Anomalous Object Tracking in Distributed Camera Network

MD Shahbaz Khan, Dr Indu Sreedevi
Delhi Technological University
New Delhi, India
mdshahbaz_bt2k16@dtu.ac.in, s.indu@dtu.ac.in

Abstract—**The paper introduces a novel framework for real-time tracking of an object with higher precision in a Pan-Tilt-Zoom (PTZ) camera network. In the above-mentioned framework, the object which behaves anomalously is picked for tracking. An SVM classifier has been used to pick anomaly behaving object among all the objects present in a frame. It is a self-initializing algorithm as it does not require user intervention for object detection. Target detection is followed by its autonomous tracking in the distributed camera network. Hence it does not make the system bulky and takes less time to execute. The paper implements multiple object tracking using Discriminative Correlation Filtering with Channel and Spatial Reliability(CSR-DCF) tracking algorithm. Existing works are based on Particle filtering or Kalman filtering algorithms which are computationally complex and not self-initializing.**

I. Introduction

An efficient and fast algorithm is required for tracking a specified object from a frame having many objects using distributed PTZ camera based network. For each frame, some objects are at a higher priority for tracking than the rest. Although it is possible that none of the objects needs to be tracked in a particular sequence of frames. In such cases, tracking is not initialized till an anomaly behaving object is detected. Act of running, bending or picking was considered as anomalous behaviour. For detecting an anomaly behaving object, SVM based classifying model is trained. The distance of centroid of an object from its contour was used as a feature for developing the SVM model. This leads to the enhancement of the computational efficiency of the proposed algorithm. Sink frame of a particular camera in the network serves as the source frame for the other. For the mentioned decision-making task, the paper proposes a real-time, computationally efficient algorithm in which the camera having sink frame generates a histogram and is transmitted to all the adjacent cameras. Other cameras generate a histogram for their respective source frame and compare it with the received one. Camera having the same histogram starts tracking the object. Use of histogram formation and their comparison results in a smooth tracking at the time of object handover from one field of view (FOV) to another. Real-time and computationally fast algorithms are required for surveillance of densely populated public places using distributed PTZ cameras. The proposed approach assumes overlapping FOV between adjacent cameras in the network with the same resolution of the sink and the source frame of the neighbouring cameras.

II. Literature Layout

Several algorithms have come up to make object tracking autonomous with higher precision and real-time. However, on increasing the number of cameras in the network, their computational efficiency decreases manifold. Algorithms like [1],[2] do not consider the formation of the shadow of objects while few other algorithms like [3] are not suited for public places where the number of objects is more than an ordinary scenario. Current works in this field [3] attempts to track the object using particle filtering algorithm along with user intervention. As far as places with larger crowds are concerned, particle filtering algorithms are not efficient and do not yield real-time processing of data. CSR-DCF based tracking is a multi-object tracking algorithm and hence it can handle a large number of objects in crowd places. Paper [2] deals in Kalman filtering based object tracking, which is not efficient for real-time object tracking. In the case of Kalman filtering, noise is measured by modelling it as additive white Gaussian noise (AWGN). This makes the noise measurement in each frame as a joint Gaussian distribution. This, in turn, makes the object tracking vague when there is a bifurcation in the path of the object. This paper implements CSR-DCF tracking algorithm which is one of the best available real-time object tracking algorithm as discussed in table II. In papers like [1], object tracking is initialized by detecting the object in the very first frame by using the user interface. This, in turn, binds the algorithm from being fully autonomous. To eliminate this bottleneck, our approach proposes the concept of self-initialization of the object. Self initialization can be defined as the detection of an object of interest by the system itself. This paper implements self-initialization by choosing the object of interest as per its anomalous behaviours among all the objects present in that frame. The anomaly behaving object is picked using the SVM model. The model has been trained by using the distance of centroid of the object from its contour. No sooner did the object is detected, then its tracking gets started in all the cameras present in that distributed camera network. This algorithm proves to be efficient for the crowded region as it can track the movement of a suspicious object by itself. Each camera in the network tracks the object and communicates with adjacent cameras when the object of interest leaves its FOV. This novel method of object handover eliminates the chance of object missing during tracking in a multi-camera system and hence yields a higher accurate tracking.

III. PROBLEM STATEMENT

The paper aims to find the solution for the detection of anomaly behaving object followed by its tracking in a distributed camera network. The intent is to improve the tracking accuracy and precision in comparison with probability and Gaussian distribution-based tracking algorithms[3] and [2] like particle filtering and Kalman filtering. The proposed methodology also eliminates random id generation technique[3] used to track objects in multiple cameras. Moreover, the proposed algorithm of object handover reduces the probability of object missing during its tracking in a multi-camera network system.

IV. ANOMALOUS BEHAVIOUR DETECTION

The prerequisite for this algorithm is the minimum number of objects to be present in the camera frame which is achieved by classifying objects as normal and abnormal behaving groups. Activities like running, bending, or the act of picking are classified as anomalous behaviour in this paper as shown in figure 5 & 8. The paper proposes Principal Component Analysis (PCA) based feature extraction followed by its classification as anomalous behaviour using Support Vector Machine (SVM) classifier as discussed in [4] and the result has been collected in the table I. To prepare an SVM model, the maximum distance of any point on the contour from the centroid of the object was calculated and was used as features for SVM classier.

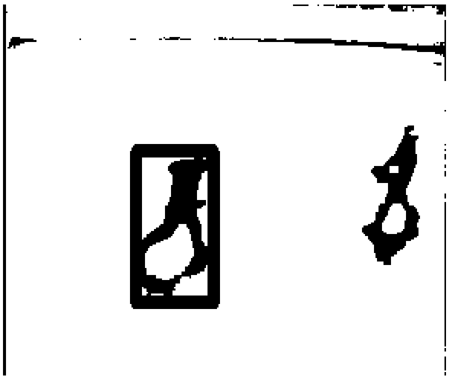


Fig. 1. A person running classified as anomalous behaviour

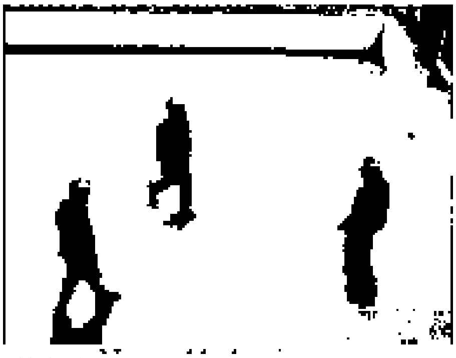


Fig. 2. Normal behaviour

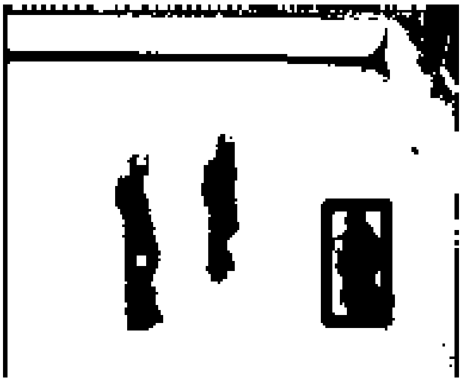


Fig. 3. A person bending in the pathway – classified as anomalous behaviour.

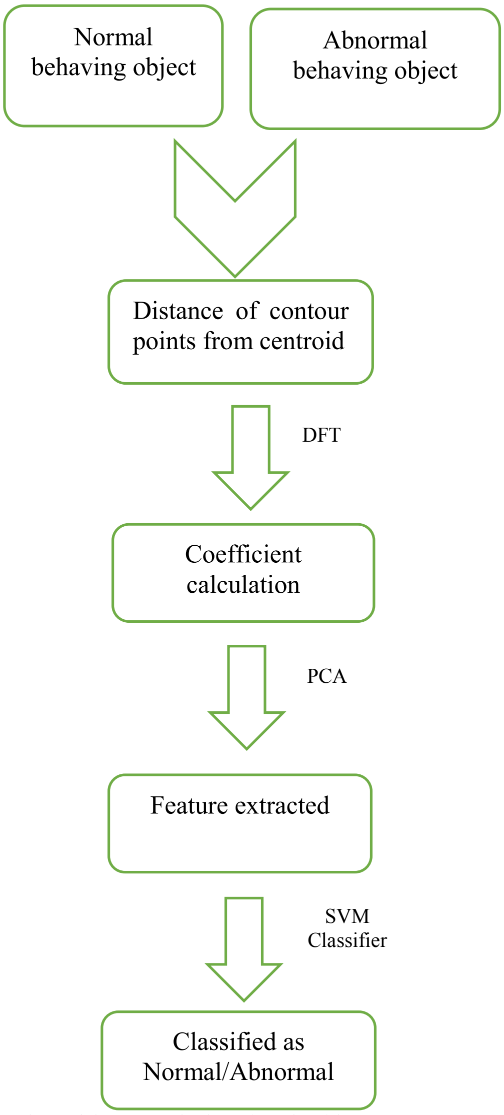


Fig. 4. Flowchart of the Anomaly detection algorithm.

First of all, the contour of every object was formed. Then using the below-mentioned formula, the centroid of that object was obtained.

$$x_c = \frac{1}{N}\sum_{1}^{N} x_i \tag{1}$$

$$y_c = \frac{1}{N}\Sigma_1^N y_i \qquad (2)$$

where x_c, y_c are the coordinates of the centroid of contour.

For each point on contour, Euclidean distance was calculated and was transformed to coefficients by using DFT (Discrete Fourier Transform), which was later used for feature extraction using PCA (Principal Component Analysis).

A. PCA Based Feature Extraction

To extract features from each frame of camera input, PCA algorithm was implemented. The training dataset was divided into two groups namely G1, G2 each having 'n' training samples. Eigenvector of features was generated as follows: -

1) For each group of the dataset, a matrix '*X*' was formed having 'n' rows of samples and 'c' columns of features.

2) From the feature matrix '*X*', covariance matrix 'C' was computed as

$$C = (X - X^{`})^{T} (X - X^{'}) \qquad (3)$$

3) For the computed covariance matrix, eigenvector and eigenvalues were determined using the following relation

$$CV = V\Lambda \qquad (4)$$

Where V is eigenvector and Λ is an eigenvalue.
Out of 'c' columns of Λ, we picked 'k' number of columns thus reducing the feature vector size to n x k.

B. SVM Classifier

As per the statistical learning theory [5],[6] & [7], SVM is one of the most precise classifying algorithms. The paper implements SVM for classifying the extracted features into normal and abnormal behaviour by using separating hyperplane. The constructed hyperplane was localized by maximizing the separating margins and consequently, it led to the formation of nonlinear decision boundary in the training dataset.

By implementing SVM algorithm, an optimal hyperplane was found such that it divides the dataset into two groups completely. In the course of building the SVM model, a hyperplane is adjusted by maximizing the margin between data values and hyperplane. Margin maximizes by minimizing the cost function.

Where cost function $J(x, y, f(x))$ is defined as,

$$J(x, y, f(x)) = \begin{cases} 0, \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), \text{else} \end{cases} \qquad (5)$$

On building the SVM model for training data, it was tested on the testing dataset whose labels was known. The hyperplane was further adjusted to get the maximum accuracy of classification. At the end of testing, the SVM model was used for unknown data.

Anomaly Behaviour Detection Algorithm

Prerequisites: Continuous video dataset, the dataset must contain anomaly behaviour, the dataset should contain video of the different timing of day having varying brightness.

Behaviour Detection:

1. Dividing the dataset into training and testing data group in a ratio of 1:4.
2. Extracting features from training dataset.
3. Optimizing number of features required to detect behaviour using PCA.
4. Adjusting the location of hyperplane by minimizing the cost function using SVM.
5. Applying the trained model to the testing dataset and evaluating the precision percentage.
6. If the precision is not more than 85% (as depicted in table I), then the number of data example in training dataset was increased and step 6 was repeated.
7. Else proceed to apply the trained model to an unknown dataset.

V. CSR-DCF ALGORITHM

In this paper, we have applied background subtraction to our input data before detecting anomaly behaviour. For the abovementioned task, background segmentation algorithm (based on Gaussian Mixture) was used, as it flexible for diurnal changes and for shadow detection as well. This algorithm is based on two research papers [8],[9] given in the year 2004 and 2006 respectively. On detecting an abnormal behaving object in a frame, the tracking algorithm starts executing. A rectangle bounding box is formed around that region of interest with a different colour for different objects in each frame. CSR-DCF algorithm was implemented for tracking objects which is a real-time processing algorithm with higher precision.

As discussed in [10], CSR-DCF is a tracking algorithm which provides learning with higher efficiency and better integration of filter updates and tracking. In this algorithm, spatial reliability map is used which helps to adjust filter support to enhance the accuracy of tracking. Similarly, reliability scores are calculated to check the quality of filters and thus helps in evaluating coefficients in localization.

This paper proposes a novel framework of object tracking based on discriminative correlation filtering followed by channel and spatial reliability scores[11]. It is a proximity-based algorithm which learns a set of filters to represent the object and uses them to find the highest correlated point in the adjacency of tracking an object from the previous frame.

A. Spatially Correlation Filters

With the help of the following approach, we comprehend the importance of learning correlation filters. For a given set of 'n' channels with features f = $\{f_k\}_{k=1:n}$ and target templates h = $\{h_k\}_{k=1:n}$, and the position of the object is determined by locating the highest frequency of ĝ(h),

$$\hat{g}(h) = \sum_{k=1}^{n} f_k \star h_k \qquad (6)$$

where $\star$ is circular convolution operator & $\hat{g}(h)$ is correlation response.

Similarly, to find optimum correlation filter h, we have to minimize

$$\mathcal{E}(h) = \|\hat{g}(h) - g\|^2 + \lambda\|h\|^2 \qquad (7)$$

where g is the expected output.

To find the efficiency of tracking, we transformed it into Frequency Domain as

$$\mathcal{E}(h) = \| \sum_{k=1}^{n} diag(\hat{f}_k)\hat{h}'_k - \hat{g}' \|^2 + \lambda \|h\|^2 \qquad (8)$$

And on minimizing the above equation we get,

$$\hat{h}_k = (diag(\hat{f}_k)\hat{g}') \odot^{-1} (\sum_{k=1}^{n} diag(\hat{f}_k)\hat{f}'_k + \lambda) \qquad (9)$$

where $\odot^{-1}$ is element-wise division operator and the above result can be used to design correlation filter trackers.

On considering each channel to be independent, we get the cost function as

$$\mathcal{E}(h) = \sum_{k=1}^{n} \|f_k \star h_k - g\|^2 + \lambda \|h\|^2 \qquad (10)$$

To scale, we introduce a factor depending upon the discriminative power of feature channels which is termed as channel weights w = $\{\hat{w}_k\}_{k=1:n}$ and whose correlation response is given by,

$$\hat{g}(h) = \sum_{k=1}^{n} f_k \star h_k . \hat{w}_k \qquad (11)$$

B. Correlation Filter Learning

As mentioned above, it has been assumed that the correlation filter is independent of channels, hence it allows to consider the case of one filter only. The constraints are formalized as following

$$h_c \;\; m \odot \hbar \equiv 0 \qquad (12)$$

where h_c is defined as the dual variable with the above constraints.

On using Lagrangian augmentation [9], we find the following outcome

$$s\mathscr{L}(\hat{h}_c, h, \hat{I}|m) = \|diag(\hat{f}_k)\hat{h}'_k - \hat{g}\|^2 + \lambda \|h_m\|^2$$
$$+ [\hat{I}^H(\hat{h}_c - \hat{h}_m) + \hat{I}^H(\hat{h}_c - \hat{h}_m)'] + \mu \|\hat{h}_c - \hat{h}_m\|^2 \qquad (13)$$

Here in the above equation, we have introduced a new term μ which is the constraint penalty value and its value is updated by the following equation.

$$\mu^{i+1} = \beta\mu^{I} \qquad (14)$$

C. Spatial Reliability Map

On locating the target, the region of interest is extracted from the frame and is used for updating the filter. The constrained filter uses spatial reliability map m_p, which is the collection of pixels of training model matching with that of the testing mode. To determine m_p, we have designed the following algorithm.

While the algorithm is getting executed for tracking, foreground, and background of the object is stored in colour histogram c. Let $y_i = [y_i^c, y_i^x]$ is the observation i.e., it denotes the colour of y_i^c along with the y_i^x position for each pixel and m_p = random variable indicatıng the unknown foreground/background. For y_i to be observed on output, its joint probability was found as follows

$$p(y_i) = \sum_{j=0}^{1} p(y_i | m_i = j)p(m_i = j)$$
$$= \sum_{j=0}^{1} p(y_i^c | m_i = j)p(y_i^x | m_i = j)p(m_i = j) \qquad (15)$$

Algorithm 1 CSR-DCF

Prerequisites: Image I_i, the position of an object in previous frame pos_{i-1}, scale k_{i-1}, filter h_{i-1}, image histograms c_{i-1}, channel reliability w_{i-1}.

Determination and approximation of scale:-

1. Locating target at pos_i : finding the highest frequency in a correlation between filter (h_{i-1}) and extracted features (f) at pos_{i-1} position and adjusted by reliability scores (w).
2. Calculation of $\hat{w}$(det) (detection validation) with the help of per-channel responses.
3. Scale k_i was calculated, with the help of pos_i.
4. Colour histograms for foreground as well as the background were calculated and updated using $\hat{w}$(det) (detection validation).
5. Reliability map m_p was calculated using the above-mentioned relation.
6. Filter $\hat{h}$ was calculated and updated using m_p.
7. Using h, learning channel reliability was determined.
8. Calculate channel reliability from learning channel reliability & detection reliability.
9. Each filter hi was updated using the relation $h_i = (1-\eta)h_{i-1} + \eta\hat{h}$.
10. Channel reliability w_i was updated using the expression $w_i = (1-\eta)w_{i-1} + \eta\hat{w}$.

VI. Object Handover

In a distributed camera network, each camera is capable of processing the data and extracting information out of it. Cameras in a network can communicate with each other wirelessly in a range of 100m. The exchange of data between two cameras was done using OFDM modulation (Orthogonal

Frequency-Division Multiplexing). The advantage of using OFDM modulation is that it provides versatility in the presence of multiple channels.

The problem arises in handing over the object from one camera to the other. To eliminate the object handover problem, this paper implements the following algorithm:-

OBJECT HANDOVER ALGORITHM

Case 1: If the system has not yet detected any anomaly behaviour.
    i. In this case, each camera will keep on processing the input frames for anomaly detection.
    ii. If any anomaly behaviour is detected, then that camera will switch to case 2 else will continue to execute case1.
Case 2: if any anomaly behaviour is detected.
    i. Object tracking algorithm will be applied to the camera input in which anomaly action has been detected.
    ii. Camera tracking the anomaly behaving object will stop executing the anomaly detection algorithm for new objects.
    iii. As soon as the object becomes the sink for a camera, the camera generates its histogram from the last frame (say h1).
Case 3: when the object moves out of FOV of one camera
    i. In this case, the anomaly detection algorithm is paused and for each camera histogram of the new entering object is calculated (say h2).
    ii. Then both colour histograms (h1 and h2) are compared and the camera (say C2) whose histogram matches with h1, is now used for tracking.
    iii. The object tracking algorithm is applied to the camera (C2) input frame and case 2 is followed.

*Figures 6,7 and 9,10 illustrate the above-mentioned algorithm.

VII. Experimental results

The above experiment was performed on a dataset collected from Computer Vision Laboratory CVLAB ('https://cvlab.epfl.ch/data/data-pom-index-php/'). The mentioned dataset consists of inputs of 4 cameras located at different worldly points but with the same intrinsic parameters. On implementing the above-proposed algorithm, real-time object detection was done using anomaly behaviour detection technique with the following results

TABLE I. RESULT OF SVM CLASSIFIER

RESULT OF SVM CLASSIFIER		
Behaviour	***Success rate***	***No. of testing samples***
Normal	97%	297
Anaomaly	86%	328

The above-proposed algorithm was tested on the dataset at VOT2016 [12] benchmark which is comprised of 14 videos (7 training and 7 testing). On rigorous testing and comparison with various tracking algorithms, we got the following result (Table II).

TABLE II. COMPARISONS OF TRACKING ALGORITHMS

Tracking Algorithm	**EAO**	**Aav**	**Fav**	**Total**
Proposed method	0.34	0.510	0.851	-0.001
CCO Tracking	0.33	0.520	0.856	-0.006
TCN Net.	0.33	0.521	0.961	-0.110
SSA Tracking	0.32	0.531	1.044	-0.193
MLD Filter Based	0.31	0.482	0.833	-0.041
Staple	0.30	0.512	1.351	-0.539
DOC	0.29	0.513	1.233	-0.430
EB Tracking	0.29	0.441	0.902	-0.171
SRB Tracking	0.29	0.504	1.251	-0.547
STAPLEp	0.29	0.533	1.321	-0.498

*EAO – Expected Average Overlap
*A_{av} – Average Accuracy
*F_{av} – Average Failure
*Total = EAO + A_{av} - F_{av}

The above-mentioned comparison of various tracking algorithms was carried on the ground of expected average overlap, average accuracy and average failure. For an ideal tracking algorithm expected average overlap and average accuracy should be higher while lower average failure is preferred. In the case of real tracking algorithms, we need to compromise between these three parameters. To bring uniformity in comparison, a total usefulness function formed whose formula has been mentioned above. The higher total usefulness value for a tracking algorithm is, better that algorithm is to use. From the above table, we conclude that our tracking algorithm is better among the rest as its total usefulness value is higher than rest tracking algorithms.

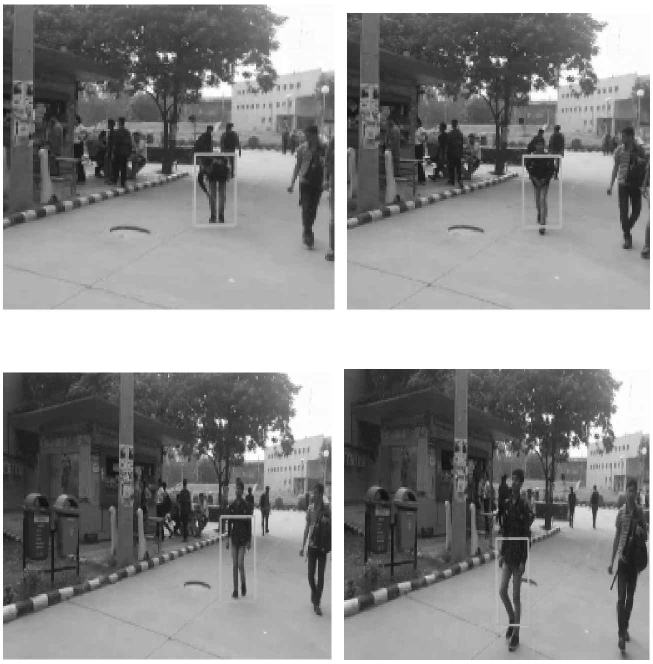


Fig. 5. A person bending classified as anomalous behaviour by the camera C1.

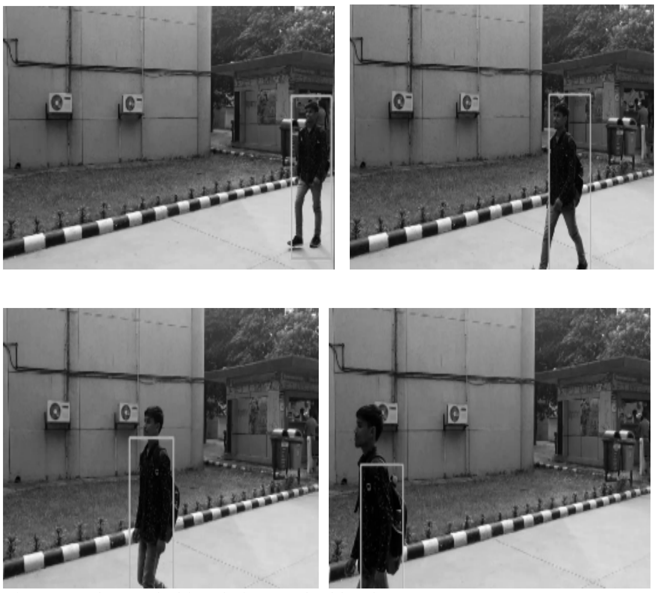


Fig. 6. C1 detected object being tracked by the camera C2.

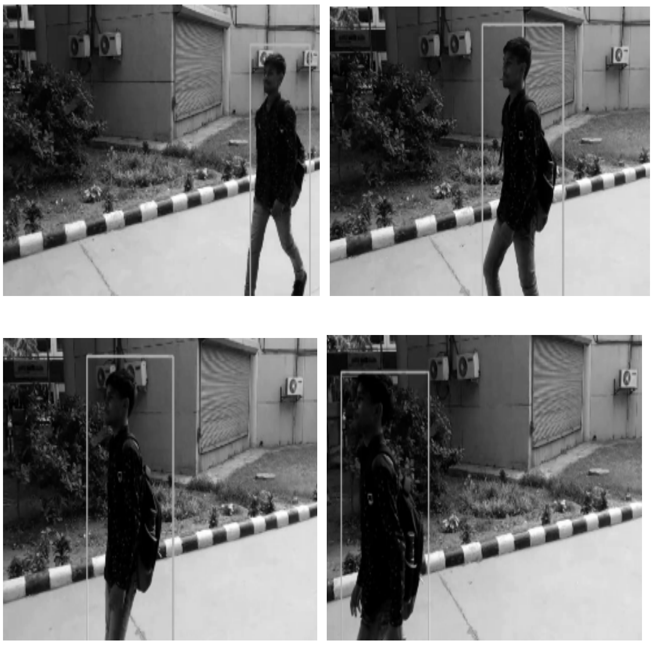
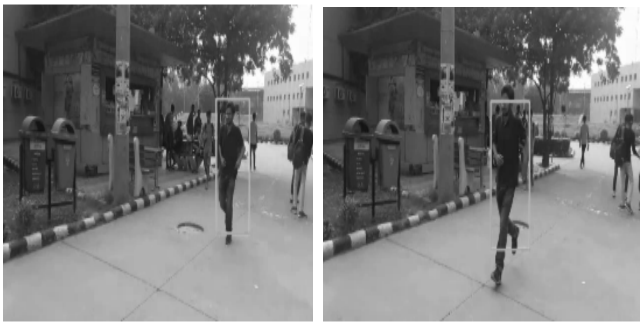


Fig. 7. Tracking object moving out of C2 camera being tracked by the camera C3



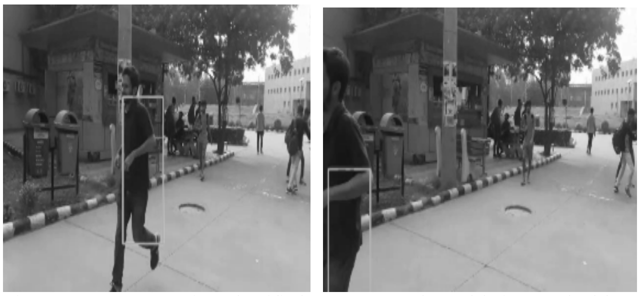


Fig. 8. A person running classified as an anomaly and being tracked by the camera C1.

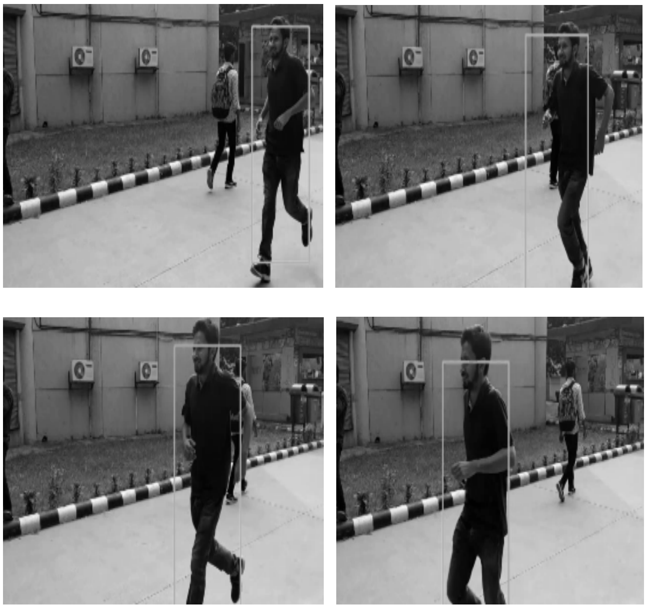


Fig. 9. C1 detected object being tracked by the camera C2

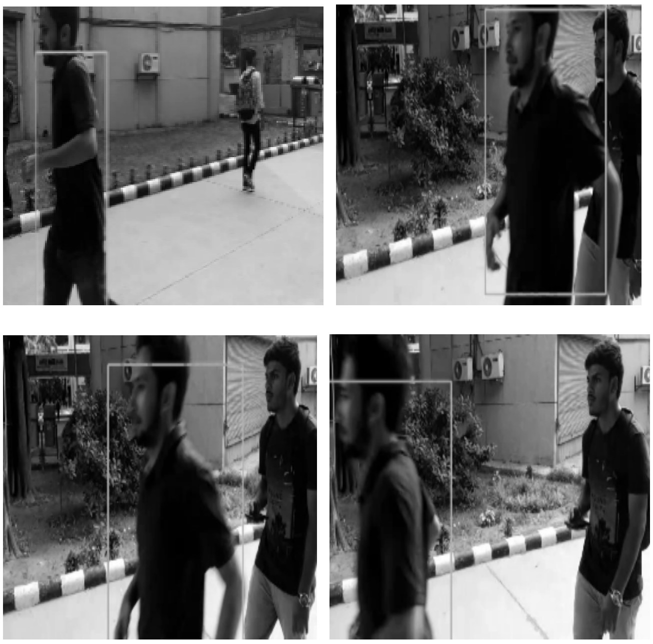


Fig. 10. Tracking object being moved out of C2 and tracked by camera C3

*Fig. 6-10 depicts the results of packing anomaly behaviour (running, bending and picking) using 3 cameras C1,C2 and C3.

VIII. Conclusion and future scope

This novel framework proposes an accurate and precise method for object tracking in distributed cameras. It is a real-time based approach with higher computational efficiency for tracking in a camera network. One of the important feature of this algorithm is self-initialization, which do not require user interventions. The proposed algorithm deals with the tracking of anomaly behaving objects using CSR-DCF based algorithm. Running, bending or picking has been classified as anomaly behaviour and based on this classification, objects are picked for tracking. For tracking the object, CSR-DCF algorithm has been implemented which is faster and more precise. Histogram comparison-based algorithm has been implemented to handover the object from one camera to the other in the network system.

Acknowledgement

I express my deepest gratitude to all those who helped me make this research complete and meaningful in all sense. First of all, I would like to thank our faculty in-charge, Prof. S. Indu & Dr Ayesha Chowdhary, whose contribution to this research work is incredible. They helped me to get through this research work by providing their valuable suggestions and sparing their precious time. Furthermore, I acknowledge the contribution of other staffs of the ECE department (DTU), who helped me by providing their valuable suggestions related to my topic. Last but not least, I am highly thankful to my parent and friends, who provided me moral support in times of distress and failure and played a vital role to make this work complete in all sense.

References

[1] G. R. Bradski, "Real time face & object tracking as a component of a perceptual user interface," IEEE workshop on Application of Computer Vision, Princeton, NJ USA, 1998.

[2] S. Weng, C. Kuo and S. Tu, "Video Object Tracking using Adaptive Kalman Filter", *Journal of Visual Communication & Image Representation*, 17(2006), 1190–1208.

[3] A. Choudhary, S. Sharma, I. Sreedevi and S. Choudhary, "Real-Time Distributed Multi-object Tracking in a PTZ Camera Network", *Springer*, vol. 9127, 2015, pp. 183-192.

[4] X. Wu, Y.Ou, H. Qian, Y. Xu, "A detection system for Human Abnormal Behavior" RSJ International Conference on Intelligent Robots and Systems, Canada, 2005.

[5] K. Lee and Y. Xu, "Modeling human actions from learning" RSJ International Conference on Intelligent Robots and Systems, Japan, 2004.

[6] B. Scholkopf and A.J Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Massachusetts: MIT Press, 2002.

[7] Y. Ou and Y. Xu, "Learning human control strategy for dynamically stable robots: Support Vector Machine approach" International Conference on Robotics and Automation, Taiwan, 2003.

[8] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction" ICPR, UK, 2004.

[9] Z. Zivkovic and F. Hejiden, "Efficient adaptive density estimation per image pixel for the task of background subtraction", *Pattern Recognition Letters*, vol. 27, No. 7, 2006, pp. 773-780.

[10] A. Lukezic, T. Vojir, L. Zajc, J. Matasa and M. Kristan, "Discriminative Correlation Filter with Channel and Spatial Reliability", *Internaltional Journal of Computer Vision*, 126(2017), 6309–6318.

[11] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovi, G., G. Fernandez and T. Vojir "The Visual Object Tracking, The Visual Object Tracking Challenge Results", 2013, pp. 98-111.

[12] M. Kristan, et al, "The Visual Object Tracking VOT2016 Challenge Results", Proc. European Conference Computer Vision, 2016.