SOM Based Activity Learning for Visual Surveillance System

Lin Qu
Institute of Advanced Digital Technology and Instrument
Zhejiang University
Hangzhou, China

Fan Zhou
Institute of Advanced Digital Technology and Instrument
Zhejiang University
Hangzhou, China

Yaowu Chen
Institute of Advanced Digital Technology and Instrument
Zhejiang University
Hangzhou, China

***Abstract*—This paper proposes a new object activity learning algorithm based on self-organizing map (SOM) to detect anomaly events and predict activities in intelligent visual surveillance system. Two SOM networks are used to construct the distribution patterns of sub-trajectories and trajectories respectively. Sub-trajectories are first sampled to reveal the local activities. Before constructing the distribution patterns, trajectories are represented based on the distribution patterns of sub-trajectories. Finally, the distribution patterns of trajectories are merged to form clusters using agglomerative hierarchical clustering algorithm. By using the patterns of sub-trajectories, the learning process is accelerated and the representation of trajectory is simplified. The patterns of sub-trajectories and trajectories learned are then used to detect local and global anomaly events. A fuzzy set theory based predicting method is also proposed to predict the activity of object. Experimental results on real scene demonstrate the effectiveness of the proposed algorithm.**

***Keywords*—self-organizing map; visual surveillance; trajectory classify; anomaly detection; activity prediction**

I. INTRODUCTION

Visual surveillance is an active research field of computer vision. The problems in visual surveillance system[1-2] include motion detection, tracking, objects classification, activity understanding and semantic description. Among these problems, activity understanding and semantic description are important to intelligent surveillance system because it can produce high level descriptions of object activities. This paper focuses on the problem of activity understanding in intelligent visual surveillance system.

Many schemes have been proposed to understand the motion activities using Hidden Markov Model (HMM) [3], spectral clustering[4] and fuzzy clustering[5]. Other schemes are proposed to learn activity patterns in the coefficient feature space [6]. Self-Organizing Map (SOM) is also used to automatically generate activity patterns of moving objects. Johnson *et al.* [7] use flow vectors to describe trajectory, and generate the statistical model of object trajectories using two competitive learning networks which are connected with leaky neurons. Sumpter *et al.* [8] apply a feedback mechanism to [7] and present a novel approach to learn spatiotemporal patterns of objects. They also provide a neural network paradigm to predict object behavior. Jonathan *et al.* [9] propose a scheme to learn the distribution patterns of flow vectors using SOM and detect novel events using the patterns learned. Hu *et al.* [10-11] present new self-organizing neural network models to learn the trajectory distribution patterns, and use the patterns learned to detect anomaly events and predict object behaviors.

This paper proposes an object activity learning algorithm based on self-organizing map[12] (SOM) to detect anomaly events and predict activities in intelligent visual surveillance system. The workflow of the algorithm is shown in Fig. 1. Sub-trajectories are first constructed to reveal the local activities. Two SOM networks are then used to construct the distribution patterns of sub-trajectories and trajectories respectively. Finally, the distribution patterns of trajectories are merged to form categories using agglomerative hierarchical clustering algorithm. The patterns of sub-trajectories and trajectories learned are then used to detect anomaly events and predict object behaviors. By constructing sub-trajectories, the sequence patterns of activities can be learned by the first SOM network. No leaky neuron is used between two SOM networks. So the distortion problem[7] caused by leaky neurons can be avoided. Since the number of sub-trajectory patterns is less than the number of flow vector patterns, the learning process can be accelerated. Before constructing the distribution patterns, trajectories are represented by equal length based on the distribution patterns of sub-trajectories. This has two advantages. Firstly, the representation of trajectory can be directly used as the input of SOM network without cutting and padding. Secondly, the dissimilarity between trajectories can be measured by their Euclidean distance. When predicting the activity of object, a fuzzy set theory based method is proposed which is used to improve the predict accuracy. Experimental results on real scene demonstrate the effectiveness of the proposed algorithm.

II. TRAJECTORY PREPROCESSING

The trajectories are acquired by tracking moving objects. Image sequences are sampled at fixed time intervals (frames) to acquire object centroids by object tracker. Centroids in different samples are connected to form the trajectory. Because there are noises in the raw trajectory data, a moving average filter is used to smooth the trajectories. So a trajectory T of the moving object can be defined as:

$$T=\{ (x_1,y_1), (x_2,y_2), \dots , (x_i,y_i), \dots(x_l,y_l) \}$$

where (x_i,y_i) is the two dimensional coordinates of the moving object in ith sampling, l is the number of samples.

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

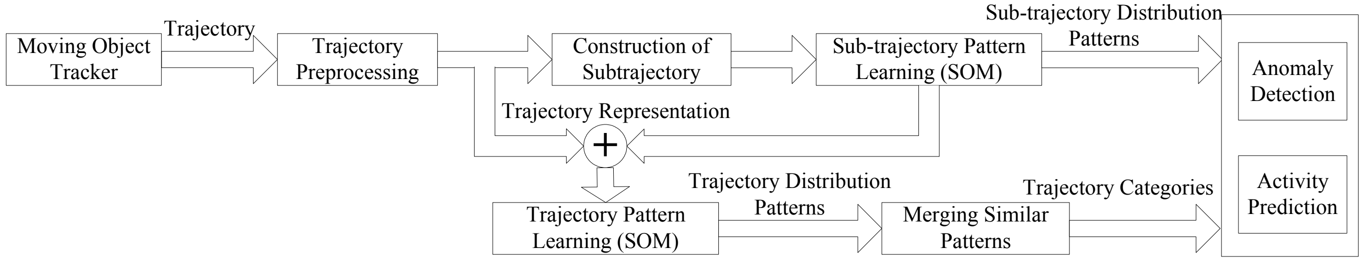


Figure 1. Overview of the algorithm

Besides position, velocity is also very important for measuring the similarity between trajectories. The velocity can be described by the difference of the centroids in adjacent samples: $\delta x_i = x_{i+1} - x_i$, $\delta y_i = y_{i+1} - y_i$.

So at the ith sampling, the position and the velocity of the moving object can be described by a flow vector[7] f:

$$f_i = (x_i, y_i, \delta x_i, \delta y_i)$$

The ranges of the velocity and the position components in flow vector are different. In order to balance the relative contribution of velocity and position, each component in the flow vector is normalized to the range [0,1]. After the preprocessing, trajectory can be described by the sequence of flow vectors:

$$F = \{ f_1, f_2, \ldots, f_i, \ldots, f_l \} \tag{1}$$

III. Learning Algorithm

This section proposes the object activity learning algorithm. The procedure of the algorithm is shown in Fig. 2.

A. Construction of Sub-trajectory

Sub-trajectory is a continuous subsequence of a trajectory, which depicts the local activities of the object. In order to be used as the input of the learning algorithm, sub-trajectories should be in equal length. Suppose a trajectory T whose length is l, its sub-trajectories can be constructed by copying subsequences with length ξ and interval λ. So the sub-trajectory set S of trajectory T is:

$$S = \{ s_1, s_2, \cdots, s_i, \cdots, s_n \} \quad s_i = \{ f_{i\lambda-\lambda+1}, f_{i\lambda-\lambda+2}, \cdots, f_{i\lambda-\lambda+\xi} \}$$

where $n = \lceil (l - \xi + 1)/\lambda \rceil$.

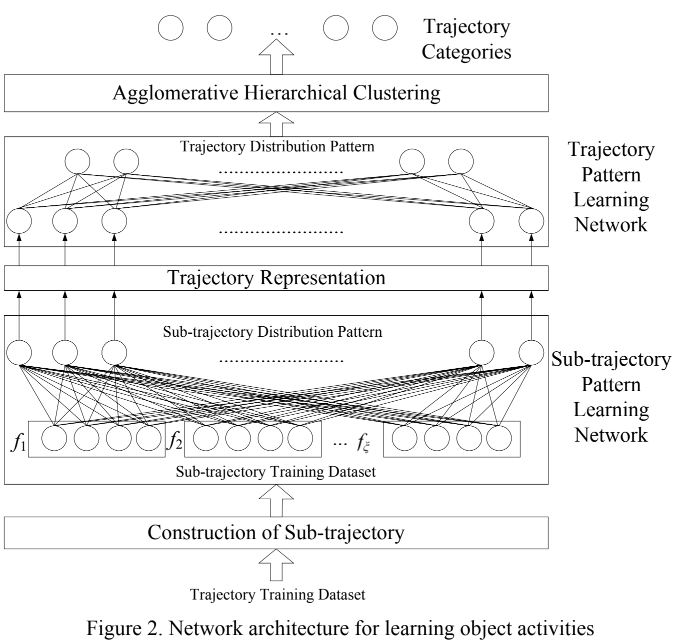


Figure 2. Network architecture for learning object activities

From the definition above, it can be seen that sub-trajectory includes both spatial and temporal information of the local activities. When $\lambda=\xi=l$, S has one sequence which is just F. When $\lambda=\xi=1$, S has l sequences of which each contains one flow vector in F. Larger λ and ξ imply more temporal (sequence) information, while smaller λ and ξ imply more common patterns. To balance the two factors, ξ and λ can be defined as $\xi = \min(\sqrt{L}, l_{\min}) \quad \lambda = \xi/4$, where L is the average length of the trajectory data set and $l_{\min}$ is the minimum length of the trajectory data set.

B. Sub-trajectory Distribution Pattern Learning

Using the sub-trajectory construction algorithm, each trajectory in the training data set can produce a sub-trajectory set. The sub-trajectory training data set can be constructed by merging all the sub-trajectory sets produced. This paper uses Self-Organizing Map[12] (SOM) to learn the spatiotemporal distribution pattern of the sub-trajectory. SOM is a two layers neural network. The network proposed is abbreviated as SPLN (Sub-trajectory Pattern Learning Network) in the following sections. The input layer of the SPLN has $\xi \times 4$ neurons of which each corresponds to a component of the flow vectors in the sub-trajectory sequence. The output layer of the SPLN has M neurons which correspond to the patterns of sub-trajectories. Larger M can depict sub-trajectory patterns more precisely, but may increase the computational complexity. M can take any values, but must be smaller than the number of training sub-trajectories and larger than the number of output neurons of TPLN (refer to III.C).

The learning process of SPLN consists of the following steps:

a) Randomly choose M sub-trajectories from training data set as the initial value of the weight vectors $\{w_j\}$

b) Choose a sub-trajectory s from training data set. Calculate the Euclidean distances from s to the weight vectors. The "winning" neuron is the one whose Euclidean distance is the smallest:

$$i(s) = \arg\min_j \| s - w_j \| \tag{2}$$

c) The winning neuron and its neighbor neurons update their weight vectors according to (3)

$$w_j\left(t+1\right)=w_j\left(t\right)+\eta\left(t\right)h_{j,i(s)}\left(t\right)\left(s-w_j\left(t\right)\right) \tag{3}$$

$h_{j,i}(t)$ is the neighborhood function which is defined as $h_{j,i}(t)=\exp(-d_{j,i}^2/2\sigma^2(t))$. $h_{j,i}(t)$ takes the maximal value 1 when $j=i(s)$ and falls off with the distance $d_{j,i}$ between j and $i(s)$, $\sigma(t)$ is an exponentially decreasing width parameter and $\eta(t)$ is an exponentially decreasing learning rate.

d) Repeat b-c for n_a training cycles. After n_b training cycles ($n_b < n_a$), the width parameter and learning rate are reinitialized to make the learning result converge.

After training, SPLN can be used to classify the input sub-trajectory. The output of the ith output neuron in SPLN can be then defined as:

$$O_i(s)=\|s-w_i\|$$

where w_i is the weight vector of the ith output neuron.

Given an input sub-trajectory s, the output of SPLN will be:

$$O(s)=\{O_1(s),O_2(s),\ldots,O_i(s),\ldots,O_M(s)\}$$

$O_i(s)$ depicts the dissimilarity between s and the ith sub-trajectory patterns. The neuron whose output is smallest ($\arg\min_i O_i\left(s\right)$), represents the best matching sub-trajectory pattern for s.

C. Trajectory Representation and Learning

The trajectory descriptor as defined in (1) has some drawbacks. First, the trajectories are not in equal length and second, it is difficult to measure the similarity between trajectories. This paper proposes a new representation for trajectories to solve these problems.

Given a trajectory T, input all its sub-trajectories to SPLN and the output set is $O(S)=\{O(s_1),O(s_2),\ldots,O(s_n)\}$.

According to III.B, the temporal nature (sequence pattern) of trajectory has been included in the patterns of sub-trajectory. The representation R of trajectory T is defined as the minimum distance from all its sub-trajectories to each sub-trajectory pattern:

$$R=\{r_1,\ldots,r_k,\ldots,r_M\}, \text{ where } r_k=\min(O_k(s_1),O_k(s_2),\ldots,O_k(s_n)).$$

The distance between trajectories can be defined as the Euclidean distance of their representation:

$$d(R_i,R_j)=\|R_i-R_j\| \tag{4}$$

From (4), it can be seen that the representations of similar trajectories are also similar because their distances to sub-trajectory patterns are close to each other, and vice versa.

Using the representation of trajectory, trajectory patterns can be learned using a second SOM network. It is abbreviated as TPLN (Trajectory Pattern Learning Network) in the following sections. The input layer of the TPLN has M neurons of which each corresponds to a sub-trajectory pattern. The output layer of the TPLN has N neurons which correspond to the patterns of trajectory. Since the number of trajectory categories is unknown, the value of N can be chosen arbitrarily. Larger N can depict trajectory patterns more precisely. However, N must be smaller than the number of training trajectories. The learning process and parameters are the same as SPLN, except using (4) as the distance measurement function.

D. Merging Similar Patterns

After training, each neuron in TPLN corresponds to a trajectory pattern which is described by the weight vector W_i. Since the patterns of trajectory do not correspond to trajectory categories, there may be similar patterns in the result learned. In this section, an agglomerative hierarchical clustering algorithm is used to merge the similar patterns and infer the best classification of trajectories. During the hierarchical clustering process, the maximal dissimilarity in cluster[13] is used to judge the termination condition.

The algorithm consists of the following steps:

a) Initialize: treat each trajectory pattern as a cluster and produce the initial clustering $C_0=\{W_i, i=1\ldots N\}$. Then initialize the proximity matrix using (4)

b) $k=0$

c) repeat

d) $k=k+1$

e) Find the least dissimilar pair of clusters in the proximity matrix, say pair W_i and W_j

f) Merge clusters W_i and W_j into a single cluster $W_q=W_i\cup W_j$ and produce the new clustering $C_k=(C_{k-1}-\{W_i,W_j\})\cup\{W_q\}$

g) Update the proximity matrix by deleting the rows and columns corresponding to W_i and W_j. The proximity between the new cluster W_q and old cluster W_k is calculated by the weighted pair group method average (*WPGMA*) algorithm:

$$d\left(W_q,W_k\right)=\frac{1}{2}\left(d\left(W_i,W_k\right)+d\left(W_j,W_k\right)\right)$$

h) For each cluster I in C_k

Calculate the average distance $\mu(I)$ and the variance $\sigma(I)$ of the trajectory patterns in cluster I.

Calculate the maximum distance $h(I)$ between the trajectory patterns in cluster I:

$$h(I)=\max\{d(W_s,W_t),\ W_s,W_t\in I\}$$

i) until there is only one cluster in C_k or the following termination condition is satisfied:

$$\exists\ I\in C_k:\ h(I)>\mu(I)+\beta\sigma(I) \tag{5}$$

j) return C_{k-1} as the trajectory classification result

(5) is the termination condition of the clustering process. An intuitive explanation is that when merging two similar clusters, the distance between the trajectory patterns is not far from the average μ, and vice versa. In (5), β is a predefined parameter. The value of β determines the maximum dissimilarity between trajectory patterns in each category. According to the central limit theorem, the distribution of the distance between trajectory patterns in a category can be considered as a normal distribution. So the distances in each category fall in the range of one σ with a probability of 84.13% and fall in the range of three σ with a probability of 99.87%. Therefore, it is appropriate to choose β from the range of [1,3].

Since the trajectory patterns have been classified into clusters, the weight vector of each trajectory category can be calculated by averaging the weight vectors of the trajectory patterns in the category. Assuming the number of trajectory categories in C_{k-1} calculated by the above algorithm is X, the weight vector of the trajectory category I can be defined as:

$$K_I = \sum_{W_j \in I} W_j \Big/ \sum_{W_j \in I} 1$$

The set of the trajectory categories can be defined as:

$$C=\{K_1,K_2,\ldots,K_I,\ldots,K_X\} \tag{6}$$

Finally, any trajectory T_x can be classified using the following algorithm:

a) Apply moving average filter to T_x and construct its sub-trajectories

b) Calculate the representation of T_x using the output of SPLN, which is denoted by R_x

c) Find the smallest distance between R_x and the weight vectors in (6), the corresponding category is the best matching category of T_x

IV. ANOMALY DETECTION AND ACTIVITY PREDICTION

A. Anomaly Detection

Based on the patterns learned, this paper proposes two anomaly detection algorithms. One is sub-trajectory anomaly detection algorithm which detects local abnormal activities when object is moving. Another is trajectory anomaly detection algorithm which detects global abnormal activities of the object when the tracking is completed.

Given an incomplete trajectory T_x, its sub-trajectory set S_x can be constructed after smoothing. Considering any sub-trajectory s_i in S_x, the distance between s_i and its best matching pattern w_c can be calculated. If the distance is larger than a threshold ε_c, s_i is considered as abnormal. Suppose J_c denotes the training sub-trajectories whose best matching patterns are w_c, the threshold ε_c can be defined as the half of the maximal Euclidean distance between w_c and J_c:

$$\varepsilon_c = \frac{1}{2}\max_{s \in J_c} d\left(s, w_c\right).$$

When the trajectory T_x is complete, the best matching category of T_x can be found using its representation R_x. If the distance between T_x and its best matching category K_I is larger than a threshold ε_e, T_x is considered as abnormal. Similar to ε_c, suppose J_e denotes the training trajectories whose best matching categories are K_I, the threshold ε_e can be defined as the half of the maximal Euclidean distance between K_I and the representation of J_e:

$$\varepsilon_e = \frac{1}{2}\max_{R \in J_e} d\left(R, K_I\right).$$

B. Activity Prediction

The trajectory categories learned represent the possible activities in the scene. Given an incomplete trajectory, the following activity of the object can be predicted by the matched trajectory categories.

When the trajectory is complete, the trajectory classification algorithm in III.D can judge its category accurately according to its complete sub-trajectory distribution patterns. But when the trajectory is incomplete, only portion of the final sub-trajectory patterns can be determined. The classification algorithm in III.D may make the wrong decision because of the undetermined patterns. So we must find the determined sub-trajectory patterns first.

Given part of a trajectory T_p, its most possibly determined sub-trajectory patterns (MPDS) can be found using the following steps:

a) Construct its sub-trajectory set S_p

b) For each sub-trajectory s_j in S_p , find its best matching sub-trajectory pattern $i(s_j)$ using (2)

c) The set of the patterns found is defined as the MPDS of T_p:

$$Q_p=\{\, i(s_1),\, i(s_2)\ldots i(s_x)\}$$

Using MPDS, the partial distance between T_p and the trajectory category K_I can be defined as:

$$d\left(R_p, K_I\right) = \sqrt{\sum_{j \in Q_p}\left(R_p\left(j\right) - K_I\left(j\right)\right)^2}$$

where R_p is the representation of T_p

In order to compare the similarity between T_p and the trajectory categories, the membership of T_p to trajectory category K_I is defined as:

$$u_I\left(R_p\right) = \frac{\left(1\big/d\left(R_p, K_I\right)\right)^2}{\sum_{j=1}^{X}\left(1\big/d\left(R_p, K_j\right)\right)^2} \tag{7}$$

From the definition of u_I, it can be seen that $0 \le u_I \le 1$. Larger value of membership indicates higher similarity between T_p and the corresponding trajectory category. So the membership can

Figure 3. Background scene with smoothed object trajectories

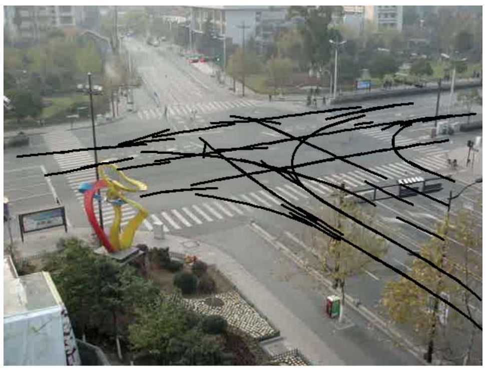


Figure 4. Sub-trajectory pattern learning results

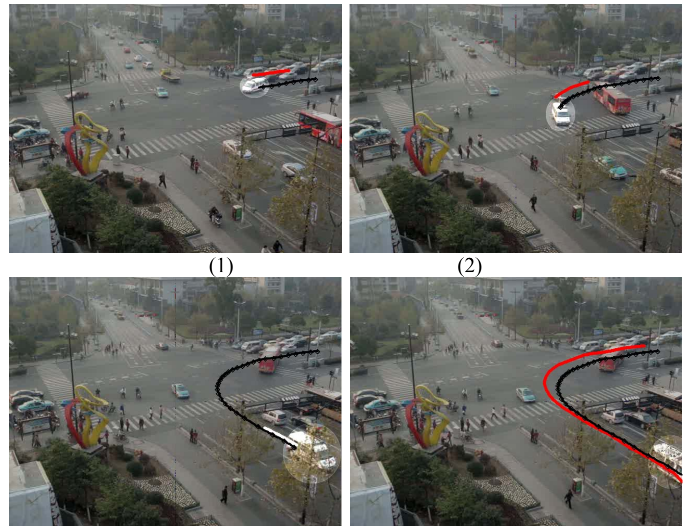
Figure 5. Trajectory category learning results

be used to predict the future activity of the moving object. The trajectory category with the largest membership is chosen as the most possible future activity of the object.

V. Experimental results

The algorithm proposed in this paper is implemented using Visual C++ 2005 on the Windows XP platform. The training set contains 220 trajectories which are generated by object tracker. The smoothed training set is shown in Fig. 3.

The number of input neurons in SPLN is 32 and the number of output neurons is 144. To train the SPLN, the training cycles n_a is set to 2000. The initialized learning rate $\eta(0)$ is 0.1 and $\sigma(0)$ is set to cover all the output neurons. After 1000 cycles, η is reinitialized to decrease from 0.01 and σ is reinitialized to the value which can make the final width decrease to one neighbor of the winning neuron. The sub-trajectory patterns learned are shown in Fig. 4. It can be seen that the sub-trajectory patterns depict local activities closely.



(1) (2) (3) (4)

Figure 6. Anomaly detection experiment

TABLE I. Data Used to Detect Abnormal Sub-trajectories in Fig. 6

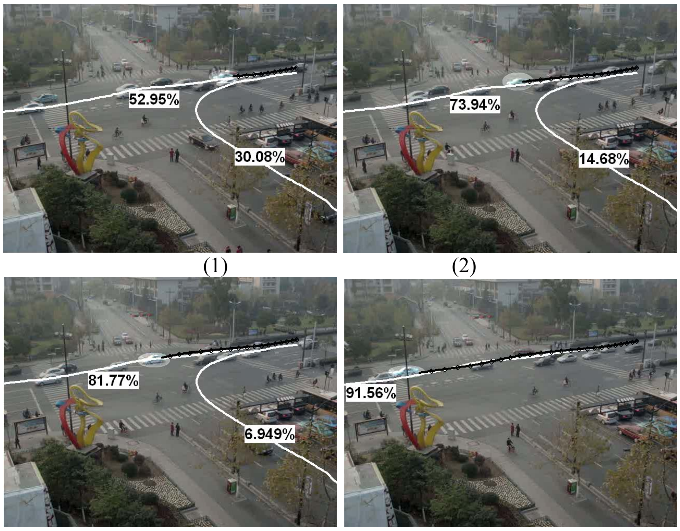
No. of Figure	Distance Between Sub-trajectory and its Best Matching Pattern	Threshold ε_c	Detection Result
(1)	0.1774	0.08296	abnormal
(2)	0.1201	0.05389	abnormal
(3)	0.09074	0.09430	normal

TABLE II. Data Used to Detect Abnormal Trajectory in Fig. 6

No. of Figure	Distance Between Trajectory and its Best Matching Pattern	Threshold ε_e	Detection Result
(4)	0.8207	0.3963	abnormal

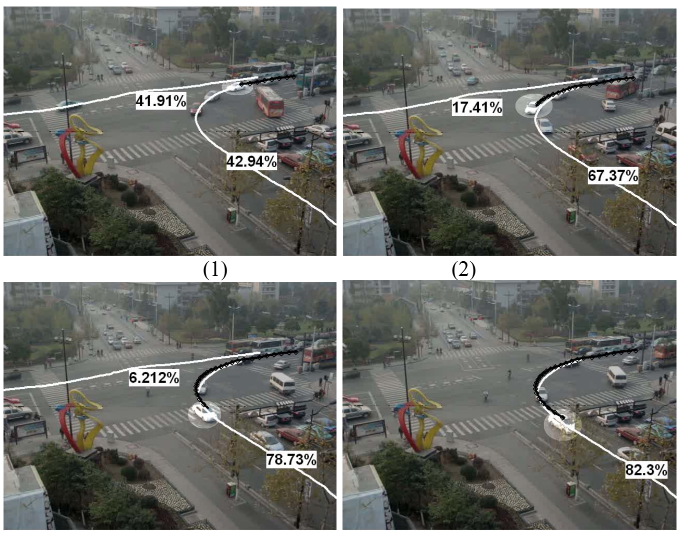
Fig. 5 illustrates the trajectory categories generated by the trajectory pattern learning and agglomerative hierarchical clustering algorithm. The number of categories generated is 11. Each category is represented by the trajectory whose distance to it is the smallest in the training data set. During the learning process, the number of output neurons in TPLN is 64 and the threshold β is set to 3. It can be seen that the proposed algorithm depicts the trajectory distribution completely and accurately by using few categories.

Fig. 6 is an example of anomaly detection. In the figures, both sub-trajectory and its best matching sub-trajectory patterns are shown. The black line with symbol "♦" represents the trajectory of object. When an anomaly activity is detected, its best matching pattern is marked with bold red line. Otherwise, the pattern is marked with bold white line. Table I shows the data used to detect abnormal sub-trajectories in Fig. 6, while Table II shows the data used to detect abnormal trajectory in Fig. 6. Figure (1)-(3) show the sub-trajectory anomaly detection results, while figure (4) shows the trajectory anomaly detection result. In (1), an ambulance is traveling in the wrong lane. It can be seen in Table I that the distance between the sub-trajectory and its best matching pattern is larger than the threshold ε_c. The sub-trajectory is correctly marked as abnormal. Similarly, the subsequent ambulance sub-trajectory in (2) is also marked as abnormal. In (3), the ambulance travels in the correct lane and the distance in Table I is smaller than its corresponding threshold ε_c. So the sub-trajectory is correctly marked



(1) (2) (3) (4)

Figure 7. Activity prediction experiment



(1) (2) (3) (4)

Figure 8. Another activity prediction experiment

as normal. In (4), the complete trajectory of the ambulance is marked as abnormal because the distance to its best matching category is larger than the threshold ε_e in Table II.

Fig. 7 is an example of activity prediction. In the figures, the black line with symbol "♦" represents the trajectory of the vehicle and the white line represents the trajectory predicted. The percentage besides the predicted trajectory represents the membership of the vehicle trajectory to the predicted trajectory. In (1), the car enters the scene and two most possible trajectories are shown. In (2)(3), as the car goes straight, the membership to the forward trajectory is increasing while the membership to the left-tune trajectory is decreasing. In (3), the membership to the left-tune trajectory is less than 10%. In (4), there is only one predicting trajectory left and the membership to it is more than 90%. Another similar example of activity prediction is demonstrated in Fig. 8.

From the experimental results above, it can be seen that the algorithm proposed in this paper can correctly detect the global and local abnormal activities of moving objects. It also predicts the activities of the objects accurately and the prediction is consistent with the human judgment.

VI. CONCLUSION

This paper proposes an object activity learning algorithm based on self-organizing map (SOM) to detect anomaly events and predict activities in intelligent visual surveillance system. The algorithm uses two SOM networks to learn the patterns of local and global activities in the scene respectively. Sub-trajectories are constructed to reveal the local activities and simplify the representation of trajectory. The local and global activity patterns learned are used to detect local and global anomaly events. A fuzzy set theory based predicting method is also proposed to predict the activity of object. Experimental results on real scene demonstrate the effectiveness of the proposed algorithm.

ACKNOWLEDGMENT

This work is supported by the National High-Tech Research and Development Plan of China (2003AA1Z2130) and the Science and Technology Project of Zhejiang Province of China (2005C11001-02).

REFERENCES

[1] W. Hu, T. Tan, L. Wang and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.34, 2004.

[2] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," in *Proc. Vision, Image and Signal Processing*, *2005*.

[3] J. Alon, S. Sclaroff, G. Kollios and V. Pavlovic, "Discovering clusters in motion time-series data," in *Proc. the Conference on Computer Vision and Pattern Recognition*, *2003*.

[4] S. Atev, O. Masoud and N. Papanikolopoulos, "Learning Traffic Patterns at Intersections by Spectral Clustering of Motion Trajectories," in *Proc. the International Conference on Intelligent Robots and Systems*, *2006*.

[5] N. Anjum and A. Cavallaro, "Unsupervised fuzzy clustering for trajectory analysis," in *Proc. the International Conference on Image Processing*, *2007*.

[6] A. Nafte and S. Khalid, "Motion Trajectory Learning in the DFT-Coefficient Feature Space," in *Proc. the Fourth IEEE International Conference on Computer Vision Systems*, *2006*, pp. 47.

[7] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image Vision Computing*, vol.14, pp. 609-615, 1996.

[8] N. Sumpter and A. J. Bulpitt, "Learning spatio-temporal patterns for predicting object behaviour," *Image Vision Computing*, vol.18, pp. 697-704, 2000.

[9] J. Owen and A. Hunter, "Application of the Self-Organizing Map to Trajectory Classification," in *Proc. the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, *2000*, pp. 77.

[10] W. Hu, D. Xie, T. Tan and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol.34, 2004.

[11] W. Hu, D. Xie, T. Tan and J. Shen, "A Hierarchical Self-Organizing Approach for Learning the Patterns of Motion Trajectories," *Chinese Journal of Computers*, vol.26, 2003.

[12] T. Kohonen, *Self-Organizing Maps*, Berlin: Springer-Verlag, 1997.

[13] S. Theodoridis and K. Koutroumbas, *Pattern recognition*, Academic Press, 2006, p. 580-582.