

Abnormal activity detection in video sequences using learnt probability densities

Supriya Rao and P.S. Sastry
Electrical Engineering Department
Indian Institute of Science
Bangalore, INDIA
080-293-2696, 080-544-4004
supriya_rao@blore.tcs.co.in, sastry@ee.iisc.ernet.in

Abstract—Video surveillance is concerned with identifying abnormal or unusual activity at a scene. In this paper, we develop stochastic models to characterize the normal activities in a scene. Given video sequences of normal activity, probabilistic models are learnt to describe the normal motion in the scene. For any new video sequences motion trajectories are extracted and evaluated using these learnt probabilistic models to identify if they are abnormal or not. In this paper, we have employed the commonly used prototype based representation to describe the movement of individual objects. The model parameters are estimated in the Maximum-Likelihood framework.

1. INTRODUCTION

Video surveillance is concerned with identifying abnormal or unusual activity. When viewed in the pattern recognition framework, the video surveillance task consists of a feature extraction stage and a classification stage. There are surveillance algorithms where the classification rules are prespecified and no learning is involved [1]. In other surveillance algorithms, prior domain knowledge and data are combined to yield classification rules [2].

In some algorithms [3], no prior knowledge about the scene is available, but labeled training samples of every type of abnormal activity that must be classified are available. Algorithms of the above type are however limited in that they can only learn to classify activities for which labeled training samples are available. In the video surveillance scenario, it is easy to see that while sufficiently many samples of normal activities may be easily provided, it would be difficult to provide a comprehensive set of examples that describes all types of unusual activity that we wish to detect. Often an activity is unusual because there are no previous occurrences of this activity! Hence it would be desirable to have a video surveillance algorithm which uses only samples of normal activity in a particular scene in order to make a decision as to whether a particular activity is normal or abnormal. The algorithms that we present fall into this category.

2. FEATURE EXTRACTION AND PROCESSING

There are several static features that are useful in identifying abnormal activity. However, in a generic video surveillance system, information about the movement of objects provides the strongest clue to detecting abnormal activities. Many surveillance algorithms use motion information as the main feature [4], [5]. Our algorithms are aimed at detecting motion patterns of objects that are abnormal in a given scene.

The simplest representation of the motion of an object is in terms of its motion trajectory which is a sequence of its image coordinates. In order to obtain the motion trajectories of the objects in the video sequence, it is necessary to first isolate moving objects and then track these through the video sequence. We have combined some existing algorithms [6] for moving object segmentation and object tracking to automatically extract motion trajectories of object points from the video sequence.

If an object i persists in the video sequence for n_i frames, then its trajectory is given by $T_i = \{(x_{ij}, y_{ij}) : 1 \leq j \leq n_i\}$. The collection of all trajectories is represented as $\mathcal{T} = \{T_i : 1 \leq i \leq N\}$. The motion trajectories are the input to the feature extractor. From every object trajectory T_i , the extracted feature sequence is represented as $S_i, S_i = \{s_{ij} : 1 \leq j < n_i\}$.

The individual feature vectors, s_{ij} , should capture details of the object's instantaneous motion as well as the location in the image where this motion is happening. Therefore, our choice for the s_{ij} vector is $s_{ij} = (x_{ij}, y_{ij}, \theta_{ij}, b_{ij})$, where θ_{ij} and b_{ij} represent the direction and the speed of the motion respectively. These s_{ij} would be referred to as *flow vectors*. The s_{ij} 's are vector quantized using the K-means algorithm. We denote the set of prototypes obtained through vector quantization by $\mathcal{P} = \{p_k : 1 \leq k \leq K\}$. The quantized s_{ij} are represented as f_{ij} . Every trajectory is now represented as: $F_i = \{f_{ij} : 1 \leq j \leq n_i\}$ where $f_{ij} \in \mathcal{P}, \forall i, j$.

3. PROBABILISTIC MODEL FOR MOTION TRAJECTORIES

In this paper, we present algorithms to learn the distribution from which the sample trajectories T_i (of normal motion) have been drawn. We have used a model-based approach and

estimate the model parameters in the Maximum Likelihood (ML) framework.

F_i defined earlier is the prototype representation for a single motion trajectory. $\mathcal{F} = \{F_i : 1 \leq i \leq N\}$ is the entire data set. The first assumption that we make is that the individual trajectories (in the training set) are drawn in an independent identically distributed manner. This essentially means that the motion of one object is not affected by the motion of other objects. Since the set of trajectories is our data set and we utilize the ML formulation for estimation, this independence assumption is necessary. Such an assumption is made in almost all surveillance algorithms. Suppose the F_i 's were drawn from a density $g(\cdot)$, then the likelihood of the data set $\mathcal{F} = \{F_i : 1 \leq i \leq N\}$ is

$$L(\mathcal{F}) = \prod_{i=1}^N g(F_i) \quad (1)$$

In general, there could be different kinds of normal motion. For example, in a highway scene with traffic moving in two directions, at least two distinct kinds of motion are normal. Hence, we model the density $g(\cdot)$ as a mixture density given by:

$$g(F_i) = \sum_{m=1}^M \lambda_m g_m(F_i) \quad (2)$$

In the above equations, λ_m 's are the mixing parameters and $g_m(\cdot)$'s are the component densities. The next question is what kind of model should be employed for the individual densities in the mixture, namely, $g_m(\cdot)$. Each F_i is given by $F_i = \{f_{ij}, 1 \leq j \leq n_i\}$ and different F_i may be of different lengths. Hence, it is difficult to choose a single parametrized density function for F_i . Since every f_{ij} takes values from a finite set, however, it is possible to characterize their distribution using a probability mass function. In this particular case, we do not need to explicitly assume any parametric form for the distribution of f_{ij} . Suppose that $P_m(\cdot)$ is the probability mass function of the f_{ij} . To be able to express the density $g_m(\cdot)$ in terms of $P_m(\cdot)$, it is necessary to model the relation between f_{ij} 's in the same sequence. Two simple assumptions are those of independence and a Markovian dependence. In the first case, we could write $g_m(\cdot)$ in terms of $P_m(\cdot)$ as:

$$g_m(F_i) = \prod_{j=1}^{n_i} P_m(f_{ij}) \quad (3)$$

Here P_m denotes the mass function of the prototype flow vectors. In the second case (Markovian dependence), we can write $g_m(\cdot)$ as:

$$g_m(F_i) = \phi_m(t_{i1}) \prod_{j=2}^{n_i} P_m(f_{ij}/f_{i(j-1)}) \quad (4)$$

Here, $P_m(\cdot)$ is the conditional mass function for f_{ij} conditioned on $f_{i(j-1)}$ and $\phi_m(\cdot)$ is the density function for the

initial 'state' t_{i1} . Here t_{i1} is the same as s_{i1} , the first flow vector. That is, we are leaving s_{i1} as unquantized. This is to ensure that under the density learnt, motion can, in principle, start at any point.

It is possible to use either Eq. (3) or (4) in place of $g_m(\cdot)$ in Eq. (2). In the next two sections, we explain how the model parameters can be estimated in each of the above cases under the ML framework.

Assuming that successive f_{ij} in the same trajectory are stochastically independent is obviously unreasonable. The motion of an object in a frame certainly depends on the motion in the previous frame. However, most current surveillance algorithms employ such an independence assumption. On the other hand the Markovian assumption is much more reasonable. Also, the model estimation in the Markovian case is not particularly more difficult. We show through our results that the Markovian model performs better.

Independent Prototype based Model - IPM

In the case of the IPM, we assume that successive prototypes in the same sequence are independent of each other. Hence, $P_m(\cdot)$ is the probability mass function in the m^{th} mixture component. This mass function is defined over the elements of a finite set of prototypes \mathcal{P} . Thus, $g_m(\cdot)$ is given by Eq. (3).

The model to be estimated is got by substituting Eq. (3) in Eq. (2):

$$g(F_i) = \sum_{m=1}^M \lambda_m \prod_{j=1}^{n_i} P_m(f_{ij}) \quad (5)$$

The model parameters are the mixing components λ_m and the mass functions P_m . The ML estimates of these are found using the Expectation Maximization (EM) algorithm. The update equations can be derived using the EM method. (The details can be found in [6].)

Markovian Prototype based Model - MPM

Here we assume that the sequence of feature vectors f_{ij} of a trajectory have Markovian dependence. Hence the model to be estimated is given by substituting Eq. (4) in Eq. (2) as:

$$g(F_i) = \sum_{m=1}^M \lambda_m \phi_m(t_{i1}) \prod_{j=1}^{n_i-1} P_m(f_{i(j+1)}/f_{ij}) \quad (6)$$

Here, $\phi_m(\cdot)$ is the density from which the first element of the sequence is drawn (in the m^{th} mixture component). It is modeled as a Gaussian distribution with mean μ_m and covariance matrix Σ_m . This Gaussian is defined only over (x_{i1}, y_{i1}) , the starting locations of every trajectory. $P_m(\cdot/\cdot)$ is a conditional mass function over the set of prototypes \mathcal{P} . As before the model parameters are estimated in the ML framework.

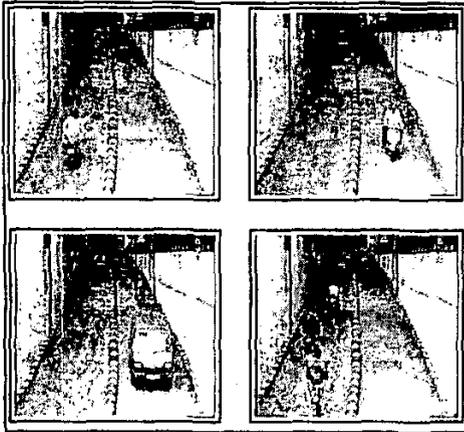


Figure 1. Frames of subway sequence

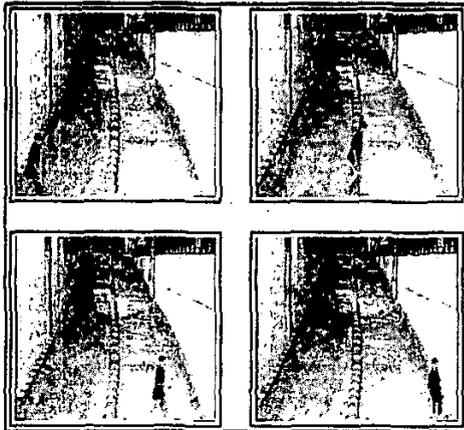


Figure 2. Abnormal activity in subway sequence

4. RESULTS

We show results of both the models that we have developed on a sequence shot at a subway in the Indian Institute of Science campus. We show a few image frames for this sequence in Fig. 1. Normal motion consists of movement on the two roads in predefined directions.

We obtain samples normal motion by recording a video (with a static camera) at the subway for about 30 minutes. The object tracks are automatically extracted and these tracks are used to generate prototypes. The number of prototypes has been set to 50. During training, the number of mixture components has been set to 2 in both the models. Learning the models in both cases requires around 500 iterations of the EM algorithm. Once the model is learnt, given any motion trajectory, it is declared as abnormal if its likelihood under the learnt density is below a threshold.

In this example, two abnormal trajectories have been used to test the algorithms. The first abnormal movement consists of a person crossing the road. This has been shown in Fig. 2. The second is a car moving in the opposite direction on the road. This has been simulated by running the video in reverse.

In both the IPM as well as the MPM, the trajectory corresponding to the first abnormal sequence (man crossing road) has low likelihood and is therefore identified as abnormal.

However, the second abnormal movement (car moving in backward direction) leads to some interesting results. The sequence of prototypes that the trajectories get mapped to all have high likelihood under a learnt model. This is to be expected given that we find the nearest prototype for each feature vector s_{ij} . The specific order of the sequence of prototypes would be wrong. However in IPM, under the independence assumption, the order has no effect on the final likelihood. This is why IPM fails to detect the abnormality. In the case of the MPM, the Gaussian density ϕ , defined earlier, has a very low likelihood for the very first flow vector, causing the sequence to immediately get identified as abnormal. Even ignoring the ϕ , it is seen that the sequence gets identified as abnormal due to the fact that the prototypes occur in the sequence in the wrong order. Even though the same prototype based representation is used in both, due to the Markovian dependence among flow vectors, the MPM performs better than IPM.

5. COMPARISON AND CONCLUSIONS

Both the MPM and the IPM are very simple algorithms to implement. The use of prototypes allows us to learn the probability mass function (or conditional mass function in the MPM) over the set of prototypes \mathcal{P} without assuming anything regarding the form of the distribution. This is the reason for the popularity of the prototype based representation.

However, as shown through our results, the Markovian assumption yields better results than the independence assumption. Consider an abnormal motion trajectory $F' = \{f'_k; 1 \leq k \leq m\}$. In the prototype based representation, we have $f'_k \in \mathcal{P} \forall k$. This is because we first obtain the flow vectors of the trajectory and then find the 'nearest' prototype to each flow vector. In the case of the IPM, for the trajectory F' to have a low probability, at least one of the f'_k must have low probability in all the mixture densities learnt and this may not happen in general, because the mass function is learnt using the normal motion trajectories. Hence, by representing all motion trajectories by prototypes that have been derived from the normal motion trajectories, the prototype based representation seems to cause a 'normalizing' of the motion trajectories. Of course, we could use a distance based threshold during the process of quantizing the flow vectors into prototypes and declare a trajectory as abnormal if we can't find a prototype which is close enough. In fact, in the previous example of a car moving in the opposite direction on the highway, by setting a threshold of 0.85, we are able to identify the tra-

jectory as abnormal. However, the method is sensitive to the threshold that we set. Also, it is possible that different prototypes may have different thresholds and that is difficult to capture in such a method. More importantly, in this method, we are not using the learnt probability density at all! Further, such a method may be able to detect only some types of abnormal motion trajectories.

The Markovian assumption is useful in reducing the effect of normalizing in the prototype based representation. In the case of the MPM, for the same motion trajectory F' , we require that the conditional probability of at least one successive pair of prototypes is low in all the mixture densities. Hence, even though the conditional distribution has been estimated for the normal trajectories only, some of the abnormal trajectories do not get 'normalized' as in the case of the IPM. This is highlighted in the example of the car moving in the opposite direction. In this sequence, the order of the prototypes (in terms of the successive (x, y) coordinates) is incorrect. Therefore, in the MPM, such a sequence is identified as abnormal even though the IPM is not able to do this.

Also, the MPM is sensitive to where the motion trajectory begins. This is because of the factor $\phi(\cdot)$. This behavior might actually be preferred in some cases. For example, in a room, we might expect that people enter only through the door. On the other hand, in other situations where we might have to contend with incomplete trajectories, this might prove a hindrance. In cases where this behavior is undesirable, we could actually disregard the learnt $\phi(\cdot)$ densities during the testing phase. In this case, the likelihood of a sequence would be calculated as:

$$g(F_i) = \sum_{m=1}^M \lambda_m \prod_{j=1}^{n_i-1} P_m(f_{i(j+1)}/f_{ij}) \quad (7)$$

In this paper we have developed algorithms for detecting abnormal activities in an input video. The technique is based on learning a probabilistic model of normal or usual motion trajectories. These models are learnt based only on examples of normal activity and this is a novelty of the method. Also, unlike most other methods, we do not have to assume independence of successive flow vectors. The model employing Markovian dependence is shown to be good through some empirical results.

REFERENCES

- [1] D. Ayers and M. Shah, "Monitoring Human Behavior from Video Taken in an Office Environment", *Image and Vision Computing*, 19, 833-846, Oct 2001.
- [2] H. Buxton and S. Gong, "Advanced Visual Surveillance Using Bayesian Networks", *Proceedings of the 5th IEEE International Conference on Computer Vision*, Massachusetts; USA, Jun 1995.
- [3] P. Remagnino and G.A. Jones, "Classifying Surveillance Events from attributes and behaviour", *Proceedings of the British Machine Vision Conference*, Manchester, UK, Sep, 2001.

- [4] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition", *Image and Vision Computing*, 14, 609-615, Aug 1996.
- [5] C. Stauffer and W.E.L. Grimson, "Learning Patterns of Activity using Real-Time Tracking", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22, 747-757, Aug 2000.
- [6] S. Rao, "Learning probabilistic models of motion trajectories for video surveillance", *MSc (Engg) Thesis, Dept. of Electrical Engineering, Indian Institute of Science, Bangalore, India, 2002*