International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 4, August 2019, pp. 2403~2415
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i4.pp2403-2415

Autonomous abnormal behaviour detection using trajectory analysis

Muhammed Shuaau, Ka Fei Thang, Nai Shyan Lai
School of Engineering, Faculty of Computing, Engineering & Technology,
Asia Pacific University of Technology & Innovation, Malaysia

Article Info

Article history:

Received Jun 16, 2018
Revised Nov 21, 2018
Accepted Mar 4, 2019

Keywords:

Abnormal behaviour
Anomaly detection
Security
Trajectory analysis
Visual surveillance

ABSTRACT

Abnormal behaviour detection has attracted signification amount of attention in the past decade due to increased security concerns around the world. The amount of data from surveillance cameras have exceeded human capacity and there is a greater need for anomaly detection systems for crime monitoring. This paper proposes a solution to this problem in a reception area context by using trajectory extraction through Gaussian Mixture Models and Kalman Filter for data association. Here, trajectory analysis was performed on extracted trajectories to detect four different anomalies such as entering staff area, running, loitering and squatting down. The developed anomaly detection algorithms were tested on videos captured at Asia Pacific University's reception area. These algorithms were able to achieve a promising detection accuracy of 89% and a false positive rate of 4.52%.

Corresponding Author:

Ka Fei Thang,
School of Engineering,
Asia Pacific University of Technology & Innovation,
Technology Park Malaysia, Bukit Jalil, Kuala Lumpur, 57000, Malaysia.
Email: dr.ka.fei@apu.edu.my

1. INTRODUCTION

Abnormal behaviour detection is one of the most important research area in computer vision. It is a very challenging and diverse area that has attracted a signification amount of attention in the past decade. Authorities and corporations very often rely on surveillance video feeds to monitor public places and other common areas such as reception areas. However, the amount of data from surveillance cameras have exceeded the capacity of human operators. Human operators are often sloppy, suffer from fatigue and get distracted easily. Hence human operators are unable to effectively monitor the video feeds and could result in dangerous occurrences being neglected. The solution to this problem is to use an autonomous anomaly detection in surveillance videos to automatically detect when a suspicious event has occurred based on the context.

Many different approaches to autonomous anomaly detection have been used by researchers in recent years. In [1], suspicious behaviour detection was performed by utilising contextual information. This system consist of a context space model that provides context sensitive information which was represented by the behaviour class and frequency of its occurrence. Then a data stream clustering algorithm was used to update the behaviour model efficiently from the video feed with limited resources and time. Finally, an inference algorithm was used to classify the behaviour by using the information from current context and the previously learned context to make an inference about an observed behaviour. In [2] the researchers proposed an unsupervised anomaly detection system using feature clustering. Gaussian Mixture Model (GMM) based foreground detection was used with adaptive region updating in which the input frame was divided in to non-overlapping N×N blocks and gradient similarity between the background and input frame was calculated. Multiple object tracking was then performed and object features were extracted followed by scene analysis to classify the event. Anomaly detection without prior knowledge about the environment was made possible by extracting patterns through feature clustering and matching the trajectory to cluster by comparing with a predefined threshold under Gaussian distribution to detect abnormal part of the trajectory. The algorithm was able to achieve good results.

In [3], abnormal behaviour detection was based on trajectory Sparse Reconstruction Analysis (SRA). Trajectories extracted from object tracking of normal behaviours were collected and categorised in to different Route sets and sampled with Least-squares Cubic Spline Curves Approximation (LCSCA). Test trajectories were also represented with LCSCA features and trajectories were classified using SRA on the dictionary dataset used. In [4] a method was proposed for loitering detection which is an abnormal behaviour in many contexts. The method was based on Trajectory Direction History Analysis (TDHA) and Inverse Perspective Mapping which was used to resolve distortion of trajectory direction due to perspective effect. In TDHA, direction between two vectors were calculated for direction history and angle between them was calculated to analyse direction variations between vectors. In [5], a covariance feature descriptor over the whole video frame using Horn-Schunck optical flow computation algorithm was used to encode moving information and one-class support vector machine algorithm was used to classify abnormal events.

In [6], abnormal detection algorithm was proposed based on an image descriptor and a non-linear classification method. Histogram of optical flow orientation was used to encode moving information of every frame and one-class support vector machine for classification. Then the researchers used a state transition model to reduce false detections due to short abnormal events which occur very rarely in small number of frames in the long sequence. The state transition model changed short abnormal events to normal state and vice versa and it was found to be very effective. In [7] anomaly detection based on a hierarchical activity-pattern discovery framework was proposed. In the offline training phase, normal videos were input and images were split in to fixed size cells to get low level visual features from the cells. Then analysis was carried out to find different normal activity patterns present in the training videos. Then in the test phase, a unified energy function was designed to calculate anomaly energy of each cell in the test frame. Finally, a combination of energy value and spatial-temporal relationship of cells were used to find abnormal regions present. In [8], a trajectory based sparse reconstruction framework was used for video anomaly detection involving multiple objects. The linear sparsity model was kernelized to enable superior class separability. This led to an improved detection rate.

In [9], a loitering an algorithm was proposed to detect loitering. Trajectory extraction was performed and loitering detection was performed by analyzing the trajectory through calculated angles between vectors on the trajectory and a fixed point. Then trajectory is considered loitering if the trajectory duration is more than a fixed time or the variance of the difference between the angles is more than a fixed constant. In [10], an anomaly detection system was proposed using object tracking and classifying activities based on semantics-based approach. The researchers detected suspicious activities such as loitering, stolen luggage, abandoned objects, etc. In [11], histogram of optical flow orientations was used to encode moving information and one class support vector machine or kernel principal component analysis method was used for classification of abnormal activities.

In [12], the researchers highlighted that the presence of a passive, standing crowd is an indication an abnormal event could occur. The methodology involved identifying still crowd by using edges and colour variations dominated by skin colour within the crowd. When the crowd was detected for a certain number of frames, the incident was analysed for abnormal behaviour. In [13], anomaly detection was based on short local trajectories of foreground super-pixels. In [14], an online framework for video anomaly detection was proposed with compact set of highly descriptive features extracted from a novel cell structure. A cell structure was constructed for the entire scene to define spatio-temporal regions to be analysed and compact set of features were extracted. The compact features were then analysed to construct various models and finally an inference mechanism that uses local spatio temporal neighbourhood of cells were used to distinguish abnormal actions.

In [15], a real-time moving object action recognition system was proposed based on motion analysis. The system was implemented on a PixelStreams-based FPGA. The moving object detection was performed by the delta-frame method which determines the absolute difference between two successive images. This method was used because of its ability to adapt to changes in light intensity variations. In [16], a hardware model to measure motion estimation was proposed using bit plane matching algorithm. The algorithm calculated the true motion between video frames for a block and removed temporal redundancies between video frames. Also, it tracked the motion of features in video sequences.

In this paper, a rule-based anomaly detection system in a reception area context is proposed. The advantage of such a system is that the large amounts of labelled training data required with machine learning approaches are not needed and the system is more reliable. The anomalies that are detected with the system are running, entering the staff area, loitering and sudden squat down. Sudden squat down is considered as an anomaly because if a person suddenly squats down, it could mean that there was an aggressive action from somebody such as shooting or throwing things. The block diagram of the proposed system architecture is shown in Figure 1.

The rest of the paper is organised as follows. Section 2 presents the video processing algorithm. Multiple object tracking algorithms and anomaly detection algorithms are presented in sections 3-7. In section 8, results of testing the algorithms along with a discussion are presented. The paper is finally concluded in section 9.

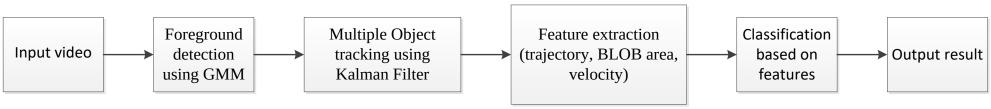

Figure 1. Proposed system block diagram

2. VIDEO PROCESSING ALGORITHM

Figure 2 shows the flowchart of video processing algorithm. The algorithm is run until all the frames of the video file are processed. Object tracking is performed first, which reads the frame from the video file and detects moving objects. The detected moving objects are then associated to tracks which store the trajectory history and many other details about the moving object. The tracking method is explained in the next section.

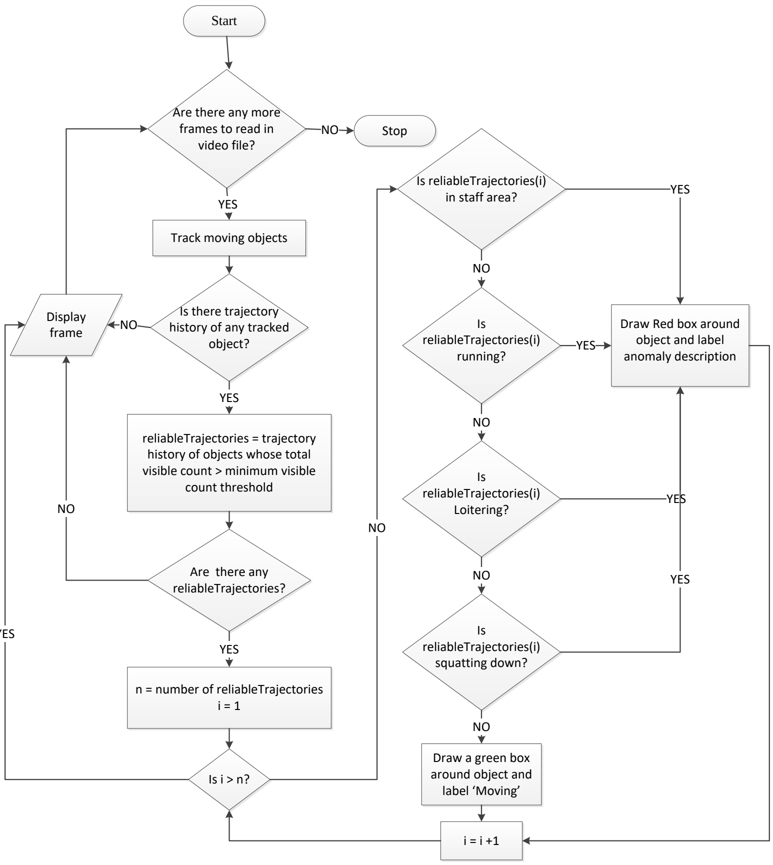

Figure 2. Video processing algorithm

After calling object tracking method, the algorithm checks to see if the method returned any tracks of a moving object which contains the trajectory history. If no tracks are returned, the algorithm continues by displaying the frame and moves on to processing next frame. If tracks are available then the algorithm checks to see if the track is a reliable track. Reliable tracks are those tracks whose total visible count is more than a set threshold. This is done to reduce false detections of noise as moving objects due to small illumination changes. The threshold has to be determined experimentally and it depends on the frame rate of the video input used. If the threshold is too high, actual moving objects might not be detected and if the value is too low, a lot of noise will be detected. The threshold used in the program was 8 and the frame rate of the video used was 9.8 frames per second. An object has to be moving for at least 8 frames before its trajectory history will be analysed.

When reliable tracks of moving objects are available, their trajectory history is analysed. The trajectories are analysed in a loop so that the algorithm can detect multiple people and check their trajectory histories for anomalies. They are checked to see if the person is in the staff area, running, loitering or squatting down. If an anomaly is detected in the trajectory analysis, the trajectory is classified as abnormal and it is highlighted with a red bounding box and labelled with a description of the anomaly. When one anomaly is detected in a trajectory, the same trajectory is not checked for other anomalies because that person will already be classified as abnormal.

If no anomalies are detected in the trajectory analysis, then a green bounding box is drawn around the person and is labelled as "Moving". Once all the trajectories are analysed and moving people are classified and highlighted, the frame is displayed and the algorithm continues processing the remaining frames until all the frames are processed. The methods used to detect the anomalies are explained in the following sections.

3. MULTIPLE OBJECT TRACKING ALGORITHM

Figure 3 shows the multiple object tracking method proposed. The frame is read from video file and foreground mask is obtained by using GMM and then morphological opening and closing with rectangular structuring elements are done to remove noise. Then blob analysis is performed to detect the moving objects. The BLOB analysis returns the bounding boxes of the moving objects and their area in pixels. The BLOB area is then used to further reduce detection noise.

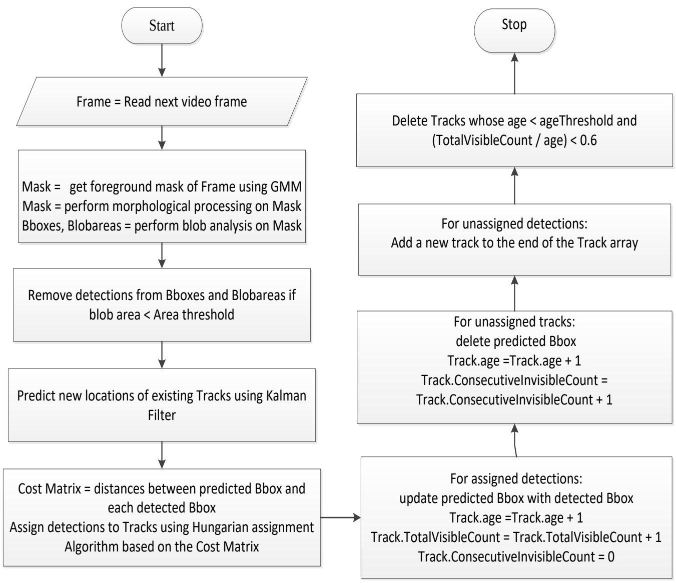

Figure 3. Multiple object tracking method

Some detections are removed if the BLOB area of the detection is less than a set threshold. This step is to reduce noise. The thresholds are set based on the distance between the object and camera. Figure 4 shows the three regions in the reception area to illustrate this approach.

The BLOB area for a person walking in region 3 is much more than other regions because this region is closer to camera position. The minimum BLOB area threshold for this region is higher and if a detection area is smaller than the threshold, that detection is deleted. This is because very small detections in this region is noise due to illumination changes. Similarly, region 2 and region 1 is processed based on BLOB area. The same person walking in region 1 has a much smaller BLOB area compared to walking in region 3. The region minimum area thresholds shoud be determined experimentally

Figure 4. Multiple object tracking method

After deleting the detections based on area, the next step is to assign detections to tracks and add the bounding box coordinates to trajectory history of the track. In order to do this, Kalman Filter is first used to predict the next location of the existing tracks which are already assigned. If there are no existing tracks then the algorithm creates new track for each detection and in the following calls to this function to process the remaining frames of the video file, it predicts the next location using Kalman Filter if motion is detected.

After predicting the next locations, cost matrix is calculated. Cost matrix is an M by N matrix that contains the Euclidian distances between each detection and predicted location of every existing track where M represents the number of tracks and N is the number of detections. Each value in the matrix represents the cost of assigning the N^{th} detection the M^{th} track. After calculating the cost matrix, James Munkres's variant of the Hungarian assignment algorithm is used to determine which tracks are missing and which detections should begin new tracks. The algorithm is also supplied with a scalar value which is the cost of non-assignment. This value represents the cost of a track or detection remaining unassigned. This value was also determined experimentally and 20 is the value used in the implementation. The assignment algorithm returns the indices of the tracks which are assigned and unassigned. It also returns the indices of unassigned detections.

For assigned detections returned from assignment algorithm, the predicted bounding box is replaced with the actual detected bounding box. Then track's age and total visible count are increased. Consecutive invisible count of the track is set to 0. For unassigned tracks the predicted bounding box is deleted from trajectory history because trajectory analysis should only be performed on actual detections and not predictions. Then the track's other properties are set accordingly.

For unassigned detections, a new track is added and stored in the tracks array. After that tracks are deleted if the track's age is less than age threshold and (total visible count/age) is less than 0.6. The above condition will become true if a track is lost for some frames which could mean that the person stopped moving or if a noise is detected and only appears for a very short time. If a person stopped moving and the track of that person is deleted, when the person starts moving again a new track will be created. The information stored in each track are track ID, bounding box history, BLOB area history, age, total visible count and consecutive invisible count.

The track ID of 1 is assigned to first track and then it is incremented for each of the following tracks. A new bounding box is added to the end of bounding box array of the track every time it is detected and this forms the bounding box history which is also the trajectory history. Centroids are the middle point of the bounding box. BLOB area history is also saved in a similar way to bounding box history. These two properties are later analysed for anomaly detection. The BLOB area history is only used in the detection of squatting down anomaly together with trajectory history. The age, total visible count and consecutive invisible count properties are used to manage tracks and to determine reliable tracks. It is also used to remove noise as explained before.

4. ALGORITHM TO DETECT ENTERING STAFF AREA

The algorithm used to detect when a person enters the staff area is shown in Figure 5. The object's last bounding box is used to get the last centroid which gives the object's current location. This centroid's x and y coordinates are checked to see if it is inside the reception desk staff area which can be bounded by a rectangle. If the centroid is within the reception staff area rectangle, the object's trajectory history is further analysed to see if the object came from outside the reception staff area. This is done to avoid classifying as abnormal when the receptionist moves inside the reception area. If any of the object's previous centroid is outside the reception area rectangle, then the person came from outside and that is detected as abnormal. However, if none of the object's previous centroids are outside the reception area rectangle, that is not detected as abnormal because it means that the receptionist is moving.

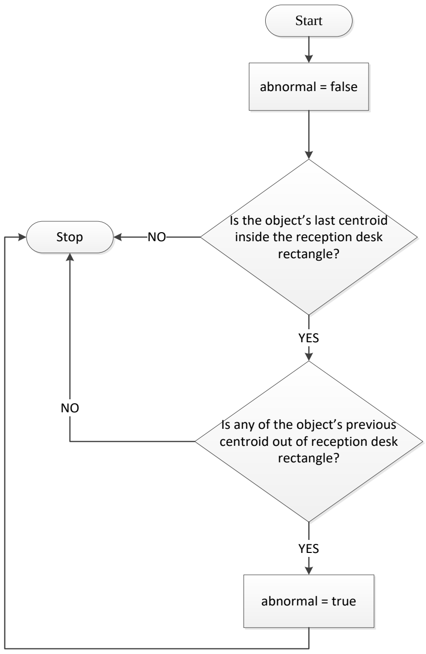

Figure 5. Algorithm to detect when a person enters staff area

5. ALGORITHM TO DETECT RUNNING

Figure 6 shows the algorithm that is used to detect when a person is running. If the number of bounding boxes or centroids in the trajectory history of the track is more than the number of frames used to find velocity then the average velocity of the track is calculated. The number of frames to consider when calculating velocity is a predefined value whose most optimum value can be determined experimentally. In this research, 10 frames were used to find velocity and this is roughly equivalent to 1 second since the frame rate of the test videos were 9.8 frames per second. Such a small value was used because the reception area is very small and it takes very short time to run across the area. If the number of points in the trajectory history are less than frames needed to calculate velocity then the method returns the trajectory as normal.

If there are enough points in the trajectory history then the instantaneous velocity between adjacent centroids are calculated. Each centroid can be represented by its x and y coordinates and this is shown in Figure 7. The velocities of the trajectories are calculated in a loop which runs downward. The counter variable i is initialised to the last centroid in the beginning and the loop is run until the counter decreases by the number of frames needed to calculate the velocity. In each iteration the instantaneous velocity between adjacent centroids i and i-1 are calculated.

$$\text{Euclidean distance} = \sqrt{(\mathrm{X_i} - \mathrm{X_{i-1}})^2 + (\mathrm{Y_i} - \mathrm{Y_{i-1}})^2} \tag{1}$$

$$\text{Velocity} = \frac{\text{Euclidean distance}}{1/\text{frame rate}} \tag{2}$$

The Euclidean distance is calculated using the formula in (1) and then the velocity is calculated by dividing the distance by duration of the frame as in (2). This velocity is added to a variable to find the total velocity of all iterations. Then when the loop has finished, the average velocity is calculated by dividing the total velocity by the number of frames used to find the velocity. Then the average velocity is compared against the running threshold which was determined experimentally. The value used for running threshold is 150. If the average velocity exceeds this threshold then the trajectory is considered abnormal. If the threshold is not exceeded then the trajectory is considered normal.

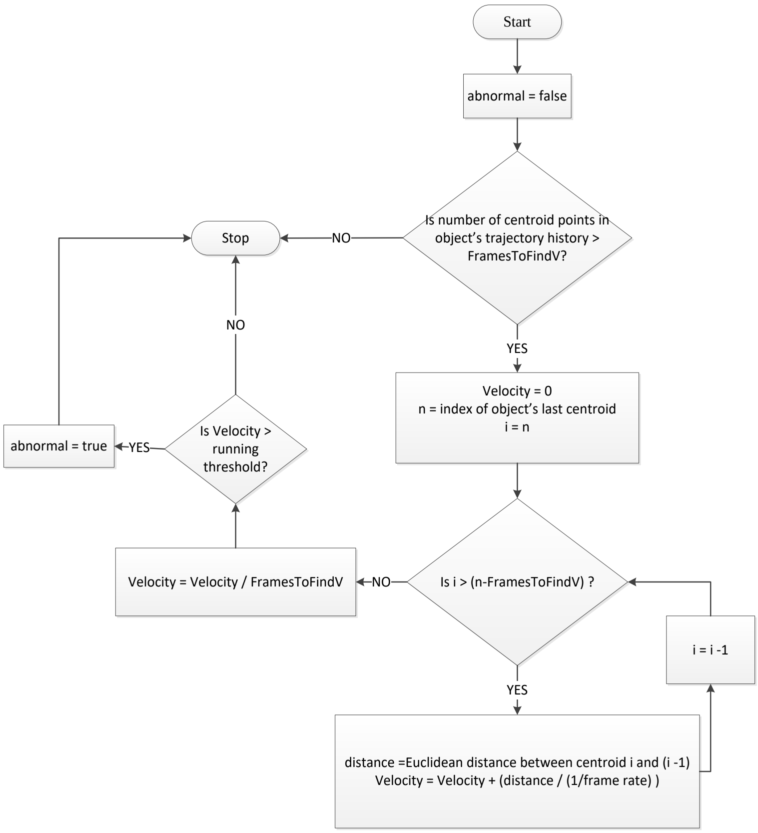

Figure 6. Algorithm to detect when a person is running

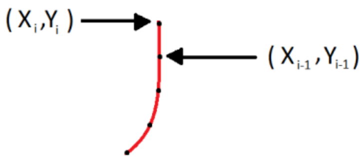

Figure 7. Centroid points on a trajectory

6. ALGORITHM TO DETECT LOITERING

Figure 8 shows the loitering detection algorithm. The method used is adopted from [9]. The two predefined constants used in the beginning of the algorithm are minimum age of the track before its trajectory can be analysed for loitering and the maximum age of the track exceeding which will cause the track to be detected as loitering. In the implementation the minimum age was set to be the age which corresponds to 30 seconds of continuous motion. The maximum age was set to be the age equal to 60 seconds to continuous motion. These constants can be set based on the location where loitering is to be detected. Since the Asia Pacific University's reception area is a very small place, 60 seconds of continuous motion can be flagged as loitering because it is not normal to keep moving continuously in such a small place.

The track's age is first compared to the minimum track age constant and if it is more than the minimum track age constant but less than the maximum track age constant, then trajectory analysis is performed to see if the person is loitering. If the above condition fails then the track's age is compared to maximum track age constant and if it exceeds the constant, the trajectory is considered loitering.

Figure 9 shows the trajectory analysis method used adopted from [9]. A point which is outside the trajectory is taken (point O) and A is the initial point while D= {D_i | I = 1,2, … m} is a collection of m points with a time interval which is a constant called "angleFrameInteval" (1 second in this research). θ_i is the angle between the vector $\overrightarrow{OA}$ and the vector $\overrightarrow{OD}_i$. The angle can be calculated in a loop as stated in [9] by using the following formula.

$$\theta i = \arg\cos < \overrightarrow{OA}, \overrightarrow{ODi} > \quad (3)$$

$$\cos\theta i = \frac{\overrightarrow{OA} \cdot \overrightarrow{ODi}}{|\overrightarrow{OA}|\,|\overrightarrow{ODi}|} \quad (4)$$

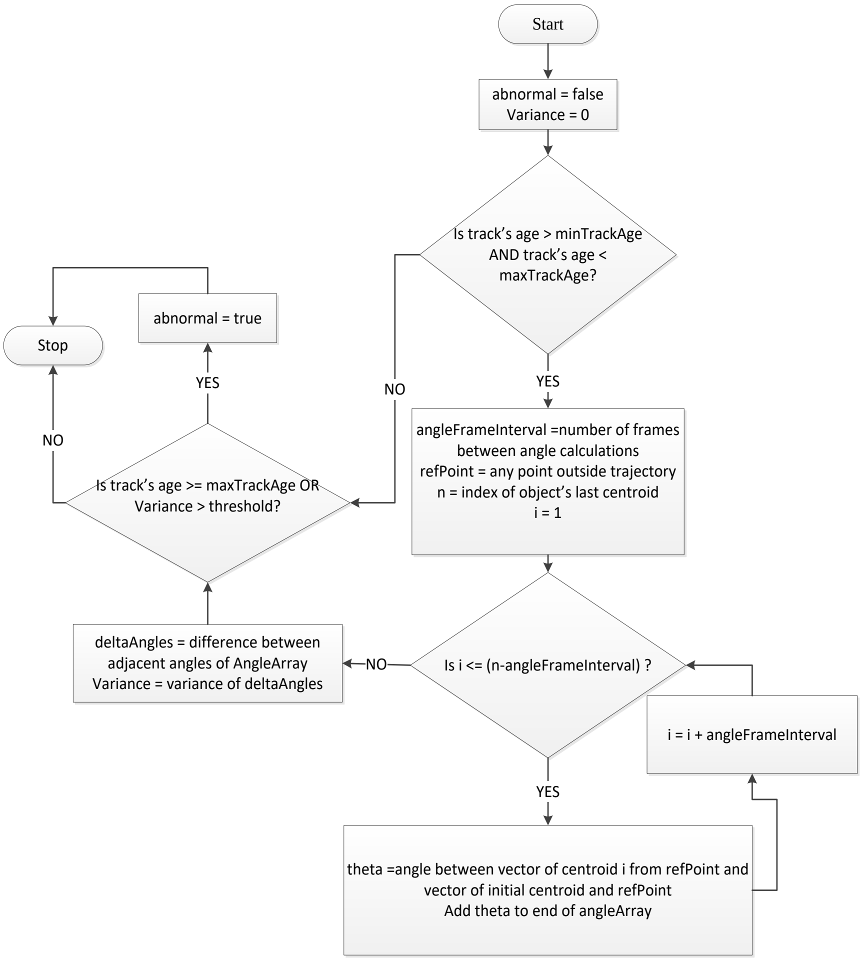

Figure 8. Algorithm to detect loitering

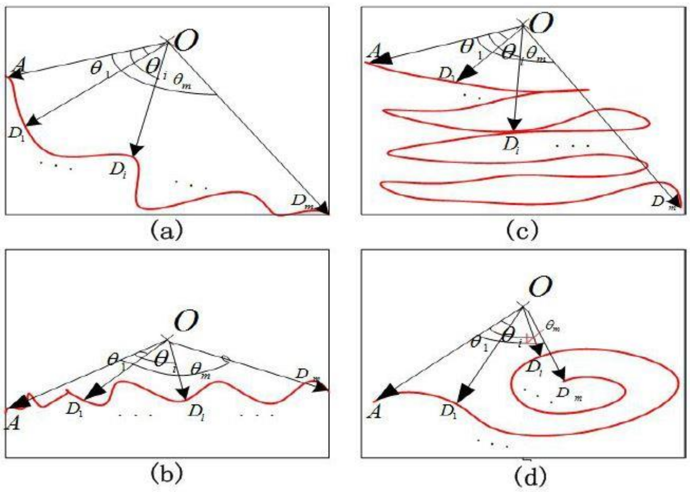

Figure 9. Loitering trajectory analysis method [9]

When the trajectory is loitering the angle θ_i changes periodically. Therefore, the difference between the adjacent angles were calculated next as stated in [9].

$$\Delta\theta i = \theta(i+1) - \theta i \tag{5}$$

The trajectory was then classified as loitering based on the following condition.

$$[\text{track's age} > \text{maximum track age}] \text{ or } [\text{Var}(\Delta\theta i) > \xi] \tag{6}$$

If the track's age is more than the maximum track age constant (60 seconds in this research) then the track was considered loitering. Also, if the variance of the difference between adjacent angles were more than a set threshold, then the track was considered loitering. The threshold can be determined experimentally and the value used in the research was 35.

7. ALGORITHM TO DETECT SQUATTING DOWN

Figure 10 shows the algorithm used to detect if a trajectory is squatting down. Predefined constant n is the number of points from the trajectory history to be used in the detection of squatting down. 'n' points of centroids and BLOB areas from the track are extracted and then they are used to determine if the trajectory is squatting down. The condition for squat down is if the BLOB areas are sorted in descending order which means they are decreasing and y coordinates of the centroids are sorted in ascending order because when someone squats down they are moving downward. Also, standard deviation of x coordinates should be less than an experimentally determined constant A and standard deviation of y coordinates should be more than an experimentally determined constant B. The reason for above conditions are that when someone squats down, horizontal motion (x coordinate movements) will be very little and vertical motion (y coordinate motion) will be more. If the conditions are true then the trajectory is classified as squatting down and the method returns trajectory as abnormal.

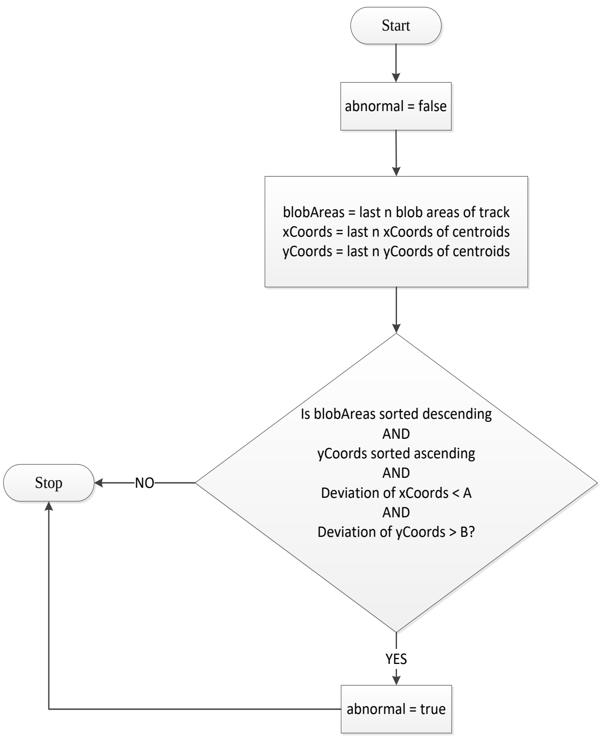

Figure 10. Algorithm to detect squatting down

8. RESULTS AND ANALYSIS

The proposed algorithms were implemented using MATLAB and tested on on an Intel Core i7-6700 machine with 3.40 GHz CPU and 16GB RAM. The MATLAB program was able to process each frame in 35ms with the longest execution path.

8.1. Experimental setup

The proposed algorithms were tested on videos captured by a surveillance camera at Asia Pacific University's reception area. The videos were taken after simulating different scenarios such as entering reception area, loitering, running and squatting down. Videos were taken using single person scenes and multiple people scenes. Figure 11 shows examples of video images used after running the algorithms.

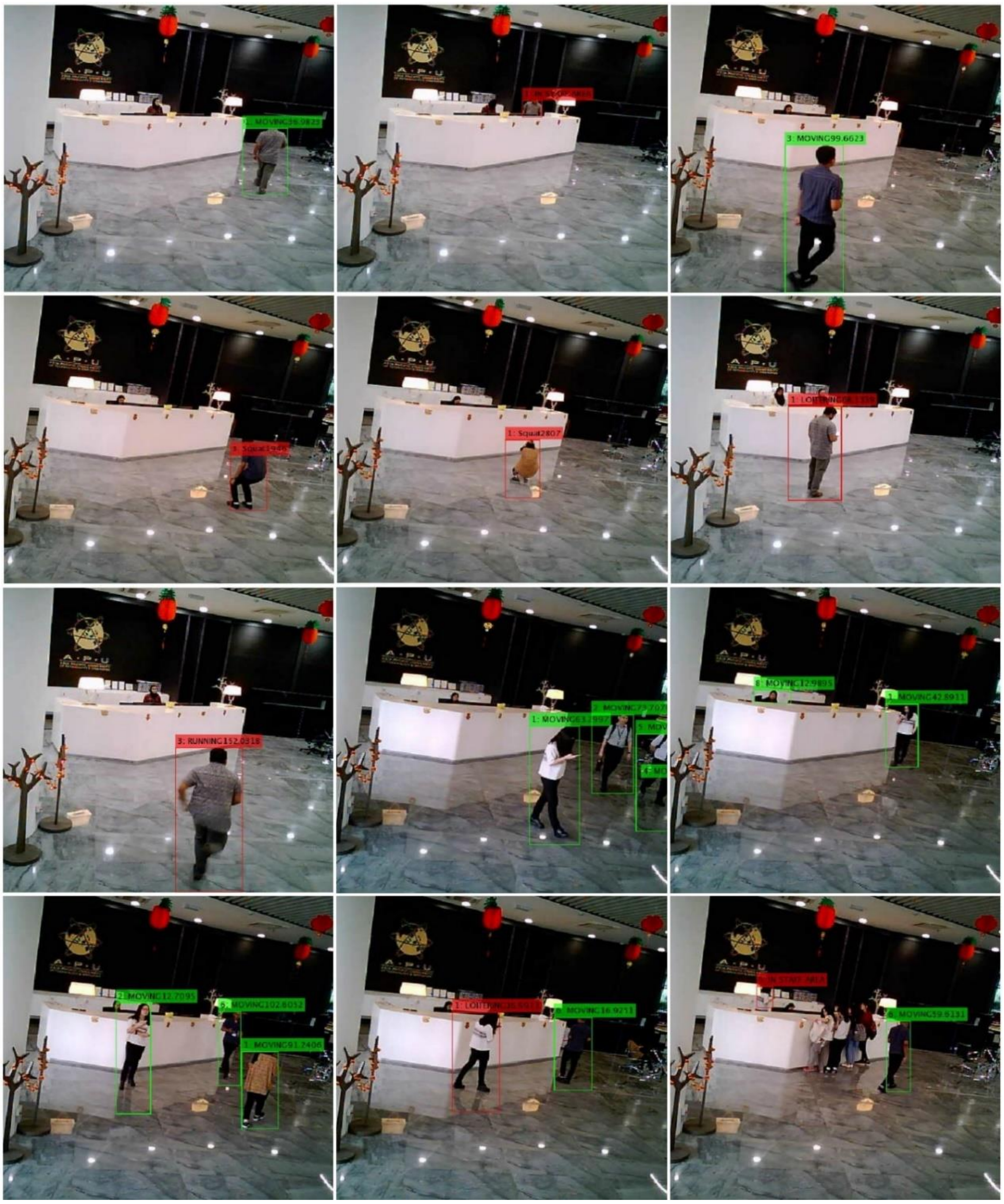

Figure 11. Video images used in testing

8.2. True positive and false positive rate

In True positive test video scenes will be input to the algorithm and the true positive rate of the anomaly detection algorithms are found. True positive rate will show the accuracy of the algorithm in which the algorithm can correctly identify the anomaly as such. The test results are tabulated in Table 1 which shows the description of the video scenes used, number of videos, true positive detections and the % accuracy in detecting a particular anomaly.

Table 1. True positive rate test results

Video scenes	No. of videos	Successful detections	% accuracy
Single person loitering	4	4	100%
Single person running	6	5	83%
Single person entering staff area	5	5	100%
Single person suddenly squats down	4	3	75%
Multiple people walking and one or two people behaving abnormally	8	7	88%
	Average % accuracy		89%

In false positive test video scenes are input to the algorithm and the false positive rate of the anomaly detection algorithms are found. False positive rate will show the rate at which the algorithm identifies anomalies although such an anomaly has not occurred. This test shows the robustness of the algorithm. The test results are shown in Table 2 which shows the description of video scenes used, False Positive (FP) + True Negative (TN) frames, false positive detections and the % of false positive detections.

Table 2. False positive rate test results

Video scenes	FP + TN Frames	False positive detection frames	% detections
Single person loitering	1120	86	7.68%
Single person running	440	7	1.59%
Single person entering staff area	583	24	4.12%
Single person suddenly squats down	315	2	0.63%
Multiple people walking and one or two people behaving abnormally	536	46	8.58%
	Average false positive %		4.52%

An overall accuracy of 89% and the low false positive rate of 4.52% shows that the algorithms are robust. The main source of errors was from the object tracking method. These generated errors also affects the anomaly detection algorithms since the algorithms uses the trajectory extracted by object tracking method. This was due to the floor of the reception area being very reflective and the shadow of the person when moving was highly noticeable in some areas. This causes the tracking algorithm to include shadow in the bounding box of the person in some frames as the shadow is moving as well, leading to the centroid of the bounding box to change very rapidly in some instances. The negatively affected running detection algorithm which calculates the distance between adjacent trajectory points is the reason why there were many instances of false positive detection of running.

In addition, the highly reflective floor also leads to BLOBs of multiple people getting mixed together to become one large BLOB because when the people get closer, their shadows pass through each other and it becomes one. This also leads to the tracks of multiple people getting interchanged when their BLOBs separate. This happens because detections of tracks are assigned based on the distance between predicted centroid and detected centroid. When two BLOBs become one and separate, their distances are very close together.

Moreover, there were tracking errors due to small light intensity changes and these get detected as moving objects. However, deleting detections based on BLOB area with respect to the distance of the person relative to the camera position as explained in section 3 was very effective and greatly reduced errors.

There was also a problem when the receptionist moved within the reception area and getting detected as an anomaly due to being in staff area. This was solved by making sure the person came from outside the reception desk before being detected as being in staff area. However, there will still be the same problem when the receptionist moves out of the reception desk area and comes back in. But the frequency at which this happens will be much less than the receptionist moving within the reception area.

9. CONCLUSION

In conclusion, multiple object tracking was performed using Kalman Filter and four anomalies in a reception area context were detected which were entering staff area, running, loitering and squatting down. The algorithms were tested on videos captured at Asia Pacific University's reception area in which an average detection accuracy of 89% was achieved showing the effectiveness of the proposed method. Besides, a false positive rate of 4.52% was achieved which shows the algorithms were very robust. This proves that a rule-based approach to anomaly detection can also achieve good performance compared to other approaches. This work could be further enhanced by developing algorithms with machine learning capability to detect crowd-based anomalies.

REFERENCES

[1] Wiliem, A., Madasu, V., Boles, W. & Yarlagadda, P., "A suspicious behaviour detection using a context space model for smart surveillance systems," *Computer Vision and Image Understanding*, vol. 116, no. 2, pp. 194-209, 2012.

[2] Li, H., Achim, A. & Bull, D., "Unsupervised video anomaly detection using feature clustering," *IET Signal Processing*, vol. 6, no. 5, pp. 521, 2012.

[3] Li, C., Han, Z., Ye, Q. & Jiao, J., "Visual abnormal behavior detection based on trajectory sparse reconstruction analysis," *Neurocomputing*, vol. 119, pp. 94, 2013.

[4] Ko, J. & Yoo, J., "Rectified Trajectory Analysis Based Abnormal Loitering Detection for Video Surveillance," *IEEE*, pp. 289, 2013.

[5] Wang, T., Chen, J. & Snoussi, H., "Online Detection of Abnormal Events in Video Streams," *Journal of Electrical and Computer Engineering*, vol. 2013, pp. 1-12, 2013.

[6] Wang, T. & Snoussi, H., "Detection of Abnormal Visual Events via Global Optical Flow Orientation Histogram," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 988-998, 2014.

[7] Xu, D., Song, R., Wu, X., Li, N., Feng, W. & Qian, H., "Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts," *Neurocomputing*, vol. 143, pp. 144, 2014.

[8] Mo, X., Monga, V., Bala, R. & Fan, Z., "Adaptive Sparse Representations for Video Anomaly Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 631-645, 2014.

[9] Li, W., Zhang, D., Sun, M., Yin, Y. & Shen, Y., "Loitering Detection Based on Trajectory Analysis," *IEEE*, pp. 530, 2015.

[10] Patil, S. & Talele, K., "Suspicious movement detection and tracking based on color histogram," *IEEE*, pp. 1, 2015.

[11] Wang, T. & Snoussi, H., "Detection of abnormal events via optical flow feature analysis," *Sensors (Basel, Switzerland)*, vol. 15, no. 4, pp. 7156-7171, 2015.

[12] Moria, K., Albu, A.B. & Wu, K., "Computer Vision-Based Detection of Violent Individual Actions Witnessed by Crowds," *IEEE*, pp. 303, 2016.

[13] Biswas, S. & Venkatesh Babu, R. "Anomaly Detection Via Short Local Trajectories," *Neurocomputing*, 2017.

[14] Leyva, R., Sanchez, V. & Li, C., "Video Anomaly Detection With Compact Feature Sets for Online Performance," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3463-3478, 2017.

[15] Sehairi, K., Benbouchama, C., El Houari, K. & Fatima, C., "A Real-Time Implementation of Moving Object Action Recognition System Based on Motion Analysis," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 5, no.1, pp.44-58, 2017.

[16] Hussain, S., Ramaiah, C. and Hussain, S., "A Hardware Model to Measure Motion Estimation with Bit Plane Matching Algorithm," *Telecommunication Computing Electronics and Control* (*TELKOMNIKA)*, vol.16, no.5, pp.2091-2100, 2018.

BIOGRAPHIES OF AUTHORS

Muhammed Shuaau was born in the Maldives in 1994. He is currently doing his undergraduate degree in Electrical and Electronic Engineering in Asia Pacific University of Technology & Innovation. His field of interests include image processing and, aritifical intelligence.

Ka Fei, Thang graduated with a Doctor of Philosophy in Electrical Engineering and a Bachelor of Engineering (Honours) degree in Electrical and Electronic Engineering from the University of Bath, UK. He is also a Chartered Engineer and a member of the Institution of Engineering and Technology (IET), UK. Dr Thang is now serving Asia Pacific University of Technology and Innovation (APU) as Associate Professor and Head of School of Engineering. His responsibilities are to manage the quality processes of the School in teaching and learning as well as ensuring continuous professional accreditation for all the engineering programmes.

N. S. Lai born in Seremban, Malaysia, is currently an Associate Professor at School of Engineering, Asia Pacific University of Technology & Innovation, working in the area of micro- and nano-electronics. He received his B. Eng. (Hons) in Electrical Engineering (2007) and Ph.D. in the field of Silicon-Based Nano-electronics and Radiation Detectors (2012) from University of New South Wales, Australia. His main research interest includes semiconductor micro- and nano-fabrication, quantum dot devices, low temperature (milli-Kelvin) measurements, and semiconductor microdosimeters.