Abnormal Event Detection Using Trajectory Features

Chin-Chuan Han[1,*], Cheng-Yi Lin[2], Gang Feng Ho[2], and Kuo-Chin Fan[2]
[1]Department of Computer Science and Information Engineering,
National United University, Miaoli, Taiwan
[2]Department of Computer Science and Information Engineering,
National Central University, Taoyuan, Taiwan
cchan@nuu.edu.tw

ABSTRACT

In this paper, an abnormal detector is proposed using trajectory features. An intelligent surveillance system could provide not only the recording function but also the detection of abnormal activities. Trajectory feature is an effective feature for detecting the abnormal activities. Since the monitoring spaces are much varied, pre-defined trajectories are not available in all cases. In this paper, the video data of normal activities were collected and segmented for training the detector. The trajectory features of moving objects were extracted and represented as a normalized feature vector. A fuzzy self-organized map based detector, an unsupervised detector, was built up to detect the abnormal activities in real time. Experimental results are given to show the effectiveness and efficiency of the proposed approach. Finally, some conclusions are made.

Keywords: *video surveillance, abnormal activity, fuzzy self-organized map, trajectory features*

1 Introduction

Developing a computer-based monitoring system is an effective approach to monitor the space for saving man power. An intelligent program could provide not only the recording function but also the detection of abnormal activities. Trajectory feature is an effective feature for detecting abnormal activities. The trajectories of moving objects should be pre-defined in traditional approaches. Since the monitoring spaces are varied very much, pre-defined trajectories are not available in all cases. In this paper, the video data of normal activities are collected and segmented to train an unsupervised detector of normal behaviors. A fuzzy self-organized map (FSOM) based detector was built to detect the abnormal activities using the trajectory features.

Bashir *et al.*[1] separated the trajectory into several segments. Each segment was represented the coefficients of principal axes. Next, the Gaussian mixture model (GMM) was utilized for modeling the distribution of trajectories. Self-organizing neural network (SONN) is an unsupervised learning architecture to automatically model the data distribution. It is unnecessary to pre-define the normal or abnormal trajectories. For example, Johnson *at al.*[2] adopted a vector to represent the trajectory and speed of a point. Two competitive networks were adopted to learn the prototype of trajectories. In addition, the feedback mechanism was utilized [3] to effectively model object's behaviors. Hu *et al.*[4] improved the architecture of Johnson's to increase the system performance. Owens *et al.*[5] utilized the self-organizing neural network(SONN) to model the distribution of features for determining abnormal trajectories.

First of all, moving objects were detected and tracked in the histogram-based background subtraction, the shadow pixel removal, and the labeling steps. The trajectory features of moving objects were extracted and represented as a normalized feature vector. Using an unsupervised clustering algorithm, normal activity patterns are thus constructed. Different from the conventional clustering method, the proposed approach combines the FSOM and the possibility *C*-means clustering algorithm. The parameters of SOM were replaced with the membership functions. They are repeatedly adjusted to obtain the desired outputs by the training samples. After completing the training process, a normalized trajectory vector is verified to determine its validity.

The rest of this paper is organized as follows. The moving objects are detected and tracked in section 2. In section 3, object trajectories were extracted and represented as a vector of the same length. A FSOM was trained to determine the validity of objects using their trajectory features in section 4. In section 5, some experimental results were conducted to show the effectiveness of the proposed approach. The concluding remarks are given in section 6.

2 Moving Object Detection/Tracking

A background subtraction-based approach is a popular method for video surveillance system. Four points, background construction, background updating, foreground detection, and shadow removal, should be considered. Stauffer *et al.* [6] proposed a mixed Gaussian model to construct the distribution of background pixels. Toyama *at al.* [7] proposed a

Wallflower algorithm find the foreground object in three levels: pixel level, region level, and frame level. The foreground pixels were found using the color information in the pixel level. The object relocation problem was solved in the region level. The lighting effect was solved in the frame level. The main goal is to reduce the detection errors of foreground. Haritaoglu *et al.*[8] utilized the minimum intensity, the maximum intensity, and the maximum intensity difference to model the background pixels. Mckenna *et al.* [9] used the color and edge information for background construction. All of their approach would like to reduce the lighting effects.

In this paper, a histogram-based background image was constructed for detecting the foregrounds. First, a median filter was performed for removing the noises. The differentiating between two images is calculated. The pixel with small differentiating value was treated as a background pixel. The statistical histograms $H_c^t(x,y,k)$ for a pixel (x,y) on RGB channels are counted at frame $t, c \in \{R,G,B\}, k = 0,1,\ldots,255$. The bin with the highest number on each RGB channel is assigned as the background value from *N* sequential images. If the differentiating value on the RGB color space between the current frame and the background image is large than a pre-determined value, the averaged value from the previous *N* frames, this pixel is classified as the foreground pixel. Otherwise, it is a background pixel and should be updated in the background updating process.

The background updating is to keep the effectiveness of detection against the change of spaces. The strategies is designed as follows: If a pixel (x,y) belongs to the background pixels whose RGB values are $I_c(x,y)$, the corresponding bins are increased by one. At the same time, the other bins for this background pixel are decreased by 1. The updating rules for the background pixels are designed as follows:

$$H_c^t(x,y,k) = \begin{cases} H_c^{t-1}(x,y,k)+1 & \text{if } \; k = I_c^{t-1}(x,y), \\ H_c^{t-1}(x,y,k)-1 & \text{otherwise;} \end{cases} \quad (1)$$

$$B_c^t(x,y) = \arg\max_k H_c^{t-1}(x,y,k), \quad c \in \{R,G,B\}, k = 0,1,\ldots,255. \quad (2)$$

The updating process for the foreground pixels is ignored. Next, the shadow problem is a crucial problem because the image data frequently suffer from lighting. The shadow pixels are frequently mis-classified as the foreground points. The shadow pixels were identified by using the Horprasert *et al.*'s approaches [10]. In their approaches, the distortion values of brightness and chromaticity were calculated from the statistical data. The pixels in moving regions were classified into *the original background, the shadow background, the highlight background* and the *foreground objects* using the original color data. In addition, the approach proposed by Prati [11] *et al.* used the following rules to remove the shadows. (1) The shadow pixels are the pixels whose brightness is smaller than the backgrounds'. (2) The variances of chrominance of shadow pixels are small. The brightness and chromatic distortions are thus defined as follows:

$$\delta Br(x,y) = \left\| B(x,y) \right\| - \frac{I(x,y)\cdot B(x,y)}{\left\| B(x,y) \right\|} \quad (3)$$

$$\delta Cr(x,y) = \cos^{-1}\left(\frac{I(x,y)\cdot B(x,y)}{\left\| I(x,y) \right\| \left\| B(x,y) \right\|} \right). \quad (4)$$

The shadow pixels are determined in the following rules:

$$0 < \delta Br(x,y) < \gamma_0 \text{, and} \quad \left\| \delta Cr(x,y) \right\| < \theta_0 . \quad (5)$$

3 Trajectory Feature Extraction

After finding the foreground objects, the next step is to continuously track the objects. In general, Kalman filter, particle filter, or dynamic Bayesian network are the popular methods to track the objects. In addition, four types of matching approaches are used during the tracking process. These four type approaches could be integrated to improve the tracking performance. They are (1) region-based, (2) contour-based, (3) feature-based, and (4) model-based approaches. The centers of foreground objects (x_i, y_i) are extracted using the tracking algorithms. Three representations of trajectories, the point-based [5], the curve-based [12], and the spatio-temporal-based [3, 4] representations, are frequently used in many studies. In this study, the last one is utilized to represent the object trajectories.

First of all, an object center position and its velocity at each trajectory point can be represented in a vector form $[x_i, y_i, \delta x_i, \delta y_i]$. However, this vector could not completely represent the object movement such the direction, speed, trajectory, etc. The velocity values are calculated as $\delta x_i = x_i - x_{i-1}$ and $\delta y_i = y_i - y_{i-1}$ in many reported papers. In this study, these values are re-formulated as the speed $s_i = \sqrt{\delta x_i^2 + \delta y_i^2}$, and the direction $d_i = \tan^{-1}(\delta y_i / \delta x_i)$. The trajectory features are represented as a vector of length *m*, $T_g = [x_1, y_1, s_1, d_1, \ldots, x_m, y_m, s_m, d_m]$. Since the lengths of trajectory vectors are different, vector T_g is extended to a new vector of length 4*n*, where value *n* is the maximal number of trajectory points. *n-m* points with the positions of the last point, zero speed, and zero directional angle, are appended to the original data as follows:

$$(x_i, y_i, s_i, d_i) = \begin{cases} (x_i, y_i, s_i, d_i) & i = 1,2,\ldots,m \\ (x_m, y_m, 0, 0) & i = m+1,\ldots,n. \end{cases} \qquad (6)$$

4 Abnormal Event Detection Using FSOM

Since the abnormal activities are not easily collected, a un-supervised detector is trained from the normal activities. In this section, a fuzzy self-organized map (FSOM) based detector is built up based on the possibilistic *C*-mean clustering algorithm. This detector can determine the abnormal activities in real time.

4.1 Possibilistic *C*-means Clustering Algorithm

Zhang and Leung [13] improved the possibilistic *C*-means clustering algorithm [14] to overcome their shortcoming. They added the fuzzy membership (FM) function into the objective function. Consider *N* points in *S* dimensional space $X = \{x_1, x_2, \ldots, x_N\}$. They are partitioned into *C* clusters. The possibilistic membership (PM) on set *X* is defined as $U = [u_{ik}]$ of size $C \times N$, and u_{ik} is the PM of x_k in cluster c_i. Two objective functions were modified as follows:

$$J_{IPCM_1}(U^{(p)}, U^{(f)}, V) = \sum_{i=1}^{C}\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(u_{ik}^{(p)}\right)d_{ik}^2 + \sum_{i=1}^{C}\eta_i\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(1-u_{ik}^{(p)}\right)^{m_p} \qquad (7)$$

$$J_{IPCM_2}(U^{(p)}, U^{(f)}, V) = \sum_{i=1}^{C}\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(u_{ik}^{(p)}\right)d_{ik}^2 + \sum_{i=1}^{C}\eta_i\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(u_{ik}^{(p)}\log\left(u_{ik}^{(p)}\right) - u_{ik}^{(p)} + 1\right) \qquad (8)$$

Here, $u_{ik}^{(p)}$ and $u_{ik}^{(f)}$ are the PM and FM of a sample x_k in cluster c_i, respectively. Values m_p and m_f are two weighting exponents for the PM and FM. Both of them are larger than one, and they are assigned as one in this study. A scale parameter η_i is defined as:

$$\eta_i = \frac{\sum_{k=1}^{N}\left(u_{ik}^{(p)}\right)^{m_p}\left(u_{ik}^{(f)}\right)^{m_f} d_{ik}^2}{\sum_{k=1}^{N}\left(u_{ik}^{(p)}\right)^{m_p}\left(u_{ik}^{(f)}\right)^{m_f}} \qquad (8)$$

According to the consequences in Reference [13], the following conditions should be satisfied for minimizing $J_{IPCM_2}(U^{(p)}, U^{(f)}, V)$:

$$u_{ik}^{(p)} = e^{-\frac{d_{ik}^2}{\eta_i}} \qquad (10)$$

$$u_{ik}^{(f)} = \frac{1}{\sum_{j=1}^{C}\left(\frac{\eta_i\left(1-e^{-d_{ik}^2/\eta_i}\right)}{\eta_j\left(1-e^{-d_{jk}^2/\eta_j}\right)}\right)^{\frac{2}{m_f-1}}} \qquad (11)$$

$$v_i = \frac{\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(u_{ik}^{(p)}\right)x_k}{\sum_{k=1}^{N}\left(u_{ik}^{(f)}\right)^{m_f}\left(u_{ik}^{(p)}\right)} \qquad (12)$$

4.2 Fuzzy Self-Organizing Neural Network

These normalized feature vectors are inputted a modified fuzzy self-organizing map (SOM) for training a detector. Consider N training samples represented by the vectors of length S to be partitioned into C clusters. The modified fuzzy SOM is trained to find the better PM and FM by the following algorithm:

Step 0: Initialization: Find *C* cluster centers C_i using the fuzzy *C*-mean algorithm, and the scaling parameters were initialized as follows:

$$\eta_i = \frac{\sum_{k=1}^{N} u_{ik} d_{ik}^2}{\sum_{k=1}^{N} u_{ik}} \qquad (13)$$

Step 1: Randomize the weights of all cluster centers v_i, $v_i = [v_{i1}, v_{i2}, \ldots, v_{iS}], i = 1,2,\ldots,C.$

Step 2: Input the training vectors $x_k = [x_{k1}, x_{k2}, \ldots, x_{kS}]$, into the FSOM, $k = 1,2,\ldots,N$.

Step 3: Calculate the Euclidean distance between sample x_k and the center v_i. $d_{ik} = \sqrt{\sum_{J-1}^{S}(x_{kj} - v_{ij})^2}$, $i = 1,2,\ldots,C, k = 1,2,\ldots,N.$

Step 4: Compute the FM and PM between cluster center v_i and point x_k.

$$\mu_{ik}^{(f)} = \frac{\eta_i\left(\frac{1}{1-\exp^{-d_{ik}^2/\eta_i}}\right)}{\sum_{j=1}^{C}\eta_j\left(\frac{1}{1-\exp^{-d_{jk}^2/\eta_j}}\right)}, \text{ and}$$

$$u_{ik}^{(p)} = e^{-\frac{d_{ik}^2}{\eta_i}}.$$

Step 5: Update the weights values

$$v_i(t+1) = v_i(t) + \frac{\sum_{k=1}^{N}\mu_{ik}^{(f)}(t)\mu_{ik}^{(p)}(t)(x_k - v_i(t))}{\sum_{k=1}^{N}\mu_{ik}^{(f)}(t)\mu_{ik}^{(p)}(t)}$$

Step 6: Adjust the scaling parameters

$$\eta_i(t+1) = \eta_j(t) + \frac{\sum_{k=1}^{N}(\mu_{ik}^{(p)})(\mu_{ik}^{(f)})d_{ik}^2}{\sum_{k=1}^{N}(\mu_{ik}^{(p)})(\mu_{ik}^{(f)})}$$

Step 7: Repeat Steps 2 to 6, until the following criteria is satisfied or the iteration number is large.

$$\max_{1\le i\le C, 1\le j\le S}\left|v_{ij}(t+1) - v_{ij}(t)\right| < \xi$$

4.3 Abnormal Activity Detection

Consider an input trajectory of *m* points $T_0 = [(x_1, y_1), (x_2, y_2), (x_3, y_3) \ldots, (x_m, y_m)]$. This vector is converted to a new vector $T_g = [x_1, y_1, s_1, d_1, \ldots, x_m, y_m, s_m, d_m]$ and extended to a vector $T_n = [x_1, y_1, s_1, d_1, \ldots, x_n, y_n, s_n, d_n]$ of length *n*. The extended vector is inputted the FSOM to find the winner node with the minimal Euclidean distance D_j. If the criteria $D_j / m > q_j$ is satisfied, this trajectory is an abnormal activity. Here value q_j is a threshold value generated from the training samples. That means: If the input trajectory is close enough to the winner node D_j, it is classified as a normal trajectory. Otherwise, it is an abnormal one. The threshold value q_j for node D_j is determined as follows: Find the distance D_{ij} for the trajectory T_i of length *m* whose winner node is D_j. The threshold value q_j for node D_j is assigned as the maximal distance of all normalized distance D_{ij} / m.

In addition, a partial trajectory could be classified which prototype it is by computing its probability. Consider a trajectory $T_g = [x_1, y_1, s_1, d_1, \ldots, x_m, y_m, s_m, d_m]$. Calculate the distances between the trajectory and the nodes of FSOM. The distance is calculated as the weighted sum between the trajectory vector and the weight vector of each node as follows:

$$r_j = \sum_{i=1}^{m} \Big((x_i - v_{4i-3,j})^2 + (y_i - v_{4i-2,j})^2 + (s_i - v_{4i-1,j})^2 \Big) \\ + (d_i - v_{4i,j})^2 K(i)$$

In order to obtain the high accuracy rate, the newer trajectory points are assigned with the larger weighted values. The weights is decreased with the time as

$$K(i) = e^{-\frac{i-S}{S}}, \text{for } i = 1,2,\ldots,S.$$

Therefore, the probability for each prototype is calculated as

$$P_j = \frac{1/r_j}{\sum_{i=1}^{C} 1/r_i}, \text{for } j = 1,2,\ldots,C.$$

5 Experimental Results

100 images with moving objects were collected to construct the background image using the histogram-based technique. The threshold value in the experiments was set in a range 10 to 20. Four illustrations of background construction and foreground detection are given as shown in Fig. 1. In addition, the background update in the proposed approach is needless. The foreground pixels (read points) and the shadow pixels (blue points) are effectively classified as shown in Fig. 1(c). The parameters in this experiment are set as (γ_0, θ_0) = (0.3, 0.04), and the parameters for the dark pixels are set as (0.35, 0.1).

To illustrate the detection of abnormal events, 30 video samples with normal trajectory features were extracted for training. A space image and three trained prototypes of normal trajectories are illustrated in Fig. 2(a) and (b), respectively. On the other hand, 30 normal and 12 abnormal samples were tested to show the detection results of the proposed method. Some examples of abnormal activities are illustrated in Fig. 3. In Fig. 3(a), a car turned left into a parking lot of bicycles. Similarly, a car turned right into the parking lot (See Fig. 3(b)). Next, a car stopped at the restricted zone (the yellow zone) as shown in Fig. 3(c). The speed of a car is slowed down due to a motorcycle as shown in Fig. 3(d). Two cars moved in an illegal 'U' turn as shown in Figs. 3(e) and 3(f), respectively. Furthermore, the prototypes of input trajectories are also predicted as shown in Fig. 4. In Fig. 4(a), the probability for the vehicle located at the green path is 32%. It is increased to 36% at the blue path. In the other example, the vehicle moves straight foreword. Its prototype probability is increased from 81% to 99%.

The algorithms were implemented on a PC-based machine of Pentium IV, 3.0 GHZ, and 1G RAM. The averaged execution time needs about 0.047 to 0.062 seconds per frame. 28 of the 30 normal activities were correctly determined. On the contrary, 11 of the 12 abnormal activities were correctly detected. The false rejection rate (FRR) is 2/30=6%, and the false acceptance rate (FAR) is 1/12=8.3%, respectively. The main mis-detection occurred at the classification error of foreground pixels. That led to the incorrect trajectory representation and the incorrect detection results.

6 Conclusions

In this paper, a FSOM verifier has been proposed for determine the normal/abnormal trajectories of objects. The video data were automatically segmented to represent the objects' trajectories. The prototypes of normal activities were trained from the training samples. The verifier could be constructed from the video data in various monitoring spaces.

References

[1] F. Bashir, A. Khokhar, and D. Schonfeld, “Automatic object trajectory-based motion recognition using Gaussian mixture models,” in *Proc. of IEEE Conf. Multimedia and Expo*, 2005.

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

[3] N. Sumpter and A. Bulpitt, “Learning spatio-temporal patterns for predicting object behavior,” *Image Vision and Computing*, vol. 18, pp. 697– 704, 2000.

[4] W. M. Hu, D. X. Tieniu, and S. Maybank, “Learning activity patterns using fuzzy self-organizing neural network,” *IEEE Transactions on Systems Man and Cybernetics– Part B: Cybernetics*, vol. 34, pp. 1618–1626, 2004.

[5] J. Owens and A. Hunter, “Application of the self-organizing map to trajectory classification,” in *Proc. of the Third IEEE International Workshop on Visual Surveillance*, 2000.

[6] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 246–252, 1999.

[7] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance,” in *Proc. Int. conf. Computer Vision*, pp. 255–261, 1999.

[8] I. Haritaoglu, D. Harwood, and L. Davis, “W4: Real-time surveillance of people and their activities,” *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 22, pp. 809–830, 2000.

[9] S. Mckenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, “Tracking groups of people,” *Computer Vision, Image Understanding*, vol. 80, pp. 42–56, 2000.

[10] T. Horprasert, D. Harwood, and L. S. Davis, “A statistical approach for real-time robust background subtraction,” in *Proc. IEEE Internal Conference on Computer Vision*, 1999.

[11] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, “Detecting moving shadows: Algorithms and evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 25, pp. 918– 923, 2003.

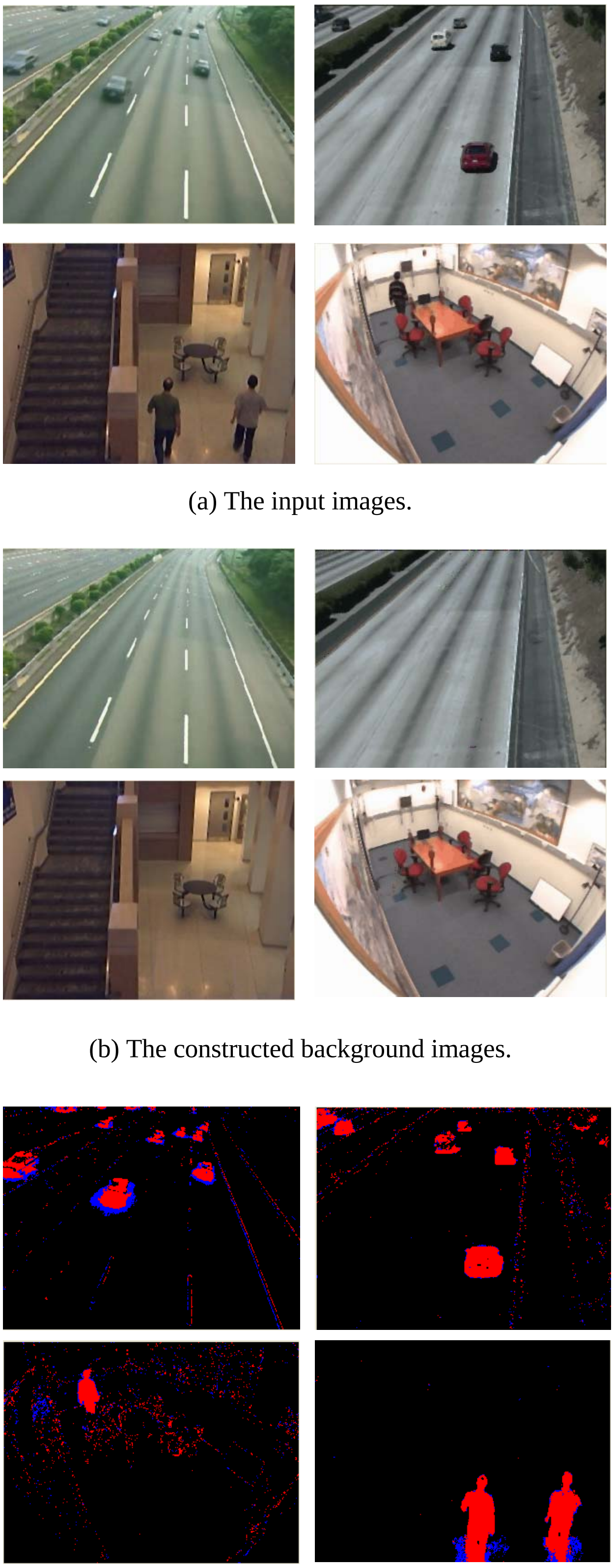
[12] M. Dahmane and J. Meunier, “Real-time video surveillance with self-organizing maps,” in *Proc. of the Second Canadian Conference on Computer and Robot Vision*, 2005.

[13] J. S. Zhang and T. W. Leung, “Improved possibilistic C-means clustering algorithms,” *IEEE Trans. on Fuzzy System*, vol. 12, pp. 209–217, 2004.

[14] R. Krishnapuram and J. Keller, “A possibilistic approach to clustering,” *IEEE Trans. on Fuzzy System*, vol. 1, pp. 98–110, 1993.

Acknowledge

The work was supported by National Science Council of Taiwan Under grant no. NSC 94-2213-E-239 -012

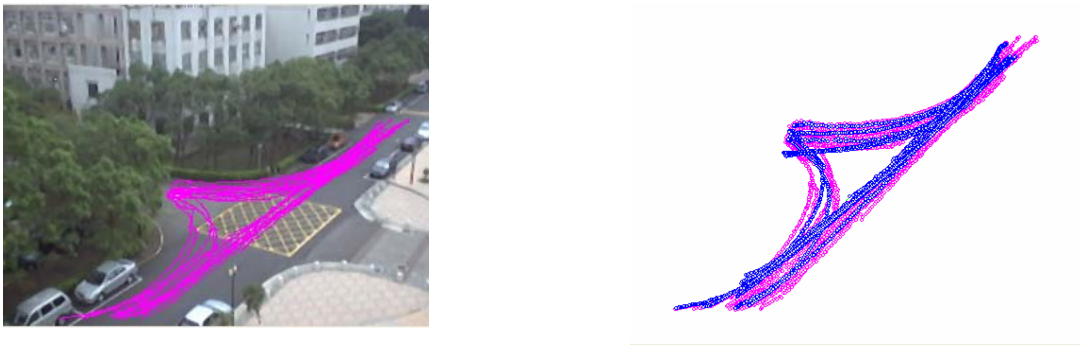


(a) The input images.

(b) The constructed background images.

(c) The detected foreground objects.

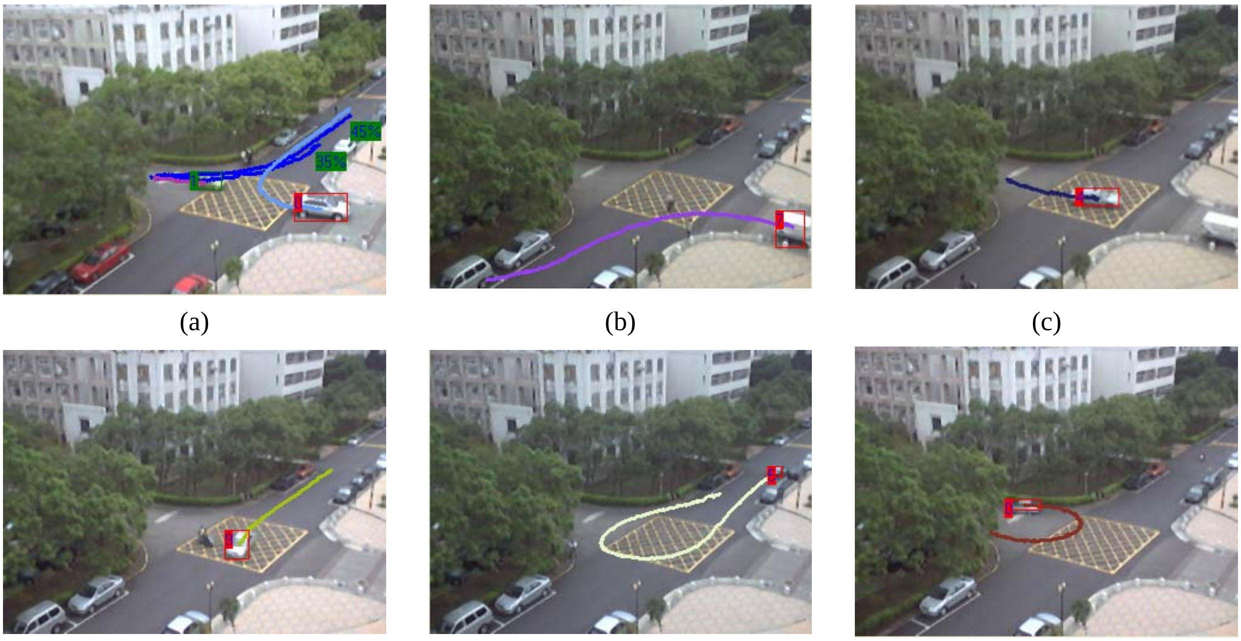
Figure 1: The background image construction and foreground object detection.



(a)

(b)

Figure 2: The training samples and the trained prototypes.



(a)

(b)

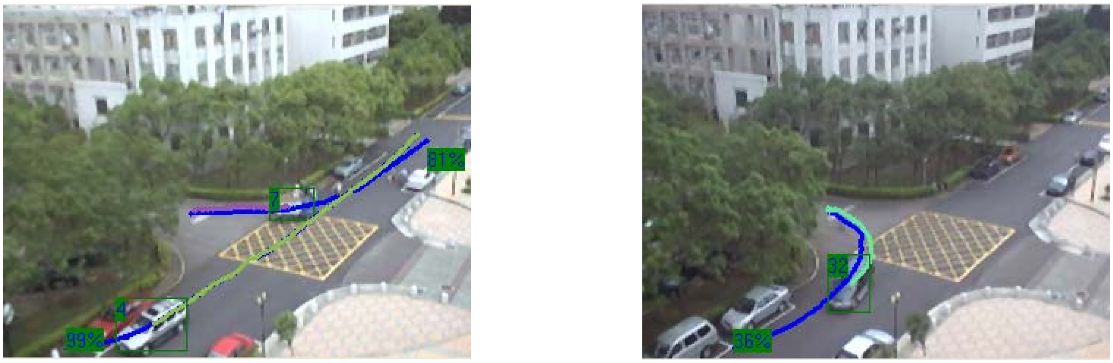
(c)

(d)

(e)

(f)

Figure 3: The abnormal activities.



(a)

(b)

Figure 4: The examples of trajectory prediction.