IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Density aware anomaly detection in crowded scenes

ISSN 1751-9632
Received on 25th May 2015
Revised 19th October 2015
Accepted on 3rd November 2015
doi: 10.1049/iet-cvi.2015.0345
www.ietdl.org

Ayse Elvan Gunduz[1], Cihan Ongun[1], Tugba Taskaya Temizel[1], Alptekin Temizel[1] ✉

[1]*Graduate School of Informatics, Middle East Technical University, Ankara, Turkey*

✉ *E-mail: atemizel@gmail.com*

Abstract: Coherent nature of crowd movement allows representing the crowd motion using sparse features. However, surveillance videos recorded at different periods of time are likely to have different crowd densities and motion characteristics. These varying scene properties necessitate use of different models for an effective representation of behaviour at different periods. In this study, a density aware approach is proposed to detect motion-based anomalies for scenes having varying crowd densities. In the training, the sparse features are modelled using separate hidden Markov models, each of which becomes an expert for specific scene characteristics. These models are then used for anomaly detection. The proposed method automatically adapts to the changing scene dynamics by switching to the most representative model at each frame. The authors demonstrate the effectiveness and real-time performance of the proposed method on real-life datasets as well as on simulated crowd videos that they generated and made publicly available to download.

1 Introduction

Owing to the vast number of surveillance cameras, it has become difficult for human agents to observe and analyse the public areas without the help of an automated system. While there are a high number of studies aiming anomaly detection through tracking of individuals, such methods are generally not feasible in crowded scenes; tracking performance decreases with the increasing crowd density due to occlusions and there is a high computational cost as there are higher number of subjects to track simultaneously. As a result, there has been an interest in developing specific methods aiming at crowd surveillance. Crowd density in an observed scene is dynamic and subject to change in time. Variations in the density have a direct effect on the observed crowd motion characteristics, making crowd density an important parameter that needs to be incorporated into the crowd anomaly detection applications. However, this has mostly been overlooked until recently and the existing solutions use a single model without taking the varying scene characteristics into account.

In this paper, we model the scene using the motion behaviour obtained from the statistics of motion flow data. Then, we generate outlier data to simulate the abnormalities and model the motion behaviour using a hidden Markov model (HMM). With each new observed frame, we match the frame to the closest trained model using similarity of the statistics based on the crowd density information and motion behaviour and perform anomaly detection using the appropriate model. In this study, anomalies are defined as events having unexpectedly different motion characteristics than the usual behaviour in the scene. The usual behaviour is learned from the normal training videos and the method can be configured to detect low- or high-velocity anomalies depending on the target application area.

Density awareness in crowded scenes is a subject that has recently received attention. A density aware person detection and tracking method is described in [1] where optimisation of a joint energy function is performed combining crowd density estimation and localisation of individuals. In a recent work [2], a crowd density map of the scene is extracted and the following analyses are conducted using this map. In [3], a unified framework for tracking individuals and groups of varying densities is presented. On the other hand, none of these methods specifically aim anomaly detection and they do not provide a model switching mechanism contrary to our proposed method.

The advantages of the proposed approach are as follows: (i) it can automatically adapt to the changing scene dynamics by selecting the appropriate model at each frame, (ii) it is able to run in real time by the use of sparse features and statistical representation of motion, (iii) it is privacy preserving as it does not require detection or tracking of the individuals and (iv) it does not require any training video containing anomaly and could be easily adapted to the specific application.

2 Related works

The first step for anomaly detection in crowded scenes is extraction of the features to represent crowd dynamics. This is then followed by a learning method to detect anomalies using these features. There are a variety of feature extraction and learning methods investigated for this purpose. In [4], pixel trajectories are obtained with particle advection, social forces [5] between the particles are computed and Latent Dirichlet allocation (LDA) [6] is used to detect anomalies. Particle advection is a costly process and non-viable for real-time processing. Another social force-based anomaly detection method is described in [7].

In [8], a spatio-temporal grid is applied on the video and optical flows are obtained in each grid. Using the flow data, atomic motion patterns are extracted using a mixtures of probabilistic components analysis [9] and the output of this process is fed into a Markov random field for anomaly detection. In [10], normal behaviour is modelled using low-level features of the scene and anomaly detection is performed by applying a threshold on the likelihood of the observation. In [11], three-dimensional grey-level dependency matrix and optical flow data are used to train a mixture model and outliers are labelled as anomalies. In [12], spatio-temporal cuboids, which allow using both texture and motion data, are extracted. A state search within the states modelled using the training data in a predetermined radius is done and if the search turns empty, the frame is classified as an anomaly. However, there is no generally accepted way of determining this radius. A method that jointly models the crowd information using mixtures of dynamic textures (MDT) is proposed in [13]. In [14], a linear combination of gradients is used to represent the scene. The number of parameters and the error in representation are minimised and a predefined threshold is used to ensure the optimality in training. Test scenes are then fed into this model and their likelihood is computed using another threshold. Optimisation of this threshold for the best performance requires prior knowledge of the test dataset. In [15],

Fig. 1 *A sample frame, corresponding foreground mask and perspective map*
a Sample frame from *Peds1*
b Foreground mask
c Perspective map where high motion activity areas are in lighter colours

the periodicity in the scene is modelled using a Markov model and significant deviations are classified as anomalies. While the periodicity assumption is valid for scenes like underground stations, such periodicity may not be always present in pedestrian zones such as parks and streets.

Use of sparse features to represent crowd motion has started to be investigated recently. In [16], scale-invariant feature transform features matched in the consecutive frames are modelled using a Gaussian mixture model. In [17], ORB (oriented FAST and rotated BRIEF) features are used to represent the motion behaviour in overlapping spatial zones which are then modelled using a coupled HMM. In [18], Kanade–Lucas–Tomasi feature tracking is used to obtain partial trajectories of corners. Then these trajectories are grouped into visual words based on their motion characteristics and a dictionary is obtained. The visual dictionary is modelled using LDA.

In [19], hierarchical feature representation is used to extract the frequently occurring geometric interaction which is then modelled using Gaussian process regression. Anomaly detection is performed on both global and pixel level. In [20], crowd motion segmentation is used to detect the anomalies. This approach uses the correlation information of the fixed sized batches and motion segmentation is performed using min cut/max flow algorithm with alpha expansion. In [21], detected corners are filtered using interaction flow. Later the data is modelled using a random forest classifier. In [22], stability analysis for dynamical systems is used to determine the crowd behaviour. In [23], feature velocities are computed and these velocities are modelled using a multilayer perceptron feed-forward neural network.

These existing solutions use the same model for anomaly detection regardless of the changing scene dynamics. A solution using separate models which are experts for different scene characteristics is expected to have a positive impact on the overall performance. Hence, the main motivation of this work is developing a solution where; (i) different models are trained for different scene characteristics automatically and (ii) a model switching approach for selecting the most representative model is used.

3 Methodology

In the training phase, first, the background is subtracted and perspective normalisation is applied. Following this, sparse features are extracted and motion characteristics are calculated based on these features. Training videos are not expected to contain any anomalies and anomalies are simulated by the proposed outlier data generation method. Different HMMs are trained for *each video* to use later in the testing phase.

In the testing phase, first, sparse features are extracted and matched, followed by perspective normalisation. Then a number of properties are extracted at each testing frame, which are then used to calculate a cost function to find the training data having the most similar characteristics. The anomaly detection is carried out with the HMM that was trained using the best matching data.

These steps are explained in more detail in the following section.

3.1 Background subtraction

As the first step of background subtraction, dense optical flow vectors [24] are calculated for each training video v having $M \times N$ resolution. Then, for each v, the standard deviation of optical flow at pixel (x, y), is calculated for all frames to obtain an $M \times N$ $\boldsymbol{S}_v$ matrix and $M \times N$ average standard deviation matrix AS is constructed

$$\boldsymbol{AS} = \frac{\sum_{v \in N_v} \boldsymbol{S}_v}{N_v} \tag{1}$$

where N_v is the number of training videos. Then, a threshold, Th, is used in classification of each pixel:

$$B(x,y) = \begin{cases} 1 & \text{if } \boldsymbol{AS}(x,y) < Th, \\ 0 & \text{if } \boldsymbol{AS}(x,y) \geq Th, \end{cases} \tag{2}$$

where values of 0 and 1 indicate foreground and background, respectively. Threshold Th needs to be selected according to the noise and camera distance. If the scene is noisy, the threshold needs to be higher as the variability in the background pixels will increase. Also, if the camera is far from the scene, the observed motion magnitude will be smaller and a lower threshold is needed. The dataset is masked using this mask (Fig. 1*b*) and the analyses in the subsequent sections are carried out using the masked datasets.

3.2 Feature extraction

For each frame, the features are obtained using an ORB detector [25] and described with the BRIEF descriptor [26]. These features are matched between the two consecutive frames. The pixel-wise velocity of a feature is calculated by the Euclidean distance between the two feature points. Eight direction bins, d, are used to capture the motion characteristics: [0° −45°):$D1$, [45° −90°):$D2$, [90° −135°):$D3$, [135° −180°):$D4$, [180° −225°):$D5$, [225° −270°):$D6$, [275° −315°):$D7$ and [315° −360°):$D8$. Pixel-wise velocity vectors are calculated for all the frames f of each video v for each direction d and the resulting matrix for each d, f, v, is called $V_{dfv}(x, y)$.

3.3 Perspective normalisation

In some cases, the positioning of the camera results in perspective distortion and the similar real-world velocities at different locations may generate different observed motion magnitudes. To overcome this issue, perspective normalisation is applied in various applications where perspective map is generated with the help of user feedback [27].

In this work, we aim to automate this process by making use of the average standard deviation matrix $\boldsymbol{AS}$ extracted for background subtraction in (1). $\boldsymbol{AS}$ provides information regarding the expected motion magnitude at each pixel. However, $\boldsymbol{AS}$ may be noisy since motion levels may be different at different locations hence, we apply smoothing on the foreground pixels of $\boldsymbol{AS}$ using moving average method on each row to obtain $\boldsymbol{AS}s$. For each row i where $i \leq M$, a new moving average window is constructed. The size of the window is set as half the number of foreground pixels in $\boldsymbol{AS}(i)$. We used k-nearest neighbour smoother in order to preserve the actual

matrix dimensions. An example perspective normalisation map is given in Fig. 1*c*.

Then, the velocities in $\boldsymbol{V_{dfv}}$ for each d, f, v are normalised using $\boldsymbol{AS}s$, producing the normalised velocity pixel matrix $\boldsymbol{NV}_{dfv}$:

$$\boldsymbol{NV}_{dfv}(x,y) = \frac{\boldsymbol{V}_{dfv}(x,y)}{\boldsymbol{AS}s(x,y)} \tag{3}$$

3.4 Outlier data generation for simulating abnormal activities

In many cases, training datasets do not include any abnormal activities as anomalies may not occur frequently. It may also be difficult to purposely generate anomalies for training as a sufficient number of anomalies need to be recorded for each camera. To eliminate the need for videos containing anomalies, we simulate the abnormal activity by generating artificial data points *based on the datasets comprising only normal activities*. The simulation approach is widely used in finance for detecting and preventing risks [28]. In this work, we adapt this technique to computer vision and define *risks* (anomalies) as synthetic data.

Let MV_{dfv} and SV_{dfv} represent the mean and standard deviation matrices of the foreground pixel (x, y) velocity values of NV_{dfv}, respectively. Then, for each v and d, we calculate the overall mean and standard deviation of velocity changes. These calculations are important in order to characterise the *overall* characteristics of the video:

$$\mu'_{dv} = \frac{\sum_{f=1}^{F} MV_{dfv}}{F} \tag{4}$$

$$\mu''_{dv} = \frac{\sum_{f=1}^{F} SV_{dfv}}{F} \tag{5}$$

where F is the total number of frames. The standard deviation values $\sigma_{dv}{}'$ and σ''_{dv} are also calculated using MV_{dfv} and SV_{dfv}, respectively. Then the outlier data for simulating anomalies are generated randomly from the following distributions:

$$A\mu_{dv} \sim N(\mu'_{dv} \times c, \sigma'_{dv} \times c) \tag{6}$$

$$A\sigma_{dv} \sim N(\mu''_{dv} \times c, \sigma''_{dv} \times c) \tag{7}$$

where the coefficient c is determined according to the type of the target anomaly and desired sensitivity. For the detection of high-velocity anomalies, c should be selected between 1 and 2 and a c value closer to 1 allows detection of marginally higher velocity anomalies. On the other hand, a higher c value allows detecting anomalies having velocities that are significantly higher than normal. For the detection of low-velocity anomalies such as congestion in a traffic flow, c should be selected between 0 and 1 depending on the level of congestion deemed as anomaly. As a result, the training dataset is ensured to comprise both data points exhibiting normal behaviour and synthetic data points simulating the abnormal behaviour.

3.5 Model switching and fitting

Pedestrian zones exhibit different characteristics throughout the day. During the rush hour, the scene may be overcrowded resulting in very slow motion due to people blocking each other; while at other times there might be fewer people, affecting the observed motion. Modelling the normal behaviour using the videos from the rush hour and using this model to detect anomalies throughout the day results in a high false-positive rate. On the other hand, using all the training data results in over-generalisation and missing of true-positives. As a remedy, we propose a scheme where different models are trained according to the varying scene characteristics. In the testing phase, a cost function C is used to select a model based on the characteristics of *each frame* and the classification is done using this selected model that was trained using the video with the most similar data characteristics to the given testing frame.

For modelling purposes, HMMs are employed due to the data characteristics. The obtained statistics form a time series. HMMs are used for modelling time series, which have latent (unobserved) variables affecting the observations. In this case, the latent variables represent the state of the scene (normal/abnormal) while the observed variables are the calculated statistics.

First, the major directions of motion are determined based on the feature counts in these directions. Then, for each training dataset, two HMMs, one for modelling normal and the other for abnormal behaviour, are fit. These HMMs become experts for the specific scene characteristics of the particular training video. The inputs of all the HMMs are MV_{dfv}, SV_{dfv} and $A\mu_{dv}$, $A\sigma_{dv}$ features in these major directions for normal and abnormal behaviour, respectively. Later, scene characteristics are calculated for each training video and stored to be later used in model switching decision making. Recall that MV_{dv} and SV_{dv} are F-dimensional vectors holding the mean and standard deviation velocity vectors of a video v for direction d. We define $\boldsymbol{Z_{dv}}$ as the feature count vector of frame f of video v in direction d where $\boldsymbol{Z_{dv}} = \langle Z_{d1v}, \ldots, Z_{dFv} \rangle$. The five features MV_{dv}, SV_{dv}, $\boldsymbol{Z_{dv}}$, $\mu_{dv}{}'$ and μ''_{dv} are calculated for each training video. $MV_{df,testv}$, $SV_{df,testv}$ and the feature count value of frame f, denoted with $Z_{df,testv}$ are used for describing the characteristics of the testing video *testv*.

C_v, the cost function for v, is calculated between all the training videos and the frame f of *testv* as follows:

$$C_v = \sum_{d \in D} \mathrm{DFC}_{dv} + \mathrm{DM}_{dv} + \mathrm{DS}_{dv} + \mu'_{dv} + \mu''_{dv} \tag{8}$$

where

$$\mathrm{DFC}_{dv} = \| \, g(Z_{dv} - Z_{df,testv}) \, \| \tag{9}$$

$$\mathrm{DM}_{dv} = \| \, g(\mathrm{MV}_{dv} - \mathrm{MV}_{df,testv}) \, \| \tag{10}$$

$$\mathrm{DS}_{dv} = \| \, g(\mathrm{SV}_{dv} - \mathrm{SV}_{df,testv}) \, \| \tag{11}$$

where $g(.)'$ is the z-score normalisation function and $\| \, . \, \|$ denotes the norm. This norm is equivalent to the eigenvalue of the normalised distance vector. Then, the training video with the lowest C is selected.

$$y = \underset{v \in V}{\operatorname{argmin}}[C_v] \tag{12}$$

where y is the index of the training video. This cost function ensures that the test frame f and the selected video y are similar in the sense that they have similar densities (measured by DFC_{dv}) and similar motion behaviour (measured by DM_{dv} and DS_{dv}). The anomaly detection for frame f from the testing dataset is carried out using the HMM built using the training video y.

Fig. 2 shows the velocity distributions of two randomly selected test frames from the *Peds1* dataset and the velocity distributions of the matching training videos from the same dataset (i) when all of the model switching conditions in (8) are applied, (ii) when the $\mu_{dv}{}'$ constraint is removed and (iii) when the μ''_{dv} constraint is removed. C chooses a model which has a similar distribution with the testing data, distribution of which has lower mean and standard deviation values. If the latter two characteristics are discarded from the computation in (8), a distribution with higher mean and standard deviation values may be selected despite the similarity of both training and testing dataset distributions, negatively affecting the anomaly detection performance for the testing dataset. The distribution similarity computed using the first three parameters guarantees that both will have similar crowd characteristics (density, motion).

4 Datasets

We used three datasets in the experiments: *Peds1* and *Peds2* contain real videos while *METUCrowd* contains simulation videos.

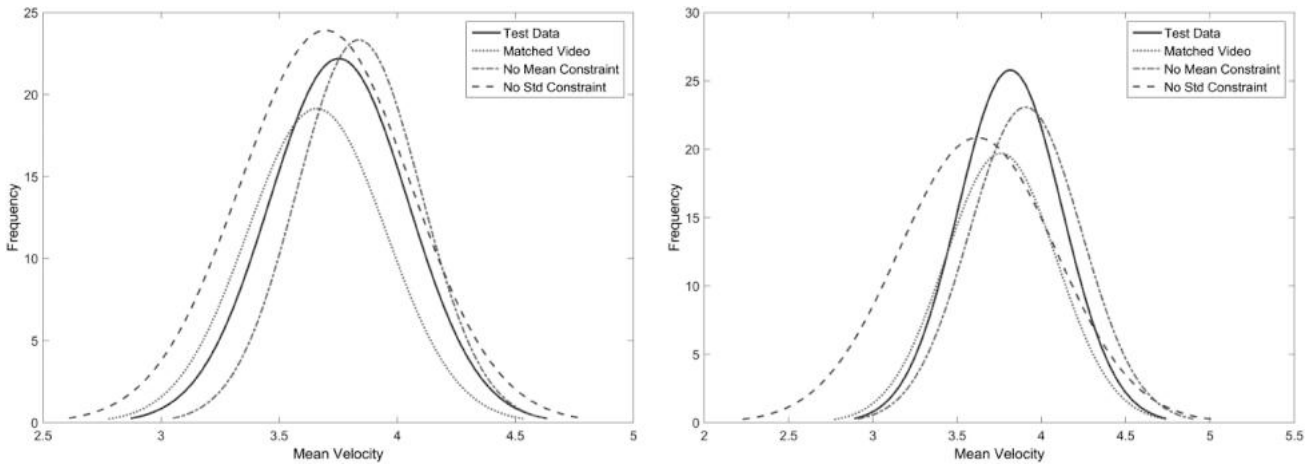

Fig. 2 *Velocity distributions of the two randomly selected frames (test data) and matched videos with various different constraints*

Peds1 contains 34 training videos having only normal behaviour and 36 test videos having both normal and abnormal behaviours [13]. Each video contains 200 frames of size 238 × 158. For this dataset, we performed perspective normalisation as described in Section 3.3.

Peds2 contains 16 training videos having only normal behaviour and 12 test videos having both normal and abnormal behaviours [13]. Each video contains a varying number of frames of size 360 × 240. This dataset does not have a perspective problem; therefore no perspective normalisation was performed.

METUCrowd consists of crowd simulation videos that we generated and made publicly available to download [29]. Simulations have been created using Unity3D [30] which has a realistic physics engine. Agents were derived from 6 human figures (3 males, 3 females) and 20 different textures [31] resulting in 120 different characters. For artificial intelligence and path finding *Navigation Mesh* technique was used. The simulations were recorded at 25 fps and videos contain 500 frames at 640 × 480 resolution. The camera angle was set high to reflect real life installations and this results in a perspective problem. Also, there are trees blocking the view in some regions, causing partial or full occlusion of moving objects.

For training, three different datasets having varying crowd densities (low, medium and high, having approximately 150, 300 and 900 agents, respectively) for both pedestrian area and bicycle lanes were generated. For testing, three datasets also having low, medium and high densities were created. The anomalies in the test videos are caused by agents riding bicycles or skateboards in the pedestrian area and by persons walking on the bicycle lane (these anomalies are not present in the training set). There are 12 videos including two videos each for low, medium and high density for both low and high velocity motions (sample frames of which are shown in Figs. 3*a*, *c* and *e*, respectively) for training and 10 videos including 6 videos with an agent riding bicycle, 3 videos with an agent riding skateboard for testing high-velocity anomalies and a video with an agent walking on the bicycle lane walking amongst the people riding bicycle for testing low-velocity anomalies. An agent riding a bicycle can be seen in Fig. 3*b* on the right and Fig. 3*f* on the left. Another agent riding a skateboard can be seen in Fig. 3*d*.

We also generated a number of variants of the videos for testing the effect of the environmental changes (trees swaying with the wind, light changes caused by a passing cloud), details of which can be found in Section 5.3.

5 Results

5.1 Experiment settings

Feature detection was performed using OpenCV [32] and the data analysis was done in Matlab 2011b using BayesNet Toolbox [33] on a PC with Intel Core i7–3630QM CPU 2.40 GHz with 8 GB RAM. The outlier data generation coefficient c was set to 1.4 to detect anomalies due to fast moving objects for all the datasets. The motion of the pedestrians is bidirectional and there may be three motion directions: in either one of the two directions or in both directions. HMM mixture parameter is set to 3 to represent this behaviour. HMM hidden state parameter is set to 3 to represent different crowd densities (low, medium and high).

5.2 Performance of the density aware model switching

We randomly selected a testing frame and classified as normal or abnormal with and without the proposed model switching approach to demonstrate the effect of proposed model switching technique. Fig. 4*a* shows the comparison of velocity histogram of all training data with that of the test data. This test frame was randomly selected from *Peds1* testing set and the training data comprises all the videos in *Peds1* training set. As can be seen from the figure, when all the training videos are used, the test video is entirely within the normal range, which inevitably results in a number of false negative detections. Fig. 4*b* shows the comparison of velocity histogram of training data selected by the proposed method with that of the same test data. As can be seen from the figure, when model-switching mechanism is used, some values fall out of the normal range, making the detection possible. Also, model switching allows finding a model having more similar distribution to the testing video. For illustration, a histogram of the training dataset is scaled to a similar range to the testing data in both figures.

To further demonstrate the effect of the density aware model switch, the results of two methods are reported: when density switch is utilised and when the models are trained using all the data in the training set as in traditional methods. Also, to demonstrate the effect of model switching for different densities, the test set is divided into three distinct density levels as low (<10 people), medium (10–20 people) and high (>20 people). In both training and test sets, there are a significantly higher number of medium-density videos than both low- and high-density videos. This results in a bias in the models as motion behaviour in medium-density videos dominates the motion behaviour. In order to correctly analyse the effect of density switching, the analysis was done separately on different density levels. The classification performances are shown in Fig. 5 as receiving operator characteristics (ROC) for both approaches where Figs. 5*a*–*c* show the ROC curves for low-, medium- and high-density videos, respectively. Since the number of medium-density videos is higher than both low- and high-density videos, the method trained with all training videos is influenced more from the models of medium-density videos resulting in a more pronounced performance advantage of the proposed scheme for low- and high-density videos.

We investigated the impact of switching on the performance of the proposed model using the *METUCrowd* dataset. As shown in Fig. 6, the proposed method with switch performs the best for all the cases as it switches to the correct training video. Similar to the previous results, improvement is higher for low and high densities. In the high-density scenes the approach with no model switching

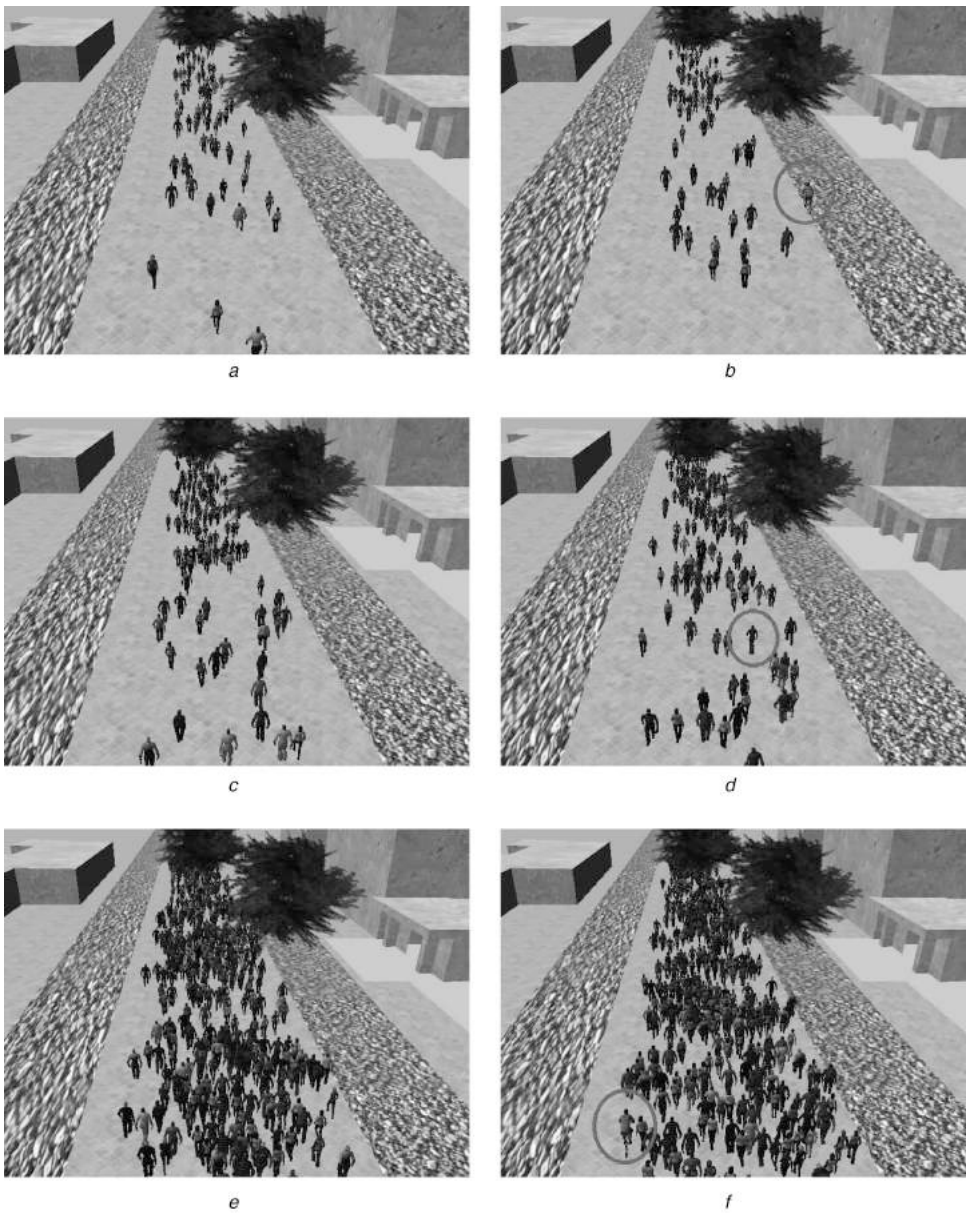

Fig. 3 *Sample frames*
a, *c*, *e* Example frames from the training videos and
b, *d*, *f* From testing videos of *METUCrowd* where the crowd density is low, medium and high. Anomalies are marked in circles

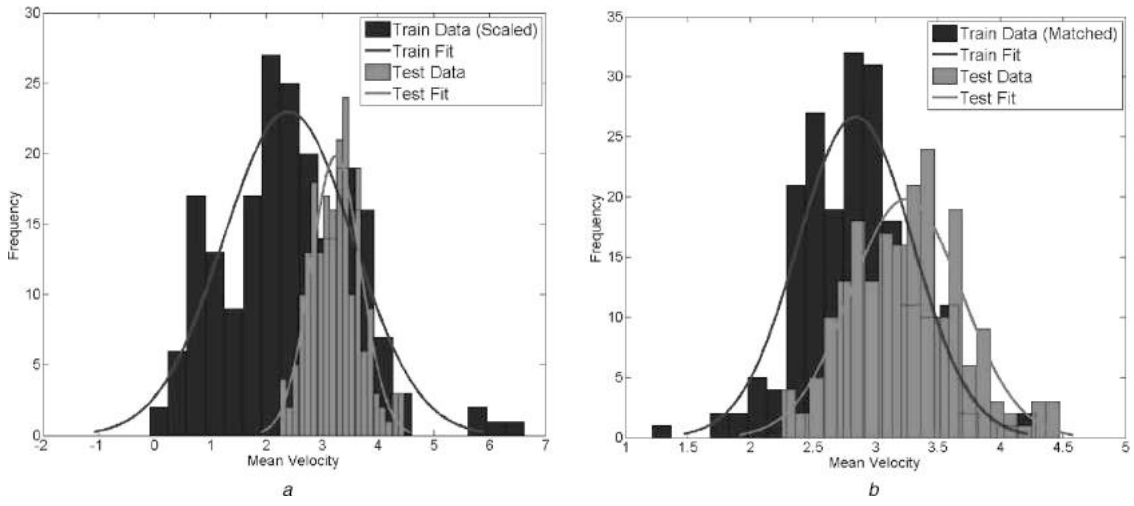

Fig. 4 *Velocity histograms of the training and test data*

performs very poorly (below chance rate). These scenes have a significantly higher number of moving objects and the motion of the anomalous object is lost amongst the motion behaviour of other objects. However, the approach with model switching performs

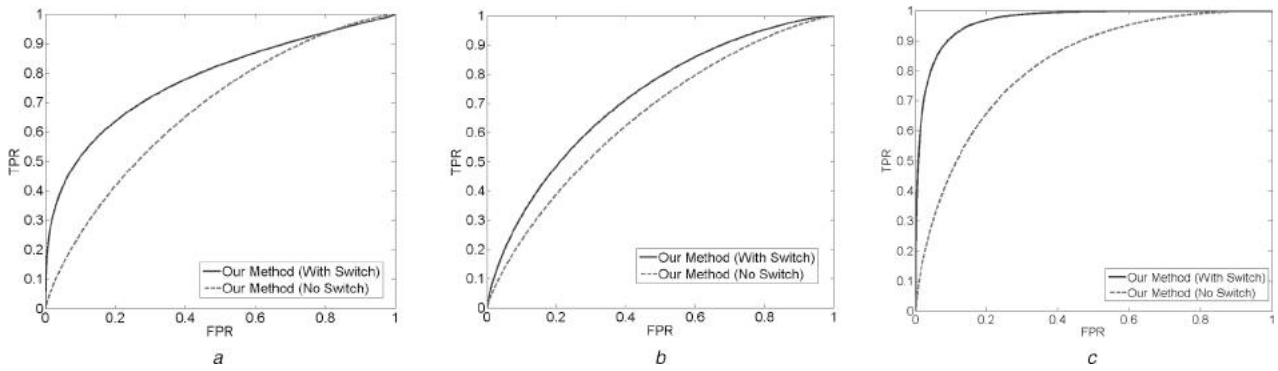

Fig. 5 *ROC curves for Peds1*
a Low density
b Medium density
c High density

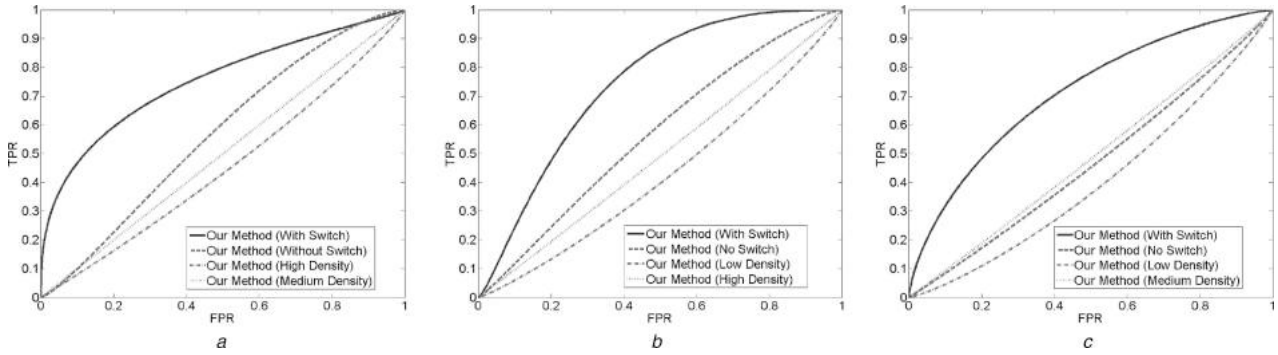

Fig. 6 *ROC curves for METUCrowd*
a Low density
b Medium density
c High density

relatively well despite having a slightly lower performance than low and medium densities. As mentioned in Section 3.5, the model switching mechanism ensures a lower mean and standard deviation in the training set, so the model is still able to detect the high velocity anomalies. On the other hand, if it is explicitly forced to select incorrect training videos, the performance is adversely affected. For example, if the test case is low density, the performance is lower when the algorithm is forced to employ medium or high training videos. The use of all videos (no switch case) also affects the performance negatively.

5.3 Performance of the method under various conditions

To analyse the effects of environmental changes and low-velocity anomalies on the performance, we generated a number of videos. One of the original high density simulation videos has been modified to create two other testing videos (i) having swaying trees and (ii) having light changes. We also generated a simulation video where the anomaly is caused by a slow moving object. In this scenario, a low-velocity pedestrian enters the bicycle lane. A variant of this video having swaying trees and light changes has also been generated to analyse their effects on the performance. The original model trained with videos not having any environmental changes was used for testing the original video and its variants having light changes and swaying trees. The results were compared with those of the original video using both the switching and non-switching approaches (Fig. 7*a*). The low-velocity anomaly was also tested with and without these environmental changes. As expected, environmental changes reduce the performance slightly in either cases. In the video having light changes, feature detection performs slightly weakly and it is not able to detect some features. In the video containing swaying trees, there are a number of false positives due to the motion of the tree and some false negatives due to the occlusions with the object causing anomaly. Even though there are performance degradations in these cases, both the non-switched and switching based approaches are affected. As a result, in all the test cases the proposed method performs better compared to the no-switch approach.

For the low-velocity anomaly detection case, c set to 0.9 and the ROC curves are presented in Fig. 7*b*. As seen in the figure, the proposed method with switching is able to detect low-velocity anomalies with better performance compared to the one without the switch. Both the non-switched and switching based approaches are affected in a similarly by the environmental changes.

5.4 Comparison with the state-of-the-art

The results were compared against those in the literature which reported anomaly detection using the same publicly available datasets: [8, 10–13, 17] for *Peds1* and [4, 8, 10, 11, 13] for *Peds2*.

ROC curves for both datasets are shown in Fig. 8 and equal error rate (EER) values in comparison with the existing literature in Table 1. The ROC curves and EER values show that the proposed method works better than most methods. While the method in [11] is the best in terms of accuracy, it works significantly slower compared to the other methods (see Section 5.6). The variability of crowd density in *Peds1* is higher than *Peds2* and there is no perspective problem in *Peds2* contrary to *Peds1*. As a result, the accuracy values reported in the literature are higher for *Peds2*. In both data sets, the proposed method has improved the results significantly compared to using a single model which does not account for density variability.

5.5 Parameter sensitivity analysis

To analyse the effect of the varying coefficient c on the overall performance, the method was run with different values of c on *Peds1* and *METUCrowd*. The distribution of the results is given in Table 2. The statistics in this table were obtained using the AUC values calculated with c values in the range of [1.1, 2]. The results presented in the table reveal that the method with the switching

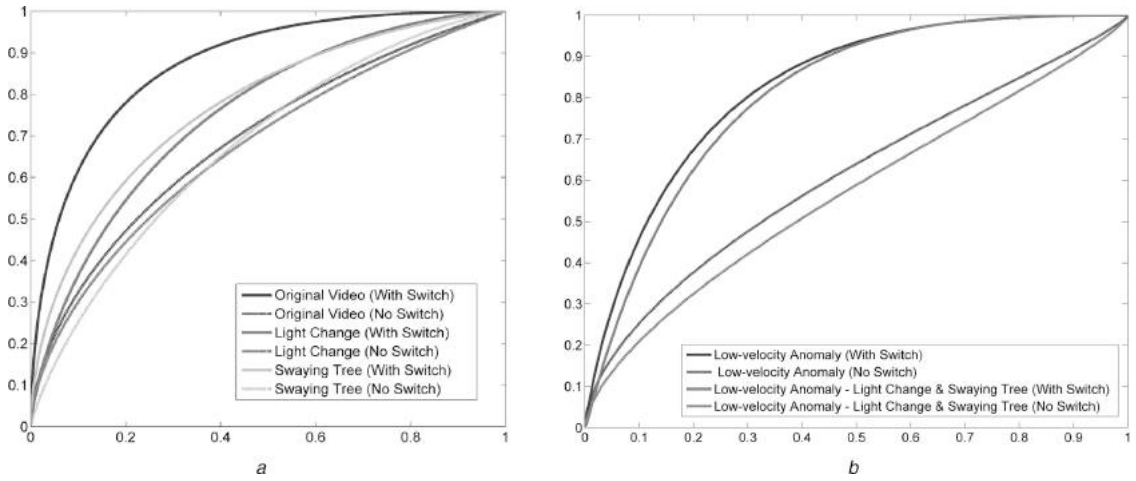

Fig. 7 *ROC curves for*
a High-velocity anomaly scene and its variants
b Low-velocity anomaly scene and its variants

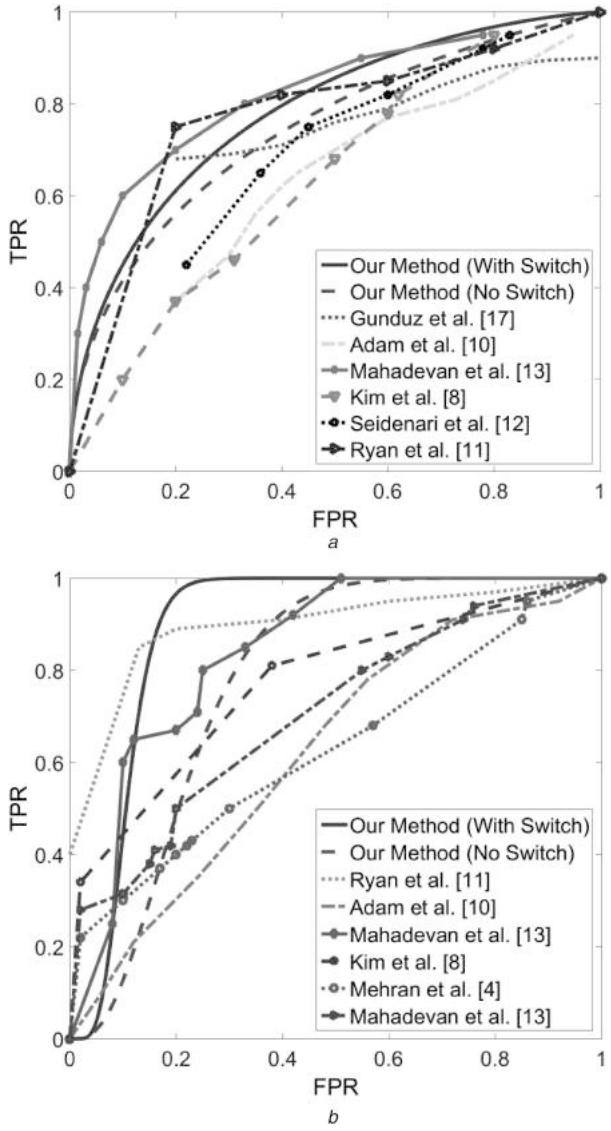

Fig. 8 *ROC curves for*
a Peds1 and
b Peds2

mechanism has better performance and has lower standard deviation meaning that it is more stable.

The coefficient is selected based on the expected nature of the anomaly for the scene. For instance, the expected anomaly might be vehicles which are not allowed in a pedestrianised area or a slow moving pedestrian entering a bicycle path. $0 < c < 1$ means that the expected anomaly has a lower velocity than the general behaviour and $1 < c < 2$ means the expected anomaly has a higher velocity.

5.6 Computational performance and complexity

The performance of the proposed method is compared against those in the literature in Table 3 in terms of frames per second (FPS). Only two of the existing works [11, 13] reported these values (no specifics on their system configurations were reported). For the proposed method, *Peds1* and *Peds2* FPS values are different. Since it searches through the training set for the appropriate model, having a higher number of data points in the training set results in a higher cost. The model switching mechanism is the most computationally costly part and the feature detection and testing parts take less than 10% of the running time. The complexity of the model switching is $O(2N)$ where N is the number of training videos.

6 Conclusion

In this paper, a density aware method for anomaly detection in crowded scenes is proposed. To detect these anomalies a simple thresholding method is not sufficient because the abnormal behaviours vary for different scenes and also changes throughout the videos. The proposed method can adapt to the varying scene characteristics by selecting the appropriate model to use at each frame and can work in real time. The proposed approach does not require any training videos containing anomalies and can be configured to detect low-velocity and high-velocity anomalies. In the future, the proposed anomaly detection method can be extended to process textural anomalies in addition to the motion based ones and can be adapted to other visual crowd surveillance tasks.

7 Acknowledgments

This work was supported by The Scientific and Technological Research Council of Turkey (grant no. 112E141).

Table 1 EER comparison

Method/dataset	*Peds1*	*Peds2*
proposed method	0.29	0.15
Ryan *et al.* [11]	0.23	0.13
MDT	0.25	0.25
Adam *et al.* [10]	0.38	0.42
Kim and Grauman [8]	0.40	0.30
Seidenari *et al.* [12]	0.39	—
Mehran *et al.* [4]	—	0.42

Table 2 AUC value statistics of different coefficients

Dataset	Method w. switch			Method w/o switch		
	Mean AUC	Std. dev.	Best AUC	Mean AUC	Std. dev.	Best AUC
Peds 1	0.703	0.047	0.761	0.682	0.067	0.746
simulation	0.660	0.051	0.757	0.593	0.054	0.669

Table 3 Processing speed in FPS

Method	Dataset	
	Peds1	*Peds2*
proposed method	44.36	94.34
Ryan *et al.* [11]	9.40	9.40
MDT	0.04	0.04

8 References

[1] Rodriguez, M., Laptev, I., Sivic, J., *et al.*: 'Density-aware person detection and tracking in crowds'. ICCV, November 2011, pp. 2423–2430

[2] Fradi, H., Dugelay, J.-L.: 'Towards crowd density-aware video surveillance applications', *Inf. Fusion*, 2015, **24**, pp. 3–15

[3] Yigit, A., Temizel, A.: 'Particle filter based conjoint individual-group tracker (CIGT)'. AVSS, August 2015

[4] Mehran, R., Oyama, A., Shah, M.: 'Abnormal crowd behavior detection using social force model'. CVPR, 2009, pp. 935–942

[5] Helbing, D., Molnar, P.: 'Social force model for pedestrian dynamics', *Phys. Rev. E*, 1995, **51**, (5), p. 4282

[6] Blei, D.M., Ng, A.Y., Jordan, M.I.: 'Latent Dirichlet allocation', *J. Mach. Learn. Res.*, 2003, **3**, pp. 993–1022

[7] Raghavendra, R., Del Bue, A., Cristani, M., *et al.*: 'Abnormal crowd behavior detection by social force optimization'. Human Behavior Understanding, Springer, 2011, pp. 134–145

[8] Kim, J., Grauman, K.: 'Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates'. in CVPR, 2009, pp. 2921–2928

[9] Tipping, M., Bishop, C.: 'Mixtures of probabilistic principal component analyzers', *Neural Comput.*, 1999, **11**, (2), pp. 443–482

[10] Adam, A., Rivlin, E., Shimshoni, I., *et al.*: 'Robust real-time unusual event detection using multiple fixed-location monitors', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, **30**, (3), pp. 555–560

[11] Ryan, D., Denman, S., Fookes, C., *et al.*: 'Textures of optical flow for real-time anomaly detection in crowds'. AVSS, 2011, pp. 230–235

[12] Seidenari, L., Bertini, M., Bimbo, A.D.: 'Dense spatio-temporal features for non-parametric anomaly detection and localization'. Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, ACM Int. Workshop, 2010, pp. 27–32

[13] Mahadevan, V., Li, W., Bhalodia, V., *et al.*: 'Anomaly detection in crowded scenes'. CVPR, 2010, pp. 1975–1981

[14] Lu, C., Shi, J., Jia, J.: 'Abnormal event detection at 150 fps in MATLAB'. ICCV, December 2013, pp. 2720–2727

[15] Leung, V., Colombo, A., Orwell, J., *et al.*: 'Modelling periodic scene elements for visual surveillance', *IET Comput. Vis.*, 2008, **2**, (2), pp. 88–98

[16] Guler, P., Temizel, A., Temizel, T.T.: 'An unsupervised method for anomaly detection from crowd videos'. IEEE Signal Processing, Communication and Applications Conf., April 2013

[17] Gunduz, A.E., Temizel, T.T., Temizel, A.: 'Pedestrian zone anomaly detection by non-parametric temporal modelling'. AVSS, 2014, pp. 131–135

[18] Bae, G., Kwak, S., Byun, H.: 'Motion pattern analysis using partial trajectories for abnormal movement detection in crowded scenes', *Electron. Lett.*, 2013, **49**, (3), pp. 186–187

[19] Cheng, K.-W., Chen, Y.-T., Fang, W.-H.: 'Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression'. CVPR, 2015, pp. 2909–2917

[20] Ullah, H., Conci, N.: 'Crowd motion segmentation and anomaly detection via multi-label optimization'. ICPR, 2012

[21] Ullah, H., Ullah, M., Conci, N.: 'Dominant motion analysis in regular and irregular crowd scenes'. Human Behavior Understanding, Springer, 2014, pp. 62–72

[22] Solmaz, B., Moore, B.E., Shah, M.: 'Identifying behaviors in crowd scenes using stability analysis for dynamical systems', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (10), pp. 2064–2070

[23] Ullah, H., Ullah, M., Conci, N.: 'Real-time anomaly detection in dense crowded scenes'. IS&T/SPIE Electronic Imaging, 2014

[24] Farnebäck, G.: 'Two-frame motion estimation based on polynomial expansion'. Image Analysis, Springer, 2003, pp. 363–370

[25] Rublee, E., Rabaud, V., Konolige, K., *et al.*: 'ORB: An efficient alternative to SIFT or SURF'. ICCV, 2011, pp. 2564–2571

[26] Calonder, M., Lepetit, V., Ozuysal, M., *et al.*: 'BRIEF: Computing a local binary descriptor very fast', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (7), pp. 1281–1298

[27] Chan, A., Liang, Z.-S., Vasconcelos, N.: 'Privacy preserving crowd monitoring: Counting people without people models or tracking'. CVPR, June 2008

[28] Jamshidian, F., Zhu, Y.: 'Scenario simulation: theory and methodology', *Financ. Stoch.*, 1996, **1**, (1), pp. 43–67

[29] METU, 'METUCrowd: Crowd simulation dataset,' Available at ftp://ftp.vrcv.ii.metu.edu.tr/Datasets/METUCrowdSimulationDataset/, December 2014

[30] 'Unity Technologies: Unity 3D v4.0.0,' Available at http://www.unity3d.com/, November 2014

[31] 3DRT, 'Male/female character packs,' Available at http://www.3drt.com/store/characters, December 2014

[32] Bradski, G.: 'The openCV library', *Doctor Dobbs J.*, 2000, **25**, (11), pp. 120–126

[33] Murphy, K.: 'The bayes net toolbox for Matlab', *Comput. Sci. Stat.*, 2001, **33**, (2), pp. 1024–1034