHC
Hierarchy Number	9624	4019	2184
Clustering Number	59	56	54

Step 1) Trajectories segmented in one group is fitted with a HMM. As a result, there are N groups with N HMMs, the j^{th} groups is represented as C_j ;

Step 2) For each trajectory group, we compute the dissimilarity measurement $d(i, j)$ and store all resultant distances in affinity matrix D .

Step 3) Find out the groups of dissimilarity smaller than d_{th} , merge them and constitute a newly-formed cluster. Supposing that there are m groups been merged in this hierarchy, when trajectory group i and j are merged, the newly-formed cluster is counted as C_i , and C_j will be deleted.

Step 4) The new HMM λ_{ij} is trained to replace λ_i and λ_j , the $N - m$ HMMs are retrained and formed a new affinity matrix \hat{D} .

Step 5) Chasing down a newly-formed cluster in the above $N - m$ feature arrays, the procedure is reduplicated until no $d(i, j) < d_{th}$.

A fast trajectory pattern learning algorithm is in need according to the current sharp increase of monitoring data. The method of ADHC in our work can solve this problem effectively. Table 1 shows the experimental comparison of our proposed ADHC with the ordinary HC [7] and DHC in Feng's work [18], which is tested on Dataset 1 described in the following Sect. 4. The result obviously shows that the hierarchy number of ADHC is much less than the others, so it is faster and much more timesaving.

3.5 Abnormality Detection

To detect the unusual events, we analyze the affinity matrix from group dissimilarity. Trajectory events are ordered with respect to their conformity scores. Then, every trajectory is compared with its own groups and computed the corresponding conformity scores. The trajectories that have low scores are identified as unusual events.

The conformity score of a trajectory j for a given feature f is the sum of the corresponding rows or columns of the affinity matrix D that belong that feature,

$$r_f(j) = \sum_{i=1}^M d_{ij} \quad (11)$$

The total conformity score for a trajectory is:

$$\varphi(j) = \arg \max_{j=1, \dots, M} \{r_f(j)\} \quad (12)$$

The trajectory with maximum score corresponds to the most abnormal event.

4. Experiments

We used the Edinburgh Informatics Forum Pedestrian Database (EIFPD) [30], which consists of a set of detected targets of people walking through the Informatics Forum. Moving objects are tracked in the image plane to obtain a series of trajectories and each point on a trajectory indicates the centroid of the tracked object. The camera is fixed overhead approximately 23m above the floor. The images are 640×480 , where each pixel (horizontally and vertically) corresponds to 24.7 mm on the ground. The capture rate is about 9 frames per second depending on the local Ethernet. Figure 5 shows a view of the scene.

In our work, two experimental datasets has been built from the EIFPD trajectory database, which contain different amounts of trajectories. Dataset 1 contains 10000 trajectories randomly selected from original dataset, and Dataset 2 contains 80000 trajectories. Table 2 shows details of these two datasets.

4.1 Motion Pattern Learning

Dataset 1 contains 10000 trajectories. Every single trajectory is segmented by uniformly sampling the Uniform Cubic B-spline and formed a codebook. Based on the trajectory preprocessing in subsection 3.1, a training database consisting of 9625 trajectories is selected from the overall trajectories, the lengths of which have exceeded 30 points.

Our learning method is multistep, and the first step is to train HMM for every group. The Baum-Welch algorithm is used for estimating final parameters of HMM, and the convergence threshold of our HMM is 1×10^{-4} , which can be computed as Formula (13), where Pr_i represents the maximum likelihood estimation in this HMM. To simplify the



Fig. 5 The view of background scene. The main entry/exit points contain the bottom left (front door), top left (cafe), top center (stairs), top right (elevator and night exit), bottom right (labs).

Table 2 Two experimental datasets.

	Trajectory amounts	After Preprocessing
Dataset1	10 000	9625
Dataset2	80 000	73276

computational complexity, the iteration of training HMM is set as 30 in our work showed in Fig. 7. In each loop the state transition probabilities will be more precise.

$$Th_{iteration} = \frac{Pr_{i+1} - Pr_i}{(Pr_{i+1} + Pr_i)/2} \quad (13)$$

According to method in subsection 3.4, 54 groups have been learned from Dataset 1, but some groups contain too few trajectories that cannot be defined as motion patterns. We set 1% as quantity threshold, which has 90 trajectories at least. The clustering result shows that there are 32 mainly Motion Patterns (MP) in our dataset. When trajectory number is added to 80000, the motion patterns will be more obvious. Figure 6 shows the compare of top 32 patterns in Dataset 1 and Dataset 2.

Because the clustering process is dynamic, the order of some patterns has changed. Comparing these two clusters, we find that most patterns are much the same, but there are some differences between them. MP 32 in Dataset 1 has been deleted from the top 32 patterns in Dataset 2, which changes to be MP34. The MP 30 in Dataset 2 is newly added, which is MP 33 in Dataset 1. The percent of these patterns will explain this change clearly. The data in Table 3 shows that percentage change is small. The numbers of trajectories are showed in Fig. 8. Most patterns in Dataset 2 are broader than that in Dataset 1, for example, the MP 1 of two clusters. We can explain this problem in a probability-based graph in Fig. 9. The size of the color depth indicates the probability of trajectories. The color of the graphs in (b) is deeper than that of (a), and middle position is deeper than sides. It means that there are more trajectories in (b), and most of them gather in a central location.

4.2 Abnormal Detection and Pattern Recognition

Abnormal behaviors might be person traversing an area which had not been traversed before, or one person moving in varying directions snaky. Based on Sect. 3.5, the behavior

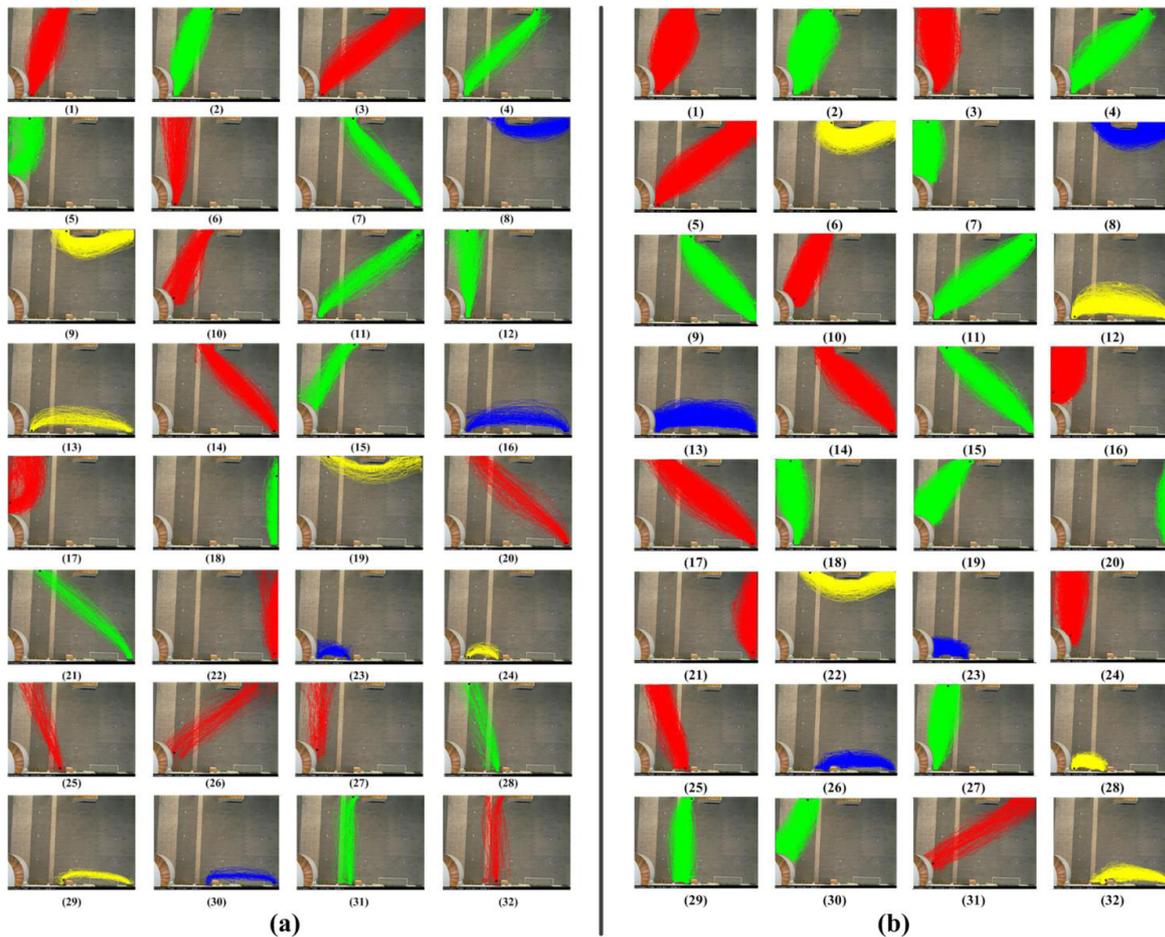


Fig. 6 Motion patterns compare of two Datasets. (a) The 32 mainly trajectory motion patterns from Dataset 1. (b) The top 32 mainly trajectory motion patterns in Dataset 2. The blank point represents the entry position of this pattern. Different color represents different directions of trajectory groups: red (upward), green (downward), blue (leftward) and yellow (rightward).

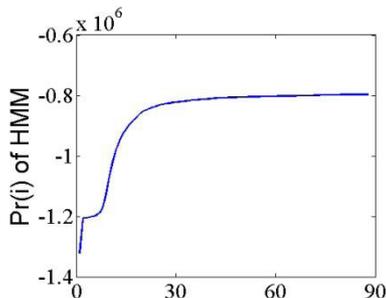


Fig. 7 The likelihood estimation curve. This curve is for iteration of overall trajectory group HMM. Experiments results show that when iteration reaches 30, the rate of convergence will be slower.

Table 3 Percent of different MPs in two datasets.

	Dataset 1	Dataset 2
MP	MP 32	MP 34
Percent (%)	0.0081	0.0070
MP	MP 33	MP 30
Percent (%)	0.0079	0.0085

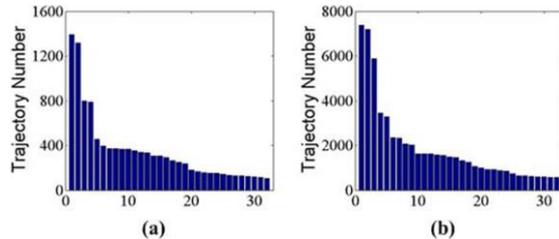


Fig. 8 Numbers of motion patterns. (a) Numbers of 32 mainly motion patterns for Dataset 1. (b) Numbers of top 32 motion patterns for Dataset 2.

pattern in the testing sets is labeled as abnormal if there are no similar patterns in the corresponding training sets. The abnormality detection measure can be showed in Fig. 10.

We have learned 32 motion patterns from Dataset 1, which contain a majority of models in this scene. To recognize the patterns of more trajectories, we can compare the similarity between new trajectory and existing patterns. Another 10000 trajectories have been picked out and pre-processed as Sect. 3.1. Every trajectory will be compared with the learned 32 MPs in Dataset 1. It can be merged into one

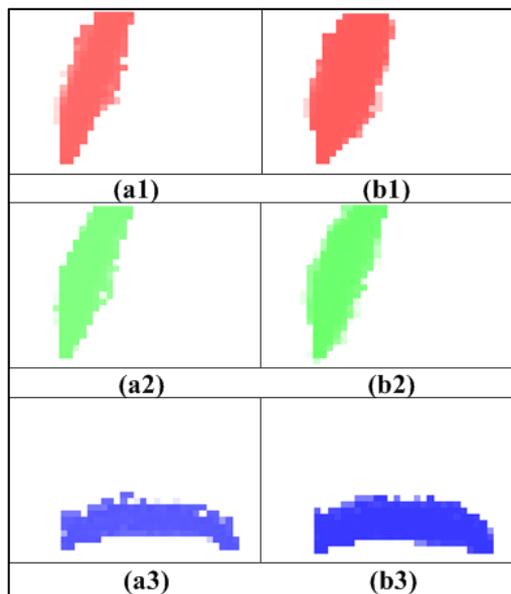


Fig. 9 The probability-based graph in two datasets. (a1), (a2), (a3) represent the probability-based graphs of Dataset 1, and (b1), (b2), (b3) represent that of Dataset 2.

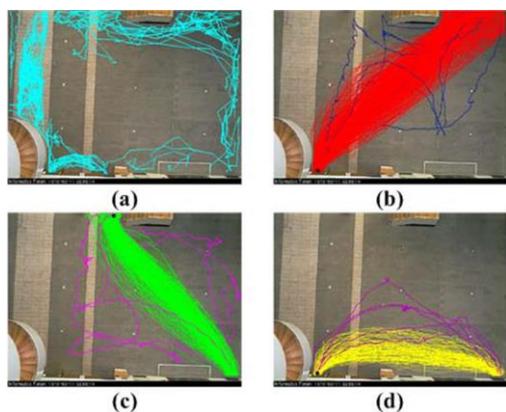


Fig. 10 Abnormal behaviors in Dataset 1. (a) The 375 short trajectories. (b), (c), (d) Some behavior abnormal trajectories. The main color trajectories represent the trajectory motion patterns, and other color trajectories represent abnormal ones.

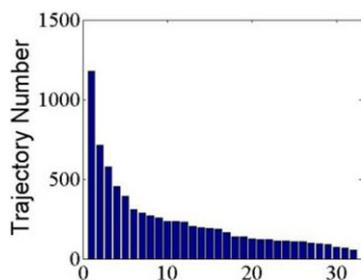


Fig. 11 Trajectory number of pattern recognition patterns.

of the pattern which has the most similarity. A threshold has been set to detect abnormal trajectory. If all the similarities are beyond the threshold, the trajectory is set to be abnormal.

Based on these results in Fig. 11, it shows that the

HMM-based method approaches a successfully representation to trajectory clustering. Our experiments also prove the proposed method is stable.

5. Conclusions

This paper proposes an unsupervised hierarchical method for learning motion patterns and detecting anomaly. We describe the feature of trajectory groups as Hidden Markov Models (HMMs), and each trajectory group can be associated with one of the learned HMMs. The Adaptable Dynamic Hierarchical Clustering (ADHC) has obvious better efficiency for trajectory clustering. In our experiment, 32 kinds of motion patterns are learned which can almost cover all the movement behaviors in the scene. The abnormality detection and behavior analysis in our work achieve a good performance.

The above work of models learning may be applied in many domains of the visual surveillance, such as virtual reality, path design, long-term predictions and atypical motion detection. Our method of clustering deserves further study and may be applicable to other areas of computer vision and pattern analysis.

Acknowledgments

This research is partially supported by National Natural Science Foundation of China (No. 61370127, No.61100143), Program for New Century Excellent Talents in University (NCET-13-0659), Beijing Higher Education Young Elite Teacher Project (YETP0583). The opinions expressed are solely those of the authors and not the sponsors.

References

- [1] J. Vlahos, "Surveillance society: New high-tech cameras are watching you," *Popular Mechanics*, pp.64–69, 2008.
- [2] H. Buxton, "Learning and understanding dynamic scene activity: A review," *Image and Vision Computing*, vol.21, no.1, pp.125–136, 2003.
- [3] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Trans. Syst. Man Cybern., B, Cybern.*, vol.35, no.3, pp.397–408, 2005.
- [4] X. Wang, X. Ma, and W.E.L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.31, no.3, pp.539–555, 2009.
- [5] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.4851–4856, 2006.
- [6] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.28, no.9, pp.1450–1464, 2006.
- [7] B.T. Morris and M.M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol.18, no.8, pp.1114–1127, 2008.
- [8] I.N. Junejo, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," *ICPR 2004. Proc. 17th International Conference on Pattern Recognit.*, 2004, vol.2, pp.716–719, 2004.
- [9] N. Sumpter and A. Bulpitt, "Learning spatio-temporal patterns for

- predicting object behaviour,” *Image and Vision Computing*, vol.18, no.9, pp.697–704, 2000.
- [10] W. Hu, D. Xie, T. Tan, and S. Maybank, “Learning activity patterns using fuzzy self-organizing neural network,” *IEEE Trans. Syst. Man Cybern., B, Cybern.*, vol.34, no.3, pp.1618–1626, 2004.
- [11] W. Hu, D. Xie, and T. Tan, “A hierarchical self-organizing approach for learning the patterns of motion trajectories,” *IEEE Trans. Neural Netw.*, vol.15, no.1, pp.135–144, 2004.
- [12] C. Stauffer and W.E.L. Grimson, “Learning patterns of activity using real-time tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, no.8, pp.747–757, 2000.
- [13] X. Wang, K.T. Ma, G.W. Ng, and W.E.L. Grimson, “Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models,” *International Journal of Computer Vision*, vol.95, no.3, pp.287–312, 2011.
- [14] D. Biliotti, G. Antonini, and J.P. Thiran, “Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences,” *Seventh IEEE Workshops on Application of Computer Vision, 2005. WACV/MOTIONS’05*, vol.2, pp.50–57, 2005.
- [15] M. Farenzena, A. Fusiello, and R. Gherardi, “Structure-and-motion pipeline on a hierarchical cluster tree,” *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp.1489–1496, 2009.
- [16] T. Wang, “CA-Tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles,” *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol.41, no.3, pp.686–698, 2011.
- [17] C. Juarez, A.R. Messina, R. Castellanos, and G.E. Perez, “Characterization of multimachine system behavior using a hierarchical trajectory cluster analysis,” *IEEE Trans. Power Syst.*, vol.26, no.3, pp.972–981, 2011.
- [18] F. Jiang, Y. Wu, and A.K. Katsaggelos, “A dynamic hierarchical clustering method for trajectory-based unusual video event detection,” *IEEE Trans. Image Process.*, vol.18, no.4, pp.907–913, 2009.
- [19] M. Bennewitz, G. Cielniak, and W. Burgard, “Utilizing learned motion patterns to robustly track persons,” *Proc. Joint IEEE Int. Workshop on VS-PETS*, pp.102–109, 2003.
- [20] L. Kratz and K. Nishino, “Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models,” *IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009*, pp.1446–1453, 2009.
- [21] F.I. Bashir, A.A. Khokhar, and D. Schonfeld, “Object trajectory-based activity classification and recognition using hidden Markov models,” *IEEE Trans. Image Process.*, vol.16, no.7, pp.1912–1919, 2007.
- [22] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, “Learning motion patterns and anomaly detection by human trajectory analysis,” *IEEE International Conference on Systems, Man and Cybernetics*, pp.498–503, 2007.
- [23] F. Porikli and T. Haga, “Event detection by eigenvector decomposition using object and frame features,” *Computer Vision and Pattern Recognition Workshop, 2004, CVPRW’04*, p.114, 2004.
- [24] T. Xiang and S. Gong, “Video behaviour profiling and abnormality detection without manual labelling,” *Tenth IEEE International Conference on Computer Vision, 2005, ICCV 2005*, vol.2, pp.1238–1245, 2005.
- [25] H. Zhou and D. Kimber, “Unusual event detection via multi-camera video mining,” *18th International Conference on Pattern Recognition, 2006, ICPR 2006*, vol.3, pp.1161–1166, 2006.
- [26] N.M. Oliver, B. Rosario, and A.P. Pentland, “A Bayesian computer vision system for modeling human interactions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, no.8, pp.831–843, 2000.
- [27] F. Jiang, “Anomalous event detection from surveillance video,” *Doctoral Dissertation, Electrical and Computer Engineering, Northwestern University*, 2011.
- [28] Y.-C. Chen, “Solving robot trajectory planning problems with uniform cubic B-splines,” *Optimal Control Applications and Methods*,

vol.12, no.4, pp.247–262, 1991.

- [29] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol.77, no.2, pp.257–286, 1989.
- [30] B. Majecka, *Statistical models of pedestrian behaviour in the Forum*, Master’s thesis, School of Informatics, University of Edinburgh, 2009.



Kai Kang received the B.S degrees in Communication Engineering from Zhengzhou University in 2011. He is now studying for a M.S. degree in Human-Computer Interaction Engineering at Beijing Jiaotong University. His current interests include image processing, virtual reality and computer vision.



Weibin Liu received the Ph.D. degree in Signal and Information Processing from Institute of Information Science at Beijing Jiaotong University, China, in 2001. During 2001–2005, he was a researcher in Information Technology Division at Fujitsu Research and Development Center Co., LTD. Since 2005, he has been with the Institute of Information Science at Beijing Jiaotong University, where currently he is a professor in Digital Media Research Group. He was also a visiting researcher in Center for Human

Modeling and Simulation at University of Pennsylvania, PA, USA during 2009–2010. His research interests include computer graphics, computer animation, virtual human and virtual environment, computer vision, and pattern recognition. He is a member of the IEEE, ACM, IEICE.



Weiwei Xing received the Ph.D. degree in Signal and Information Processing from Institute of Information Science at Beijing Jiaotong University, China, in 2006. Since then, she has been with the School of Software Engineering, Beijing Jiaotong University, where currently she is an associate professor in Digital Media. Her research interests are computer graphics, digital media processing, and pattern recognition.