Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection

Arslan Basharat, Alexei Gritai, and Mubarak Shah
Computer Vision Lab, School of Electrical Engineering and Computer Science,
University of Central Florida, Orlando, FL, USA
{arslan,agritsay,shah}@eecs.ucf.edu

Abstract

We present a novel framework for learning patterns of motion and sizes of objects in static camera surveillance. The proposed method provides a new higher-level layer to the traditional surveillance pipeline for anomalous event detection and scene model feedback. Pixel level probability density functions (pdfs) of appearance have been used for background modelling in the past, but modelling pixel level pdfs of object speed and size from the tracks is novel. Each pdf is modelled as a multivariate Gaussian Mixture Model (GMM) of the motion (destination location & transition time) and the size (width & height) parameters of the objects at that location. Output of the tracking module is used to perform unsupervised EM-based learning of every GMM. We have successfully used the proposed scene model to detect local as well as global anomalies in object tracks. We also show the use of this scene model to improve object detection through pixel-level parameter feedback of the minimum object size and background learning rate. Most object path modelling approaches first cluster the tracks into major paths in the scene, which can be a source of error. We avoid this by building local pdfs that capture a variety of tracks which are passing through them. Qualitative and quantitative analysis of actual surveillance videos proved the effectiveness of the proposed approach.

1. Introduction

Automated video surveillance is crucial for the security of various sites including airports, train stations, military bases, and many other public facilities. There have been significant advances in automated visual surveillance systems in the recent years [2, 13]. A modern surveillance system is expected to not only perform basic object detection and tracking, but also to interpret object behaviors. This higher level interpretation can have several applications including abnormal behavior detection, analysis of traffic trends, and improving object detection and tracking. In this paper, we focus on the problem of interpreting the output of the object detection and tracking module in order to gather knowledge about the scene. This knowledge is used to build a scene model which can be used to detect abnormal motion patterns and to enhance the surveillance performance by improving object detection.

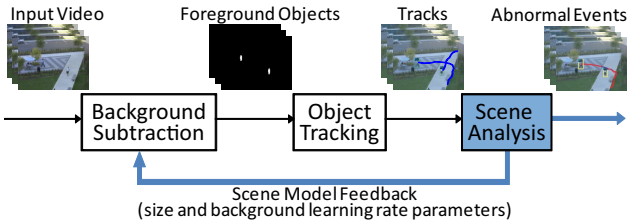


Figure 1. Proposed scene analysis approach detects abnormal events and provides scene model feedback. Traditional object detection is improved by using the pixel-level parameter feedback.

Analysis and modelling of motion patterns for surveillance scenes has been studied by several researchers. Buxton [1] provided a detailed review of the models that have been used for learning scene activity. Johnson *et al*. [9] presented a vector quantization based approach for learning typical trajectories of pedestrians in the scene, but they require entry/exit points to be marked manually. Grimson *et al*. [5] used location, velocity and size to classify activities. The activities are classified using a B-tree based approach called Numeric Iterative Hierarchical Cluster method and the co-occurrence statistics in the quantized feature space. In [14], Remagnino *et al*. use velocity and aspect ratio to classify different tracks into vehicle or person. They utilize a Bayesian classifier for this task and an HMM model to capture common events in the scene. Makris *et al*. [12] have presented a technique in which different regions of the scene are labelled as entry/exit zones, junctions, paths and stop zones. This model provides a set of scene attributes but lacks the object size-based anomaly detection. Saleemi *et al*. [15] proposed a single Kernel Density Estimate (KDE) model for the whole scene, which requires to save all training data. Their approach does not address anomalies due to

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

object size and only focuses on the object velocity.

Hu *et al*. [7] present a recently published technique in which the tracks are spatially and temporally clustered into different motion patterns. Each of these motion patterns is divided into several segments; each segment is modelled by a Gaussian model of speed and size. Anomaly detection and path prediction are the two applications of this approach. Wang *et al*. [18] have presented another approach in which the tracks are clustered into vehicle and pedestrian paths. Their model provides the source/sink information along with capability of abnormality detection.

One common factor in most of the related work is the estimation of main motion paths in the scene. Techniques presented in [7, 10, 17, 18] use multiple features of observed tracks for clustering tracks into the main paths of the scene. We argue that the explicit estimation of these paths is not necessary for typical applications of a scene model including anomaly detection and improving of object detection. In addition, these approaches only capture the instantaneous velocity, however in the proposed approach we integrate larger transition times. This captures the *global* properties of the track and therefore does not require the estimation of the main paths in the scene.

Scene modelling can also be used to feedback the scene knowledge into object detection module. In [6], Harville proposed an approach with positive and negative feedback to background subtraction for adjusting the learning rate and improving foreground detection. Tian *et al*. [19] detected the static regions that were wrongly modelled as the background. In addition to learning rate, there are other parameters that affect the background subtraction and could benefit from the feedback. In this approach we use the same scene model to provide feedback in order to update minimum object size and background learning rate parameters. The unique aspect of our approach is the use of the same scene model for both anomaly detection and improving object detection.

The framework presented in this paper has three novel contributions. First, we propose a new and intuitive approach to model object parameters (motion and size) by using a pdf at every pixel location. Stauffer and Grimson's [16] approach has been used for modelling appearance for several years, but the proposed model of motion and size at pixel-level is novel. Unlike most of the previous approaches, our model does not require extraction of major paths in the scene and is learnt directly from the individual tracking observations. Second, the motion parameters are used to capture the *local* velocity of an object, as well as the *global* velocity through the track. This helps in detecting the anomalous motion patterns that cannot be captured by local analysis only. Third, we utilize this model to provide pixel-level parameter feedback to the background subtraction module in order to improve object detection. Instead of constraining the object detection module by having fixed parameter values throughout the scene, we present a method to provide different pixel-level parameter values using the learnt scene model. Two parameters: Minimum size of the foreground objects and the background learning rate, have been used to improve object detection by our approach.

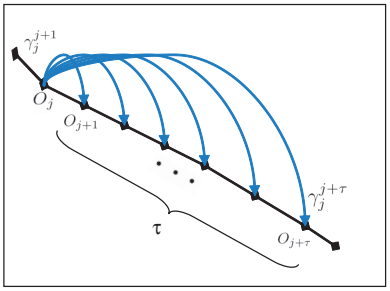


Figure 2. A set of observations with transition (blue) vectors connecting them are shown on a synthetic track. O_j and O_k represent two observations of the same object along the track. γ_j^k is the transition vector between O_j and O_k.

2. Learning the Scene Model

In this section, we present the details of the structure and learning of the proposed scene model. The visual tracking information serves as the input for our framework. We have used the object detection and tracking system presented in [8]. For a given surveillance video, the tracker produces a set of m tracks $\{T_1, \ldots, T_i, \ldots, T_m\}$, where every track is a set of observations of the same object. For instance, any ith track is a set of n observations $T_i = \{O_1, \ldots, O_j, \ldots, O_n\}$, where $O_j = (t, x, y, w, h)$ contains the time stamp t of observation, location (x, y), width w, and height h of the object. We also use the size (w, h) feature, as it provides useful information for finding anomalous behavior and improving object detection. For instance, this model assists in detecting a pedestrian on the road or a bicyclist on the sidewalk, even when the motion is not very discriminative. Using the set of observations, we want to generate a set of transition vectors that will be used to train the statistical model and provide the details about the motion and size of the objects. For every observation, we compute a set of transition vectors that capture the transition from the given observation to future observations along the same track. Relative velocity is computed for the next observation, as well as a set of subsequent observations. In order to keep the problem computationally tractable, we limit the computation to a temporal window with τ observations. Fig. 2 shows a synthetic track with marked observations and transition vectors from a particular observation O_j. This provides a means to detect abnormal tracks through the *global* analysis. In many cases mere use of *local* analysis would not be sufficient. One such synthetic example is illustrated in Fig. 4.

For any observation O_j, relative velocity is computed

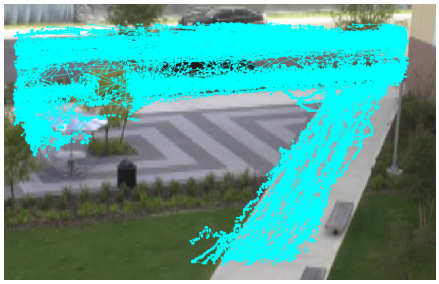


Figure 3. A subset of tracks used in the training of the scene model. Multiple transition vectors from each observation contribute towards learning the pdf at that location.

against all $\{O_{j+1}, \ldots, O_{j+\tau}\}$ to generate a set of transition vectors $\{\gamma_j^{j+1}, \ldots, \gamma_j^{j+\tau}\}$, where transition vector $\gamma_j^{j+\tau} = (x_{j+\tau}, y_{j+\tau}, \tau, w_j, h_j)$. The destination location $(x_{j+\tau}, y_{j+\tau})$ is obtained from the observation vector $O_{j+\tau}$, the duration between the two observations O_j and $O_{j+\tau}$ is τ. (w_j, h_j) represents detected size of the object in source observation O_j. τ is the length of the temporal window along the track; in the experiments we have used $\tau = 20$.

We model the motion patterns in the scene using the motion and size features, as described above. We use a 5-dimensional random variable Γ_l for every pixel location l, where $\gamma = (x', y', \delta t, w_l, h_l)$ represents one particular outcome of Γ_l. Every transition vector generated from the observations presents a five dimensional random variable. The probability density function (pdf) over this feature space is modelled as a multivariate Gaussian Mixture Model (GMM). This pdf is created for every pixel location in the scene and it models the probability of that location being the source of a transition. The pdf estimated at every location captures the probability of observing an object of a given size which is moving to a specific location in a given duration. The pdf at an intersection of multiple paths can capture the possible transitions in different directions, speeds and sizes of objects.

Learning of the model is performed after a sufficient amount of tracking data has been accumulated. The appropriate duration depends on the amount of traffic in the scene and the required accuracy of the model. For any given location l in the scene, all the observations of the tracks through that location contribute to the pdf at that location. The pdf for the random variable Γ_l is created by utilizing the training instances γ's with l being the source location. The training method described below is repeated for all pixel locations.

A multivariate GMM is used to model the pdf of the random variable Γ_l. The probability of an observation γ belonging to the GMM is given by

$$P(\Gamma_l = \gamma|\theta_l) = \sum_{i=1}^{n} \alpha_l^i p(\gamma|\theta_l^i), \quad (1)$$

where n is the number of components detected in the mixture, θ_l^i is the set of parameters defining the ith component with weight α_l^i, and $\theta_l \equiv \{\theta_l^1, \ldots, \theta_l^n, \alpha_l^1, \ldots, \alpha_l^n\}$ defines the complete set of parameters required to specify the mixture model. Each component is modelled as a Gaussian distribution of the form

$$p(\gamma|\theta_l^i) = \frac{1}{(2\pi)^{d/2}|\Sigma_l^i|^{1/2}} e^{-1/2(\gamma-\mu_l^i)^T \Sigma_l^{i\ -1}(\gamma-\mu_l^i)}, \quad (2)$$

where d is the dimensionality of the model and $\theta_l^i = \{\mu_l^i, \Sigma_l^i\}$ are the parameters of the model.

The computation of the GMM parameters is performed through an improved Expectation Maximization (EM) based algorithm, which was proposed by Figueiredo and Jain [4]. This particular approach provides a solutions to three major limitations of the basic EM algorithm. First, the number of components does not have to be fixed. This algorithm estimates the number of components by removing the components that are not supported by the data. Second, this approach does not require careful initialization and starts with a large number of components which are spread throughout the data. Third, this algorithm also avoids convergence towards a singular estimate near the boundary of the parameter space. The details of the algorithm are available in [4], but important points are included here for the sake of completion. The E-step is given by

$$\omega_l^i = \frac{\alpha_l^i(t) p(\gamma|\theta_l^i(t))}{\sum_{j=1}^{k} \alpha_l^j(t) p(\gamma|\theta_l^j(t))}, \quad (3)$$

where ω_l^i captures the conditional expectation of the missing data. $\alpha_l^i(t)$ and $\theta_l^i(t)$ are the parameter values at the iteration t of the EM algorithm. The M-step is given by

$$\hat{\alpha}_l^i(t+1) = \frac{max\{0, (\sum_{m=1}^{S} \omega_l^i(m)) - \frac{d}{2}\}}{\sum_{j=1}^{k} max\{0, (\sum_{m=1}^{S} \omega_l^i(m)) - \frac{d}{2}\}}, \quad (4)$$

for $i = 1, \ldots, n$,

$$\hat{\theta}_l^i(t+1) = \arg\max_{\theta_l^i} Q(\theta_l, \hat{\theta}_l(t)), \quad (5)$$

for $m : \hat{\alpha}_l^i(t+1) > 0$, where d is the dimensionality of each mixture component, S is the number of training samples γ used in E-step, and the Q-function estimates the log-likelihood given the current model estimate.

After learning of the complete scene has been performed, the GMM parameters for every pixel location are stored as the scene model. For a given observation, if we only update the pdf of the pixel at the centroid of the bounding box, then the created models could be spatially sparse. To achieve better spatial smoothing of the motion models in the neighboring pixels, we update all the pixels in the bounding box. Note that unlike most of the previous approaches, learning of the proposed scene model does not rely on merging track to estimate the main paths in the scene. This reduces possible sources of error due to incorrect path estimation or ambiguity of track membership between two or more paths. Another strength of the proposed structure of the scene model is the ability to perform online learning of motion patterns and adaptation to the changing object behaviors in the scene.

3. Abnormal Behavior Detection

The training phase generates a scene model $\mathbf{\Theta}$ using the observed motion patterns. This model is a set of GMM parameters $\mathbf{\Theta} = \{\theta_l\}$, where l is the location of all the pixels with sufficient training observations. We use this scene model to detect abnormal motion patterns which conflict with the trends observed in the training data. We propose an online approach for detecting anomalies in the latest observation O_t from the test track T. This observation is analyzed as soon as it becomes available after a set of previous observations in the track $T = \{O_1, \ldots, O_{t-1}, O_t\}$. For the task of anomaly detection, *local* and *global* analysis of these observations is performed. In *local* analysis, we conduct the comparison of the current observation O_t with the previous observation O_{t-1} only (first order). This captures many typical anomalies based on instantaneous velocity and size of the detected objects but, it has a limited capability for detecting more complicated anomalies. The *global* analysis, however captures more *complicated* cases by analyzing the current observation O_t with respect to a series of previous τ observations $T' = \{O_{t-\tau}, \ldots, O_{t-1}\}$ (higher order). The transition between any source observation $O_{t-i} \in T'$ and the current observation O_t is defined by the transition vector $\gamma_{t-i}^{t} = (x_t, y_t, i, w_{t-i}, h_{t-i})$, which contains contains destination location, transition time, and the object size at the source location. The pdf $P(\Gamma_{l(t-i)})$ of transition vectors at the source location $l(t-i)$ from O_{t-i} is used to determine how normal the current transition γ_{t-i}^{t} is. A very low probability value from $P(\Gamma_{l(t-i)} = \gamma_{t-i}^{t})$ is interpreted as representative of an atypical transition. Our goal is to determine if the current observation O_t is abnormal or not by analyzing the trail of observations in the track. Therefore, we use the minimum transition probability

$$\beta_t = \min_i \{P(\Gamma_{l(t-i)} = \gamma_{t-i}^{t})\}, \tag{6}$$

for $i = 1, \ldots, \tau$ and the observation O_t is declared abnormal if following condition is true

$$\beta_t < \lambda, \tag{7}$$

where threshold λ is applied to the least probable transition. This provides a means of detecting atypical transitions that originated from any one of these higher order transitions. Hence, both local and global anomalies can be detected through this framework. Our approach performs online analysis of the motion patterns to detect anomalies as soon as they occur.

We use this framework to detect various types of anomalous behaviors. Fig. 5 presents various types of detected anomalies in a real video. These include pedestrians on the road and grass, skateboarder and bicyclist on the sidewalk, pedestrians sitting down, etc. In addition, we can also catch anomalies like violations of one-way traffic, which is important on the road and in some airport hallways. Fig. 4 presents a synthetic scene to illustrate the case of global anomalies. Randomly generated tracks (Fig. 4(a)) were used for training completely follow one of the four paths. Our goal is to detect the tracks whose behavior is normal locally but not globally. This is important, for instance at the airport where pedestrians from one path are not allowed to switch to another intersecting path. Another example could be of cars that are not allowed to turn on an intersection. Fig. 4(b) and (c) show the outcome of the local and the global analysis respectively. Local analysis the first order transition between observations is not sufficient to detect such anomalies. Instead we use higher order transitions to capture the global structure of the track. This type of analysis can also be useful for detecting cyclic motion or repeated U-turns which can be abnormal.

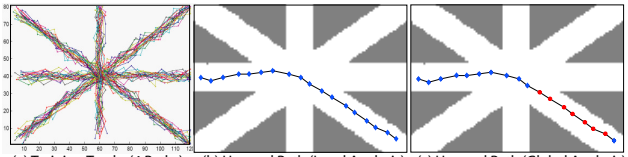


Figure 4. Global anomaly: when the tracks are not allowed to change paths, global analysis detects the violations. Every observation is labelled either normal (blue diamond) or abnormal (red circle). Gray background is the region without motion model. (a) Training set of random unidirectional tracks (along four paths). (b) Local analysis fails to identify anomaly, while (c) global analysis highlights the observation that take an unusual path.

4. Improving Object Detection

An important application of the proposed scene modelling approach is to improve object detection utilizing the patterns in the observed tracks. The knowledge of object parameters (size and speed) at every pixel location is used for this purpose. There are certain components in traditional background subtraction algorithms [16, 3] that could benefit from this scene knowledge. These parameters are traditionally considered consistent throughout the scene, but this limits the performance of object detection. The scene model provides the feedback information (see Fig. 1) for every pixel to update the parameter values according to the scene information. The use of the proposed scene model is presented in the following for two parameters, minimum object size and background learning rate.

4.1. Minimum Object Size

The minimum size (s) of the detected objects is the first parameter which benefits from our scene model. Size s is defined as the area of the blob detected after background subtraction. If this value is set too high, then detection of valid small objects in the far view camera fails. On the other hand, if this value is too low, then some noisy segments and broken parts of larger object blobs are reported as separate objects. Instead of a fixed global value for the parameter s, we present a method for automatically obtaining the appro-

priate value of the s parameter at different pixels.

In order to improve the accuracy of object detection, we use the proposed scene model to estimate the probability of observing an object of a given size at the current location. In the learnt scene model, the pdf at every pixel location captures the joint probability of motion and size. For size-based analysis, we extract the marginal pdf for the size parameters

$$P(w,h) = \sum_{x=1}^{m}\sum_{y=1}^{n}\sum_{t=1}^{\tau} P(x,y,t,w,h), \tag{8}$$

where n rows & m columns is the size of the image and the maximum transition duration modelled in the pdf is τ. As mentioned in [11], this marginal pdf for $\mathrm{x_{wh}} = (w,h)$ can be represented as

$$P(\mathrm{x_{wh}}) = \sum_{i=1}^{n} \alpha_i p(\mathrm{x_{wh}}|\theta_i^{\mathrm{wh}}), \tag{9}$$

where θ_i^{wh} represents the parameters for ith bivariate Gaussian with mean μ_i^{wh} and covariance Σ_i^{wh}

$$p(\mathrm{x_{wh}}|\theta_i^{\mathrm{wh}}) = C\exp\{-\frac{1}{2}(\mathrm{x_{wh}}-\mu_i^{\mathrm{wh}})^T\Sigma_i^{\mathrm{wh}}(\mathrm{x_{wh}}-\mu_i^{\mathrm{wh}})\}, \tag{10}$$

where

$$C = \frac{1}{2\pi|\Sigma_i^{|\Sigma_i^{\mathrm{wh}}}|^{1/2}},$$

$\Sigma_i^{|\Sigma_i^{\mathrm{wh}}}$ is Schur's decomposition of Σ_i with respect to Σ_i^{wh}, and Σ_i is 5×5 covariance matrix from original joint pdf.

The marginal pdf is created at every pixel location and it captures the density of observed object sizes at that location. For illustration purpose, we use this pdf to generate the size map shown in Fig.8. The mean value of width and height from the Gaussian component with highest weight is used in the computation of the most probable size at a given pixel location. This value of size is used as the intensity of the corresponding pixel location in the size map. Note that the size values on the road region are much higher than those on the sidewalks. The size values can be observed to be gradually reducing as the objects move away from the camera.

The parameters of the marginal pdf at every pixel are passed to the object detection module as feedback. Fig.1 shows the feedback flow of the pixel level parameters representing the size pdf at each pixel. The background subtraction algorithm generates a set of foreground blobs of different sizes. For each of the foreground blob at location (i,j) with size (w,h), we compute the probability $P(w,h)$ using the marginal at (i,j). A very low value means that the current blob is most likely a false observation. Suppressing valid objects at unexpected locations can be avoided by defining the s parameter at the current location as

$$s = s_{min}P(w,h) + s_{max}(1-P(w,h)), \tag{11}$$

where $[s_{min}, s_{max}]$ specify the range for s value. This range does not greatly affect the sensitivity of the detection module. In our experiments we used [50, 150] range for two different scenes. Pixels locations with missing models or unexpected object size produce low probability values, which generate a high s value for that pixel. This approach assures that very small noisy observations are not approved as valid objects. High probability values result in small s value which assures that even small sized valid objects are not missed. This provides a means for the object detection module to have different s values for different pixels based on the learnt scene model.

4.2. Background Learning Rate

The background learning rate (ρ) is used to update the learnt background model in order to adapt to slow changes in the scene [16]. For instance, if a table is moved in the room, the new setting is learnt as a part of the background. However this feature can cause a problem when the goal is to consistently track an object that briefly becomes stationary. For instance, if a car stops briefly on a traffic light, it can be quickly learnt as a part of the background if ρ is too large. On the other hand if ρ is too small then the valid changes in the scene would not be incorporated in a suitable time. This dilemma suggests that we locally tweak the value of ρ depending on the behavior of objects in the scene.

The proposed scene model captures different speeds at a particular location. We identify the regions in the scene where objects become stationary, including the exit zones. The learning rate is lowered only for the pixels belonging to these regions. Similar to the approach for the minimum object size, we extract the marginal pdf that captures the motion information. The marginal pdf

$$P(x,y,t) = \sum_{w}\sum_{h} P(x,y,t,w,h), \tag{12}$$

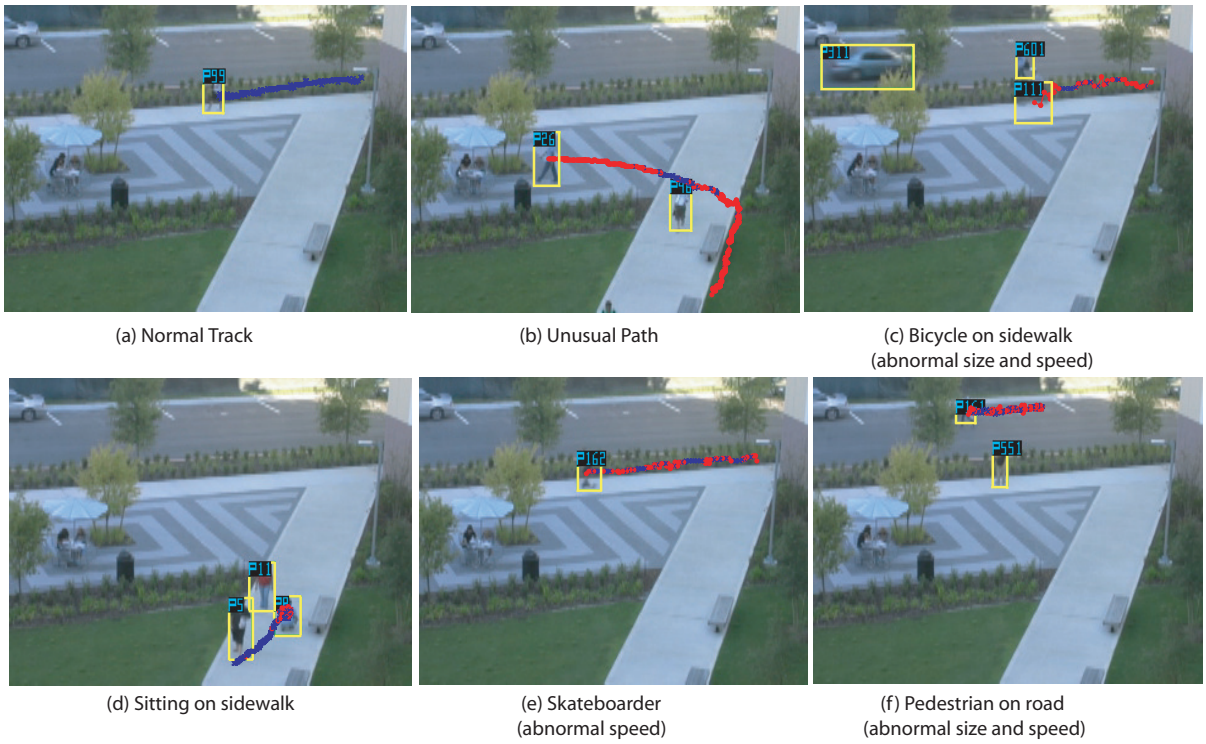
is extracted at every pixel. The GMM component parameters are updated in a manner similar to the minimum size. The object detection could fail because of the high ρ value, therefore we identify the regions where objects stop and reduce ρ. This is done by analyzing the smallest object speed ($\hat{v}$) captured at every pixel. The difference between pixel location and the GMM component mean is used to compute this speed. The interpolated value of ρ can be computed using following expression

$$\rho = \rho_{min}P_v(\hat{v}) + \rho_{max}(1-P_v(\hat{v})), \tag{13}$$

where P_v is a zero mean normal distribution used to signify reducing speed, and $[\rho_{min}, \rho_{max}]$ are the two extreme values of the learning rates to be used. The aim for this formulation is to automatically choose a value of ρ for every pixel depending on the type of object behavior observed during the training phase.

5. Experimental Results

The performance of the proposed framework was tested on real sequences captured from three different surveillance

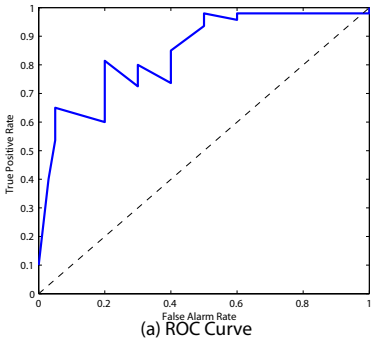


(a) Normal Track (b) Unusual Path (c) Bicycle on sidewalk (abnormal size and speed)

(d) Sitting on sidewalk (e) Skateboarder (abnormal speed) (f) Pedestrian on road (abnormal size and speed)

Figure 5. Scene 1. Detected abnormal observations are labelled red and normal observations are blue. (a) All normal observations of a typical pedestrian (b) The pedestrian follows an unusual path. (c) The observations of a bicyclist are also classified as abnormal, because of the abnormal speed and size of the object. (d) A person stops in the middle of the sidewalk and sits down. Note that the observations were correctly labelled normal before the person sat down. (e) A skateboarder, whose observed size is the same as that of the pedestrian but the speed helps in distinguishing them. Some of the observations are detected normal because of only a slight difference in speed. (f) Unusual size and speed prove to be useful in case of a pedestrian walking on the road. All of the above mentioned tracks are part of the testing video, which is different from the training video.

cameras. A typical scene observed from the first camera is shown in Fig.5. Realtime object detection and tracking was performed using the UCF KNIGHT system [8]. Initial training is performed off-line and testing for anomalous behavior detection was performed using the tracking results from a 30 minute test video. Fig. 6(b) shows the details of the training and testing sets used for this experiment. Matlab implementation runs at approximately 26 fps for this module on a 3GHz Pentium D PC machine. Fig. 5 presents the output of abnormal behavior detection in the test sequence. The proposed approach declares an observation abnormal as soon as it is received from the tracker. Fig. 5 shows a set of detected abnormal behaviors[1] in addition to a normal track. The first one is an unusual path, where a pedestrian is tracked through a region where not enough training tracks were observed. Next, a bicycle is on the sidewalk, which was not present in the training video. The unusual speed and size of the bounding box provides evidence of such anomalies. Another similar anomaly (e) shows a skateboarder going faster than pedestrians. Most of the observations are labelled as abnormal even when the observed size is very similar to that of a pedestrian. (d) shows a case where a pedestrian sits down on the sidewalk



(a) ROC Curve

	Training	Testing
Normal Tracks	1342	217
Abnormal Tracks	0	31

(b) Ground truth

Figure 6. Anomaly detection performance on the scene shown in Fig. 5. (a) ROC curve for the 30 mins test video. (b) Table with ground truth number of tracks used in training and testing.

and (f) shows a case where a pedestrian is detected on the road. This particular anomaly is captured by difference in speed and size of the observed object and the scene model. The results show only a small number of observations are misclassified. The majority decision for the complete track keeps the results accurate. Fig. 6(a) presents the ROC curve depicting the accuracy of anomaly detection.

Fig.8(a) presents the object size map extracted from the

[1] Videos available at: http://cs.ucf.edu/ arslan/surveillance/

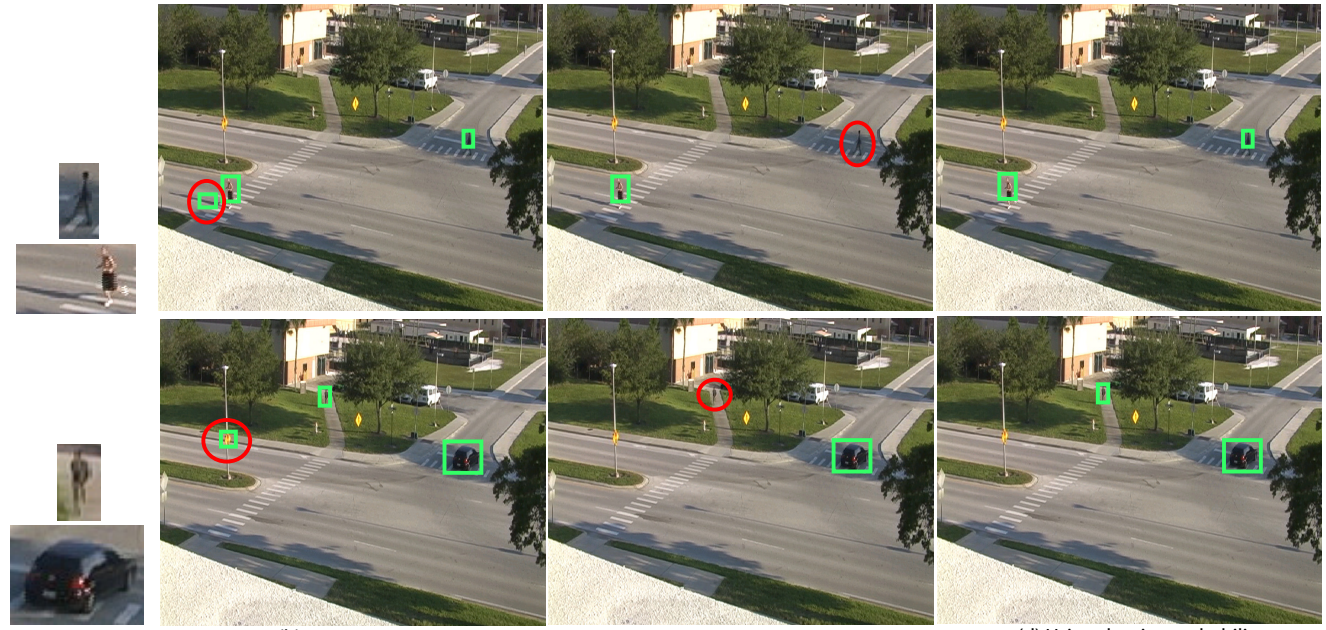


Figure 7. Scene 2. Improvement in object detection by the proposed size model. Each row presents an instance in the same video. Column (a) shows the manually extracted patches of the objects currently present in the scene. Column (b) is the output when a uniform global value of $s = 50$ is used. Noisy foreground blobs are also detected as valid objects (red ellipses). (c) presents output when $s = 150$ is used throughout the scene. Individuals are not detected (red ellipses) when the object size is small. (d) presents results of the proposed size model. In both scenarios the valid objects are detected and the noisy observations are avoided.

learnt scene model scene 1 shown in Fig.5. The high intensity values along the road are generated by the vehicles. As the objects move away from the camera the observed sizes reduce, which reflects here as reducing intensities along the sidewalk. Similarly, Fig. 8(b) shows the size map for scene 2 shown in Fig. 7.

The experiments of improving object detection are performed on video from two other surveillance cameras. Results of the improvement in the object detection using the size parameter feedback are presented in Fig. 7. Two real scenarios are shown here that support the claim that the proposed size map outperforms the case with fixed s value. In the case of (b), the lowest value of $s = 50$ is chosen and in both scenarios, false positive objects are detected. In the first scene, a small broken part of the pedestrian's shadow is detected as a valid object and in the second case, a noisy observation on the lamp post is declared as a valid object. In the case of (c), a comparatively higher value of $s = 150$ is chosen and it clearly misses the pedestrians that are farther away from the camera. Finally, (d) presents the improved object detection using the proposed size map which provides a different s value at each pixel location. All the actual objects are detected without any noisy detections. The automatically learnt size map proves to be very useful in accurately capturing the perspective distortions in the scene.

Fig. 9 presents results of automatic feedback for pixel-wise update of the background learning rate. This camera

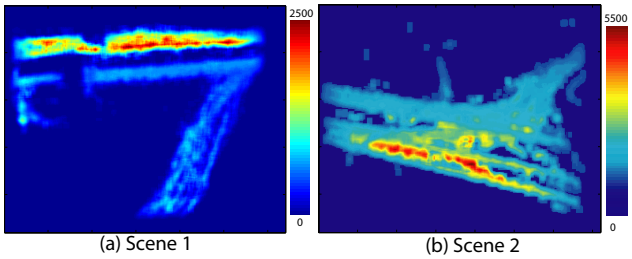


Figure 8. The object size maps are computed for scene 1 (Fig. 5) and scene 2 (Fig. 7). Intensity at every pixel location is the most probable size of the object observed at that location. The highest intensity is observed for the vehicles along the road. Note the gradually reducing sizes due to perspective effect.

covers an intersection with traffic lights where cars may stop up to approximately 40 seconds. The scenario shown in this figure contains a black car arriving, stopping for a red light, and then driving away. Fig. 9(a) shows the output using a typical value of learning rate (ρ = 0.01). The target of continuously tracking the stationary car could be achieved by increasing ρ, but this can induce spurious detections where the background changes rather quickly. Using the proposed parameter feedback approach, we can isolate this increase of ρ to only the regions where it is required (i.e. where traffic stops). In the experiments, we have used $[\rho_{min}, \rho_{max}] = [0.005, 0.1]$ as the extreme values of the learning rate. Fig. 9(b) shows the detection output by using the proposed feedback approach for learning rate. The new detection through this approach have been highlighted.

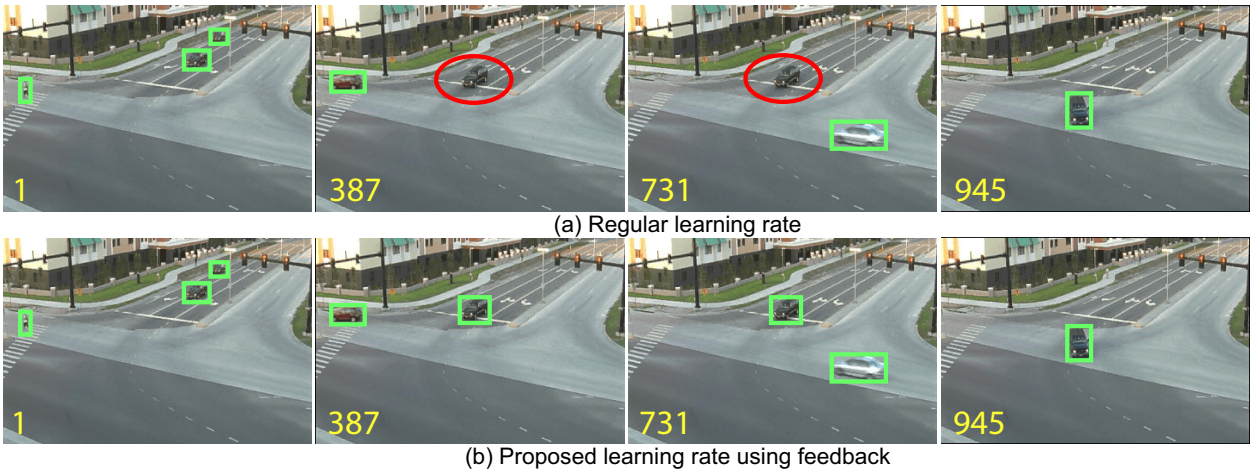


Figure 9. Scene 3. Improvement in object detection using the proposed feedback approach for updating learning rate. Video sequence progresses from left to right. (a) Using the uniform background learning rate ($\rho = 0.01$) for the whole scene. (b) Detection results using the proposed approach for updating background learning rate. Red ellipses highlight the car that was not detected by the regular approach but was later detected by our approach.

6. Conclusion

We have presented a novel framework for unsupervised learning of a scene model that captures object motion and size at every pixel location. The proposed framework provides a means of performing higher level analysis to augment the traditional surveillance pipeline. Experiments on real videos have proven the effectiveness of the proposed approach for local and global anomaly detection. Furthermore, by using the scene knowledge, we also show the improvements in object detection by using the feedback for the minimum object size and the background learning rate. This framework does not require explicit extraction of the main paths in the scene. This approach can easily benefit from online learning and can also be used for conventional applications like predicting object path and scene exit points. In summary, the proposed framework is novel, robust, and can be generalized to more features than just motion and size.

Acknowledgements

This research was funded in part by the US government VACE program.

References

[1] H. Buxton. Generative Models for Learning and Understanding Dynamic Scene Activity. *Workshop on GMBV*, 2002.

[2] R. Collins, A. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Transactions on PAMI*, 22(8):745–746, 2000.

[3] A. Elgammal, R. D., D. H., and L. S. Davis. Background and foreground modeling using nonparametric kernel density for visual surveillance. 90(7):1151–1163, July 2002.

[4] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *PAMI, IEEE Transactions on*, 2002.

[5] W. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in asite. *CVPR*, 1998.

[6] M. Harville. A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models. *ECCV*, 2002.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, and S. Maybank. A system for learning statistical motion patterns. *TPAMI*, 2006.

[8] O. Javed and M. Shah. Tracking and object classification for automated surveillance. *ECCV*, 2002.

[9] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *BMVC*, 1995.

[10] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. *ICPR*, 2004.

[11] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics*. Van Nostrand, 1951.

[12] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 2005.

[13] L. Marcenaro, F. Oberti, G. F., and C. Regazzoni. Distributed architectures and logical-task decomposition in multimedia surveillance systems. *Proceedings of the IEEE*, 2001.

[14] P. Remagnino and G. Jones. Classifying Surveillance Events from Attributes and Behaviour. *BMVC*, 2001.

[15] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Accepted for Publication in TPAMI*, 2008.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999.

[17] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *PAMI, IEEE Trans. on*, 2000.

[18] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. *ECCV*, 2006.

[19] M. L. Y.-L. Tian and A. Hampapur. Robust and Efficient Foreground Analysis for Real-Time Video Surveillance. *CVPR*, 2005.