

Anomalous Behavior Detection in Crowded Scenes using Clustering and Spatio-temporal Features

Meng Yang^{1,2}, Sutharshan Rajasegarar³, Aravinda S Rao⁴, Christopher Leckie^{1,2},
Marimuthu Palaniswami⁴

1 Dept. of Computing and Information Systems, The University of Melbourne
VIC 3010, Australia

2 National ICT Australia (NICTA), VIC 3053, Australia
myang3@student.unimelb.edu.au, caleckie@unimelb.edu.au

3 School of Information Technology, Deakin University, VIC 3125, Australia
sutharshan.rajasegarar@deakin.edu.au

4 Dept. of Electrical and Electronic Engineering, The University of Melbourne
VIC 3010, Australia
{aravinda.rao, palani}@unimelb.edu.au

Abstract. Anomalous behavior detection in crowded and unanticipated scenarios is an important problem in real-life applications. Detection of anomalous behaviors such as people standing statically and loitering around a place are the focus of this paper. In order to detect anomalous events and objects, ViBe was used for background modeling and object detection at first. Then, a Kalman filter and Hungarian cost algorithm were implemented for tracking and generating trajectories of people. Next, spatio-temporal features were extracted and represented. Finally, hyperspherical clustering was used for anomaly detection in an unsupervised manner. We investigate three different approaches to extracting and representing spatio-temporal features, and we demonstrate the effectiveness of our proposed feature representation on a standard benchmark dataset and a real-life video surveillance environment.

KEYWORDS: anomaly detection, spatio-temporal features, hyperspherical clustering

1 Introduction

Analysis of human behaviour in crowded environment is an important and challenging task for video surveillance. Significant efforts have been made to solve this task, such as using large numbers of surveillance cameras to monitor human behaviour. However, the ubiquity of cameras still causes issues, such as system overload, manual monitoring and low accuracy. Therefore, an automated system for behaviour detection is required to help improve efficiency and reduce detection errors. We aim to detect anomalous events in a target area monitored by cameras over a period of time. Anomalous events include standing statically, loitering around a place, running

among a crowd of walking people, and the number of people increasing dramatically at the entrance or exit in some stadium, cinema or other venue. These abnormal events can occur suddenly, hence, an automated and online analysis system is needed for detecting anomalous behaviours.

In this paper, we construct a framework for anomalous behaviour detection, such as remaining static or loitering in the flow of a crowd. This method is almost real-time. In particular, we use a hyperspherical clustering method on the encoded trajectories of pedestrians using novel spatio-temporal feature representations. In other words, after obtaining the tracks of objects and representing the spatial and temporal relationship of these objects, those objects that show behaviour like remaining static or loitering will be detected and declared as anomalous. The main contributions of this work are: (1) our approach to performing object detection and tracking for generating trajectories of objects; (2) we propose the three kinds of spatio-temporal encodings for feature representation, including two novel encoding schemes for anomalous behaviour detection; (3) we use a hyperspherical cluster based distributed anomaly detection method [1] to effectively identify anomalous trajectories in the data; (4) we perform an evaluation on benchmark and real data sets including videos collected from a stadium in Australia; (5) our method is completely unsupervised, hence no labelled data nor supervised training is required.

The datasets used in this work are Melbourne Cricket Ground (MCG) and Performance Evaluation of Tracking and Surveillance (PETS) 2009 videos. The MCG dataset was collected at the Melbourne Cricket Ground with six cameras named C1 to C6. Five of them were installed in a corridor and C1 was placed over a seating area. The total data are 31.05 hours video files. PETS2009 is a popular dataset with multi-sensor sequences used for crowd activity recognition [2]. We assume that the video data from cameras are directly available and the cameras are calibrated.

2 Related Work

In order to address the abnormal event detection problem, many algorithms have been proposed in the literature. The methods for anomalous behavior detection can be categorized into two types. One is trajectory analysis and the other is motion representation. Trajectory analysis [3] comprises tracking and distinguishing objects or crowds in the scenes. Motion representation methods analyse patterns such as texture and dynamic models. Optical flow methods are quite popular, for example, Kim and Grauman [4] built a model of optical flow patterns with a mixture of probabilistic Principle Component Analysis (PCA) models, then used a Markov Random Field (MRF) for global consistency guarantees. Mehran [5] learned from crowd behavior studies in [6], and used social force and some other concepts to depict crowd behavior. Then, the concepts and optical flow methods are combined with a latent Dirichlet allocation (LDA) model, and used for anomaly detection. Andrade [7] extracted an optical flow field and used component analysis for reducing dimensionality, and then trained a Hidden Markov Model (HMM) for classifying normal and abnormal behaviors.

Several methods are proposed for anomalous data detection [8]. In particular, supervised and semi-supervised schemes [9] are popular in this category. In terms of unsupervised methods, it is hard to obtain a large amount of pre-labelled data from complicated scenes and predict some unanticipated anomalous data. Recently, in [10], a clustering based anomaly detection method was proposed, which can detect new anomalous data in an unsupervised way.

Since a simple tracking algorithm cannot handle anomalous behavior detection in crowded scenes, we propose the use of tracking analysis with clustering analysis for anomaly detection. In this work, ViBe [11] based foreground subtraction, Kalman filtering and Hungarian cost algorithm tracking are first used to obtain trajectories of objects. Next, we utilize a fixed-width clustering algorithm, which is an efficient hyperspherical clustering method for abnormal behaviour detection in crowded scenarios. Figure 1 demonstrates our proposed framework for anomalous behaviour detection. The process is divided into six steps: (a) video (camera) inputs, (b) image preprocessing and object detection, (c) object tracking, (d) feature extraction and feature representation, and (e) hyperspherical clustering based anomaly detection. The main challenge we address is how to find a suitable spatio-temporal feature representation in this content.

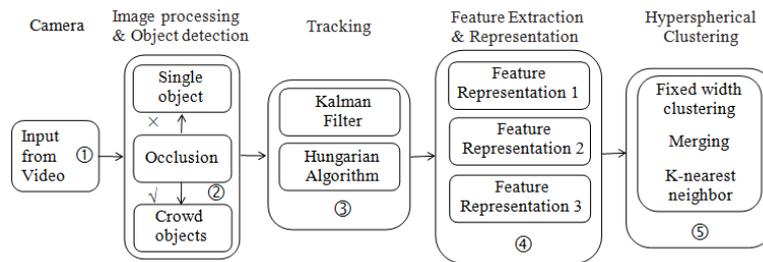


Fig. 1. Overview of our framework for unsupervised anomalous behavior detection

3 The Proposed Approach

Human behaviour has no fixed predefined features, so it is challenging to depict anomalous behaviour directly. Our proposed scheme uses unsupervised clustering to characterize normal behaviour and detect anomalous behaviour clusters. We denote the objects in the scene to be $S = \{s_i : i = 1 \dots s\}$. For each frame Δ_f , every object s_i has its position in the frame. For several frames (a video sequence), a feature representation consisting of spatio-temporal information is extracted and encoded by three types of schemes, which can depict the trajectory of each object. Further, every object s_i measures a spatio-temporal feature vector $X_i = \{x_k^i : k = 1 \dots m\}$ and all of the feature vectors build the spatio-temporal feature matrix $X = \cup_{i=1 \dots n} X_i$. Finally, X is the input to a hyperspherical based clustering algorithm [1] for outlier detection, and the detected outliers are regarded as the anomalous behaviours.

In this work, at first, video preprocessing is performed followed by object detection, which is based on ViBe background subtraction. Next, the objects are tracked by a Kalman filter and Hungarian algorithm. We then propose three alternative methods for feature representation so that clustering can then be applied to the trajectories. Finally, feature vectors are categorized into normal and anomalous clusters by using fixed width clustering. Overall performance of the framework depends on the object detection, tracking, feature representation and clustering methods.

3.1 Image Preprocessing and Object Detection

The video frames are converted to grayscale images at first. Then, these grayscale images are filtered by a 2D Gaussian low-pass filter in order to filter the high-frequency noise. The parameter of the Gaussian low-pass filter is set to be $\sigma = 0.5$ and the block size is 5×5 . The choice of filter parameters is based on maintaining a part of the edge information while at the same time keeping low-frequency information.

The object detection algorithms can be categorized into three groups: (1) frame difference method, (2) optical flow based approach, (3) background subtraction. Especially, background subtraction is a crucial category of object detection method and several techniques have been proposed in the literature. These methods can be divided into parametric methods and sample-based methods. The former methods are based on the location of each pixel and the latter one is based on aggregating previously observed values for each pixel location. Considering the complexity of the scenarios and the sensitivity of the environment, in this work, we choose a sample-based algorithm ViBe [11] for background modeling and subtraction. The algorithm begins with defining a pixel model by using a set of sample values that can be used for background estimation, then updating the model regularly. The objects can be detected by subtracting the pixel model from incoming frames.

3.2 Tracking

Kalman filtering [12] is used for tracking detected objects by the previous step. The process of Kalman filtering is: Initialization->Prediction->Correction. The motion equation and model are displayed below:

$$x_{k+1} = F_{k+1,k}x_k + w_k \quad (1)$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \Delta x_{k+1} \\ \Delta y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \Delta x_k \\ \Delta y_k \end{bmatrix} + w_k \quad (2)$$

and the measurement equation and model are given below

$$y_k = H_k x_k + v_k \quad (3)$$

$$\begin{bmatrix} xm_k \\ ym_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \Delta x_k \\ \Delta y_k \end{bmatrix} + v_k \quad (4)$$

where, $F_{k+1,k}$ is the transition matrix and w_k is additive process noise; H_k is the measurement matrix and v_k is the measurement noise; and xm_k are ym_k the observed position. Because of the crowded environment, the tracking algorithm we use should handle multi-object association and assignment. Therefore, we use the Hungarian algorithm [13] for multiple objects tracking.

3.3 Feature extraction and representation

For the video analytics, anomalous crowds can be detected at three levels: temporal, spatial and spatio-temporal, which is a combination of the spatial and temporal levels. Crowded scenes contain a large number of objects with different behaviours within one frame, and these behaviours occur and change within the image sequence. In this case, anomalous crowd behaviour should be analysed on both the spatial and temporal levels. Therefore, it would be better to choose a spatio-temporal feature representation as our encoding scheme in this scenario.

After the feature extraction and representation step, an appropriate feature representation is needed for the spatio-temporal trajectories, which can have arbitrary length. This feature representation becomes the input of clustering in order to identify normal and abnormal behaviour. In this work, we use three types of feature representation schemes for crowd anomaly detection. The three schemes are all spatio-temporal representations, including one representation which is modified from the coding scheme in [14], and two other novel feature representations. The details of the feature and representation methods are given below.

Let m be the width and n be the height of the video frame. The input frame is divided into $b_x \times b_y$ blocks, where $1 < b_x \leq m$, $1 < b_y \leq n$. Block size selection affects the encoding results, which will be discussed in the next section. Next, a feature representation with spatio-temporal information is extracted for each object. The block value starts from 1 for the original position of each object. In the following frames, the object will either stay in the same block or move to another block. The following three encoding schemes are used to represent the status of each object.

1) Feature Representation 1 (FR1): If the object enters into another block, then this new block value will be increased by one. If the object stays in the same block, the value of that block will not be changed. Further, if this object enters into the same block again after several steps, the new value will replace the old one.

2) Feature Representation 2 (FR2): The value of each block depends on the length of time the object stays in that block. If the object stays in one block for 10 frames and moves to another block for 1 frame, then the feature value of these two blocks are 10 and 1, respectively.

3) Feature Representation 3 (FR3): This is a combination of the former two types. At first, an object enters into a block and this block will be assigned the value one,

then if this object stays in the same block, the value will be the number of frames that the object stays in that block. Further, if the object moves to another block, the value of this new block will start from the value of the former block plus one.

Once we have encoded the frames and the blocks as mentioned above, we will have a collection of trajectories for the objects in the video over the period of time considered for analysis. The next step is to identify trajectories that are normal or anomalous. Next we describe a hyperspherical clustering scheme to perform this in an unsupervised way.

3.4 Clustering based anomaly detection

In this work, the clustering algorithm we used is from [1]. The method has three main steps: (1) Fixed width clustering, which is a hyperspherical clustering with fixed radius. The process is as follows. The first cluster is created with a fixed width (radius), centred at the first data point. Then the distance between the next new data vector and its closest cluster centre is calculated. If the distance is less than the radius, then the data vector is added to this cluster and the centroid of that cluster is recalculated as the mean of all data vectors in it. If the distance is more than the radius, then the data vector will form a new cluster using it as the centroid. The process is continued until all the data vectors are considered. (2) Merging, When the distance between the center of two clusters is less than a threshold τ , then they will be merged into one new cluster. (3) The anomalous clusters are identified by using the K nearest neighbor (K-NN) approach [15]. This method helps cluster similar behaviors (trajectories) and finds the anomalous events in the video in an unsupervised manner. The parameters of this algorithm are ω (cluster width) and ψ (the number of standard deviations of the inter-cluster distances used for identifying the anomalous clusters). The number of clusters yielded is based on ω , and ψ determines the sensitivity of anomaly detection. The detailed steps of this method and the parameters can be found in [1].

4 Results and Discussion

In this section, we discuss the dataset used, and then show the anomaly detection results based on the three types of encoding schemes proposed. The details of the data we used are listed in Table 1. In the data we used for evaluation, people who stand statically or loiter in the scene for a while are regarded as anomalies. We manually annotated the anomalous objects as the ground truth. The computer vision part is implemented using OpenCV 3.0 with Visual Studio 2013, and the anomaly detection part is implemented in Java. The computer used is Windows 7 (64 bit) consisting of an Intel® i7 - 4790 CPU running at 3.6GHz with 16 GB RAM. The computer also includes a 4 GB NVIDIA NVS 315 HD 4600 graphics card.

| MCG Dataset | | | PETS2009 Dataset | | | |
|-------------|------------|-------|------------------|-------|------|--------|
| Date | Camera | TL | Camera | TS | View | Frames |
| 23-Sep-2011 | C2-C6 | 22:01 | S2-L1 | 12-34 | 001 | 794 |
| 24-Sep-2011 | C2-C6 | 14:01 | S2-L2 | 14-55 | 001 | 435 |
| 24-Sep-2011 | C6_cut1-12 | 00:20 | S2-L3 | 14-41 | 001 | 239 |

※TL= Time Length (mm:ss) ※TS=Timestamp

Table 1. Details of the data used, including MCG and PETS2009 dataset.

4.1 Object detection performance

Loitering and static objects are detected based on using the three types of spatio-temporal feature representations (FR1, FR2, FR3). Loitering objects are those who walk around and come back to the same place. Static objects are those who stand statically or move slightly for a while in the scene. The number of identified objects was compared to the ground truth generated by annotating the original video manually. For these three feature representation schemes, they have similar feature matrices. The maximum value of the row vector increased with the occurrence of anomalous objects. The results are tabulated in Table 2 (MCG dataset) and Table 3 (PETS2009 dataset) respectively. For the hyperspherical clustering part, the parameters were set as: $\omega = 5$, $\tau = \frac{1}{2}\omega$, K-NN=3 and $\psi = 1$.

| | | | | | | | |
|---------------|------------|--------------|--------------|--------------|------------|--------------|--------------|
| GT=206 | 8×8 | 16×16 | 32×32 | GT=93 | 8×8 | 16×16 | 32×32 |
| FR1 | 84 | 150 | 108 | FR1 | 23 | 44 | 58 |
| FR2 | 193 | 196 | 209 | FR2 | 74 | 74 | 81 |
| FR3 | 167 | 227 | 181 | FR3 | 62 | 73 | 65 |
| GT=118 | 8×8 | 16×16 | 32×32 | GT=66 | 8×8 | 16×16 | 32×32 |
| FR1 | 48 | 76 | 96 | FR1 | 16 | 50 | 63 |
| FR2 | 96 | 140 | 125 | FR2 | 54 | 61 | 56 |
| FR3 | 98 | 128 | 106 | FR3 | 51 | 58 | 58 |

※ GT=Ground Truth

Table 2. The table lists the number of detected anomalous objects using the three types of spatio-temporal feature encoding schemes on the MCG dataset (24-Sep-2011). Left-top section is for C2, right-top section is for C3, left-bottom is for C5 and right-bottom is for C6.

| | | | | | | | | | | | |
|-------------|------------|--------------|--------------|-------------|------------|--------------|--------------|-------------|------------|--------------|--------------|
| GT:8 | 8×8 | 16×16 | 32×32 | GT:6 | 8×8 | 16×16 | 32×32 | GT:3 | 8×8 | 16×16 | 32×32 |
| FR1 | 6 | 4 | 2 | FR1 | 5 | 5 | 3 | FR1 | 2 | 2 | 2 |
| FR2 | 9 | 5 | 3 | FR2 | 8 | 4 | 7 | FR2 | 2 | 2 | 1 |
| FR3 | 7 | 6 | 2 | FR3 | 6 | 5 | 4 | FR3 | 3 | 4 | 2 |

Table 3. The table lists the number of detected anomalous objects using the three types of spatio-temporal features encoding schemes on the PETS2009 dataset.

For the MCG dataset, the anomalies detected by the three types of spatio-temporal feature representation have been tabulated in Table 2. Videos from camera C2, C3,

C5 and C6 (24-September-2011) were used, which are all of length 14 minutes and 1 second. These frames were divided into 8×8 , 16×16 and 32×32 blocks. The ground truth was generated by annotating the video files. From Table 2, we find that the number of anomalies is affected by the block size selection. For example, comparing the 8×8 and 16×16 blocks, the number of anomalies detected by the former one is larger than the latter one. This means that the larger block size results in lower resolution coding, so the number of detected anomalous objects are less. However, this is not correct for all cases.

Further, the number of detected anomalies based on FR1 is less than for FR2 and FR3. This can be explained by using an example. If there is an object standing statically in block 2 for 100 frames and then walks to block 3, then the value of block 2 should be 1 and block 3 should be 2 by using FR1. Next, the value of block 2 should be 100 and block 3 should be 1 based on FR2, and the two blocks will be 100 and 101 based on FR3. It is clear that the object will be regarded as abnormal by using FR2 and FR3, whereas normal in FR1. The large number of static objects in the MCG dataset causes the lower number of detected anomalies based on FR1. In other words, FR2 and FR3 are more suitable for detecting static objects.

For the PETS2009 dataset, from Table 3, it can be seen that the number of anomalies detected based on FR1 is similar with FR2 and FR3 in Table 3. This is because the anomalous behavior in the PETS2009 dataset is different from the MCG dataset. The anomalous objects loiter on a large scale in PETS2009, whereas objects stand statically or move only slightly in the MCG. We can assume that the three types of feature representation methods have similar effects on detecting loitering objects.

From Table 2 and Table 3, we find that for some instances the number of detected anomalies is more than the ground truth. There are two possibilities that can cause this situation. One is that normal objects are regarded as anomalies, and the other is because of the algorithm we used for extracting trajectories. The object detection and tracking algorithms we used are not based on a pre-trained model, which means we cannot always obtain a complete and smooth trajectory of one object. The trajectory of the object can be split into 2 or 3 parts, which causes the situation that one object is detected as 2 or 3 abnormal objects.

4.2 Accuracy analysis

The MCG dataset was used for our accuracy evaluation, including C2-C6 (23-Sep-2011), C2-C6 (24-Sep-2011) and 12 cuts of video of C6 (24-Sep-2011). In the last section, we obtained the number of detected anomalous objects, while we also obtained the object ID, which can denote the real behavior of his/her (normal or abnormal) movement pattern. We choose 8×8 as the block size for this evaluation part. Finally, we can obtain the detection accuracy of the detected anomalous objects. The results have been tabulated in Table 4.

| Dataset Accuracy | 23-Sep-2011 | | | | 24-Sep-2011 | | | | SD |
|---------------------|-------------|-------|-------|-------|-------------|-------|-------|-------|-------------|
| | C2 | C3 | C5 | C6 | C2 | C3 | C5 | C6 | |
| FR1 | 40.3% | 31.4% | 39.0% | 22.7% | 48.5% | 41.6% | 44.6% | 44.8% | 39.1 ± 7.8% |
| FR2 | 78.6% | 75.3% | 76.3% | 74.2% | 75.8% | 74.2% | 77.7% | 79.2% | 76.4 ± 1.8% |
| FR3 | 72.3% | 65.6% | 78.0% | 71.2% | 74.3% | 69.9% | 76.2% | 78.4% | 73.2 ± 4.1% |

Table 4. (a)

| Dataset | C1 | C2 | C3 | C4 | C5 | C6 |
|---------|-------|------|-------|-------|-------|-----|
| FR1 | 66.7% | 60% | 57.1% | 50% | 44.4% | 80% |
| FR2 | 83.3% | 100% | 71.4% | 87.5% | 77.8% | 80% |
| FR3 | 100% | 80% | 71.4% | 87.5% | 88.9% | 80% |

| Dataset | C7 | C8 | C9 | C10 | C11 | C12 | SD |
|---------|-------|-------|-----|------|------|-------|--------------|
| FR1 | 75% | 66.7% | 40% | 75% | 60% | 66.7% | 63.5 ± 9.9% |
| FR2 | 87.5% | 77.8% | 80% | 100% | 80% | 100% | 85.4 ± 9.4% |
| FR3 | 87.5% | 77.8% | 60% | 100% | 100% | 100% | 86.1 ± 12.3% |

※SD = Standard deviation

Table 4. (b)

Table 4. The detection accuracy using the three types of spatio-temporal features encoding schemes. (a) C2-C6 (23-Sep-2011) and C2-C6 (24-Sep-2011), (b) 12 video cuts of C6 (24-Sep-2011)

From Table 4(a), it is clear that the accuracy of FR1 is lower than FR2 and FR3, which is similar to the result of Table 2. FR2 and FR3 have similar accuracy. From Table 4(b), although the accuracy of FR1 is still lower than FR2 and FR3, the accuracy of FR1 is increased. This is because the time length of the video cut is short, so the time length of loitering is short. The feature matrix of these three feature representation methods has no major differences. In terms of the false positive rate, it is around 10% to 20% for FR1, FR2 and FR3. The effect is quite similar among the three encoding schemes. Generally speaking, FR2 and FR3 are more suitable for detecting static objects, especially in a crowded scene. The three types of feature representation have similar effects on detecting loitering objects.

5 Conclusion

Anomalous behavior detection in crowded and unanticipated scenarios is an important problem in real-life applications. In this work, the anomalous behaviors of standing statically and loitering in a video were detected by using two novel encoding schemes for spatio-temporal features. At first, ViBe was used for object detection. Then, Kalman filtering and a Hungarian cost algorithm were implemented for multi-object tracking. Next, the spatio-temporal features were extracted and represented by three

types of schemes. In the end, a hyperspherical clustering based algorithm was used for anomaly detection. The evaluation reveals that our proposed unsupervised anomaly detection scheme using our novel spatio-temporal features is capable of detecting anomalous events such as loitering and stationary objects with high accuracy on a real life and a benchmark dataset.

Acknowledgements

This work was supported by National ICT Australia (NICTA).

References

1. Rajasegarar, S., Leckie, C., Palaniswami, M., "Hyperspherical cluster based distributed anomaly detection in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 1833-1847 (2014)
2. Ferryman, J., Crowley, J., Shahrokni, A.: PETS 2009 Benchmark data, <http://www.cvg.rdg.ac.uk/PETS2009/a.html> (2009)
3. Stauffer, C., Grimson, W. E. L., "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 747-757 (2000)
4. Kim, J., Grauman, K.: "Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates," *IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*, pp. 2921-2928 (2009)
5. Mehran, R., Oyama, A., Shah, M.: "Abnormal crowd behavior detection using social force model," *IEEE Conference on Computer Vision and Pattern Recognition, 2009 (CVPR 2009)*, pp. 935-942 (2009)
6. Helbing, D., Molnar, P., "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, p. 4282 (1995)
7. Andrade, E. L., Blunsden, S., Fisher, R. B.: "Modelling crowd scenes for event detection," *18th International Conference on Pattern Recognition, 2006 (ICPR 2006)*, pp. 175-178 (2006)
8. Rajasegarar, S., Leckie, C., Palaniswami, M., "Anomaly detection in wireless sensor networks," *IEEE, Wireless Communications*, vol. 15, pp. 34-40 (2008)
9. Chandola, V., Banerjee, A., Kumar, V., "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, p. 15 (2009)
10. Rajasegarar, S., Leckie, C., Palaniswami, M., Bezdek, J. C.: "Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks," in *IEEE ICC*, pp. 3864-3869 (2007)
11. Barnich, O., Van Droogenbroeck, M.: "ViBe: a powerful random technique to estimate the background in video sequences," *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)*, pp. 945-948 (2009)
12. Cuevas, E. V., Zaldivar, D., Rojas, R., "Kalman filter for vision tracking," (2005)
13. Kuhn, H. W., "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83-97 (1955)
14. Rao, A. S., Gubbi, J., Rajasegarar, S., Marusic, S., Palaniswami, M.: "Detection of anomalous crowd behaviour using hyperspherical clustering," *2014 International Conference on, Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1-8 (2014)
15. Ramaswamy, S., Rastogi, R., Shim, K.: "Efficient algorithms for mining outliers from large data sets," in *ACM SIGMOD Record*, pp. 427-438 (2000)