

Multi-Camera Based Surveillance System

Reena Kumari Behera, Pallavi Kharade, Suresh Yerva, Pranali Dhane, Ankita Jain, and Krishnan Kutty

Center for Research in Engineering Sciences and Technology (CREST)

KPIT Cummins Infosystems Limited

Pune 411057, India

Abstract—In recent years, surveillance systems have gained increased importance in order to increase the safety and security of people. These systems have applications in various domains like home or bank security, traffic monitoring, defense; and in public places like railway stations, malls, airports, etc. Our goal is to develop an intelligent real-time surveillance system that can help in increasing the efficiency of the system. In order to cover a large area, we need to install more number of cameras that leads to more number of videos that are to be monitored simultaneously. This in turn increases human intervention and makes it error prone. Therefore, it is of utmost importance to automate the complete system. In the proposed system, cameras are placed in such a way that there is a significant overlap between the field of view of the cameras. This helps in establishing an association between the cameras. The proposed real time surveillance system detects and tracks the objects in motion and provides automatic warning in case of suspicious activities such as unidentified object and restricted zone monitoring. The system provides different views in different cameras, thereby helping in resolving the occlusion in a simple and novel way. The algorithm has been tested on a CPU platform (Intel dual core with 4 GB RAM) with four wireless IP cameras (1.3 MP EDiMAX camera with night vision support) with a processing rate of 10 fps. This system can handle up to eight cameras on a GPU platform (NVIDIA GeForce GTX 480) with frame rate of 25fps.

Keywords- Multi-camera, surveillance, homography, handshaking, occlusion.

I. INTRODUCTION

Surveillance system helps to monitor a given area of interest. Multiple cameras are used to cover a large area. In order to track objects successfully in multiple cameras, one needs to handshake among objects captured in multiple cameras. The key elements of our proposed surveillance system include change detection, tracking, camera coordination, occlusion handling and suspicious activity monitoring.

Change detection is a basic module of any surveillance system. The detected changes can be considered as foreground objects by modeling the background. Generally, background subtraction and its variants are used to extract the foreground objects from the video stream taken from a stationary camera [1, 2]. However, detecting the foreground objects becomes hard when the background includes variations due to light, shadows and insignificant periodic changes of background objects (e.g. swaying trees). Bayes decision framework is used to extract foreground objects from a real-time complex video [3]. This method consumes

more memory owing to the number of parameters used, thereby increasing the processing time as compared to Gaussian Mixture Model (GMM). GMM framework [4] can overcome these limitations. The value of each pixel is modeled as a mixture of Gaussians. According to the persistence and variance of each mixture of Gaussians, the pixel is determined whether it belongs to foreground or background.

Once an object is detected, tracking is required to estimate its trajectory in the image plane. In other words, a tracker assigns consistent labels to the tracked objects across different frames of a video. Some of the algorithms used for tracking include detection of discontinuities using Laplacian and Gaussian filters; and are often implemented using a simple kernel. These algorithms are simple, but sensitive to noise, and hard to generalize [5]. Fatih Porikli et al. have described tracking using multiple kernels centered at the high motion areas [6]. Other algorithms make use of pattern recognition, such as neural networks, maximum-likelihood and support vector machines. The majority of pattern recognition algorithms require a set of training data to form the decision boundary [7].

Suspicious activity detection is an essential part of effective surveillance system. Real time surveillance system collects large amount of videos and it is important to automate the system to extract useful information about different types of suspicious behavior. Ismail Haritaoglu et al. [1] have described a system that monitors suspicious activities such as depositing an object, exchanging bags, or removing an object etc.

Benjamin et al. [8] presented an overview of a multi camera surveillance system that can automatically extract useful information from a given scene. It also alerts the user if any of the tracked objects breaks certain defined rules. Xiaogang Wang [9] gives an overview of the recent advances in the field of multi camera video surveillance. It compares the existing solutions and also describes the prevalent technical challenges.

Steve Shafer et al. [10] have described prototype architecture and technologies for building intelligent environments that facilitate the unencumbered interaction of people with other people, with computers and with devices. Kiosk [11] is a vision based human sensing application which uses color stereo tracking and graphical output to interact with several users. The system can detect and track the arrival and departure of potential users in the vicinity of the kiosk and can identify individual users during multi-person interaction. Kidrooms [12] is an indoor tracking system where multiple isolated objects can be tracked in real-time.

TABLE I. COMPARISON OF THE PROPOSED SYSTEM WITH PREVIOUS WORK IN TERMS OF COVERAGE AND FUNCTIONALITY

System	Area Indoor(I) Outdoor(O)	Sensor Color(C) Grayscale(G)	Camera Single (S) Stereo (O) Multiple (M)	Detection Single Gaussian (S) Bimodal (B) Mixture of Gaussian(M)	People Tracking Single Isolated (S) Multiple Isolated (M) Multiple in Group (G)
EasyLiving[10]	I	C	M,O	S	M
S.Kiosk[11]	I	C	S,O	S	S
Kidsroom[12]	I	C	M	S	M
M.Mirror[13]	I	C	O	S	M
Pfinder[14]	I	C	S	S	S
TI [15]	I	G	M	S	S
SRI [16]	I	G	M	S	S
MIT [4]	O	C	M	M	M
CMU [17]	O	C	M	S	M
W ⁴ [1]	O	G	S	B	M,G
LOTS [18]	O	G	S	S	M
Proposed Approach	I,O	C	M	M	M

Darrell et al. [13] presents a real time system for tracking people; stereo processing is used to isolate people from other objects and background. Color and face detection modules are also included to make the system robust. Pfinder [14] works when there is only a single person in the field of view of camera. Silhouettes are used to detect human body and 2D contour shape analysis is used to identify the head, hands and feet locations. TI's system [15] uses change detection to recognize moving objects. The detected objects are tracked using first-order prediction and nearest neighbor matching. Events are recognized by applying predicates to the graph formed by linking corresponding objects in successive frames. SRI's system [16] uses stereo information for continuous detection and intensity image correlation for tracking. The system provides information for 3D detection and tracking, even in the presence of crowded scenes, blurred objects, and large-scale changes. MIT's system [4] uses GMM to model the background and tracks people in indoor environment, people and cars in outdoor environment, fish in a tank etc. CMU [17] uses temporal differencing to detect moving objects and these objects are classified as human, vehicle or background by using template matching. This system rejects background clutter and continuously tracks intended objects despite occlusion and appearance changes. Ismail Haritaoglu et al. [1] have explained a real time visual surveillance system for detecting and tracking multiple people and monitoring their activities in an outdoor environment. LOTS [18] suggested background subtraction followed by connected component labeling to detect multiple isolated moving bodies. It can detect people even in case of camouflage. The proposed approach uses multiple cameras to track multiple isolated people in indoor as well as outdoor environment.

This paper presents a set of techniques integrated into a low-cost PC based real-time visual surveillance system, for simultaneously tracking people, and monitoring their activities. The rest of the paper is organized as follows. Section II describes the system setup. Section III gives detailed description of the methodology used in the proposed system. This is followed by conclusion and future work in Section IV and V.

II. MULTI-CAMERA SETUP

Table 1 shows the comparison of the proposed multi-camera surveillance system with the systems existing in literature. In the proposed approach, cameras are set up in such a way that there is a significant overlap between their field of view (FOV).

In our experimental set up, a minimum of 30% overlap is maintained. The cameras can be placed at different heights and orientations. Fig. 1 gives an overview of the setup for the proposed multi-camera surveillance system.

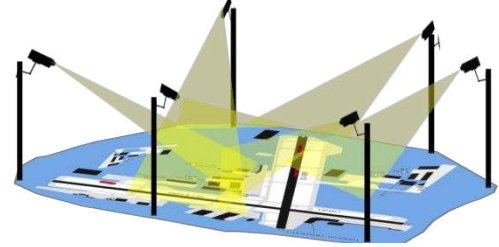


Figure 1. Multi-camera set up

A. Association between cameras

The proposed system deals with multiple cameras. Each camera has its own local coordinate system. In order to establish an association between the objects in different cameras, the coordinate system of these cameras needs to be associated.

Any two camera views can be associated using an appropriate transformation matrix. In this paper, homography-based transformation is used for establishing the association.

Projective transformations or homographies have eight degrees of freedom. The transform can be represented by

$$X_0 = HX \quad (1)$$

Where, $H = [H_s \parallel H_A \parallel H_P]$, X represents the coordinates in camera view 1 and X_0 represents the corresponding coordinates in camera view 2. Here H_s represents a similarity transformation, H_A represents an affinity and H_P represents a projective transformation [19].

The Direct Linear Transform (DLT) algorithm is used to solve the homography matrix. Consider two images for

which homography matrix have to be calculated. The relation between two corresponding points $(x \ y \ 1)^T$ and $(u \ v \ 1)^T$ in two images can be written as:

$$c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2)$$

Where c is any non-zero constant and

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \quad (3)$$

Subtracting first and second rows from third row in (2) yields:

$$-h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)u = 0 \quad (4)$$

$$-h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)v = 0 \quad (5)$$

Equations (4) and (5) can be written in matrix form as:

$$A_i h = 0 \quad (6)$$

$$\text{Where } A_i = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & xu & yu & u \\ 0 & 0 & 0 & -x & -y & -1 & xv & yv & v \end{pmatrix}$$

$$\text{And } h = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9)^T$$

A minimum of four corresponding points is sufficient to solve for the coefficients of the homography matrix. Since each point correspondence provides two equations, four correspondences are sufficient to solve for the eight degrees of freedom of H . In this paper, singular value decomposition (SVD) is used to solve the homography matrix.

B. Field Of View(FOV) Lines

In order to establish the relation between different camera's field of view (FOV), the FOV of a given camera is marked in the overlapping FOV of other cameras. The FOV of a given camera can be represented by an area bounded by a maximum of four lines in any other camera's FOV.

Considering a setup of two cameras, the following steps illustrate the methodology to obtain FOV lines using the homography matrix. H_{12} represents the homography transformation from camera 1 to camera 2 and H_{21} represents the homography transformation from camera 2 to camera 1. The matrices H_{12} and H_{21} can be calculated as explained in section II-A.

The FOV of a given camera can be defined using a rectangular region as shown in Fig. 2. The FOV lines of camera 1 are denoted by L_{11} , L_{12} , L_{13} , L_{14} and for camera 2 by L_{21} , L_{22} , L_{23} , L_{24} respectively. The projection of L_{11} , L_{12} , L_{13} , L_{14} lines in camera 2's FOV are denoted by L'_{11} , L'_{12} , L'_{13} , L'_{14} .

Extracting FOV Lines:

1. Consider the co-ordinates of the points on the boundary of camera 1 i.e., on lines L_{11} , L_{12} , L_{13} , L_{14} as shown in Fig. 2.

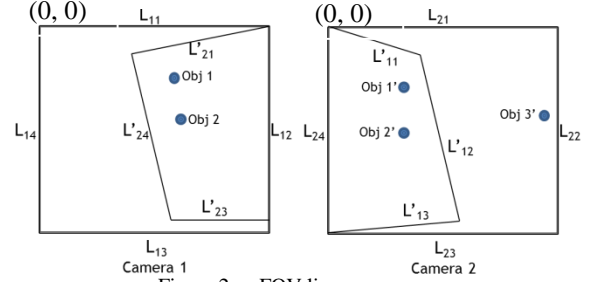


Figure 2. FOV lines

2. Transform the boundary co-ordinates of camera 1 using H_{12} transformation matrix.
3. The transformed co-ordinates result in the lines L'_{11} , L'_{12} and L'_{13} . L'_{14} is not represented because the transformed coordinates do not lie within the limits of the size of the image.
4. Repeat steps 1, 2 and 3 to obtain the FOV lines of camera 2 on camera 1.

III. METHODOLOGY

Fig. 3 provides an overview of the proposed multi-camera surveillance system.

A. Change Detection

The change detection module detects and separates the foreground changes from the background in a given frame. A background model corresponding to each camera view is built using the K-Gaussian mixture models (GMM) [4]. In the background model, recently observed values of each pixel of gray image are retained in the form of mixture of Gaussians based on their weights (ω) and are arranged in ascending order of the value of ω/σ , where σ is standard deviation. Value of each pixel (X_t) of a frame at time 't' is checked against K-Gaussian mixture models. If match found, then its mean (μ_t) and variance (σ_t^2) are updated using following equations:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (7)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (8)$$

Where,

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k) \quad (9)$$

α is learning rate and η is Gaussian probability density.

In addition, weights corresponding to K- Gaussian distributions are updated using following equation:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \quad (10)$$

The time taken to adapt a foreground object as a background object depends on the learning rate (α). The value of learning rate is taken depending on the scene on which GMM is to be applied. Morphological operations are performed on foreground objects, which are obtained

through GMM. Finally, the objects are segmented using connected component labeling as explained in Fig. 4. The system handles variations due to light and eliminates shadows and insignificant periodic changes of background objects.

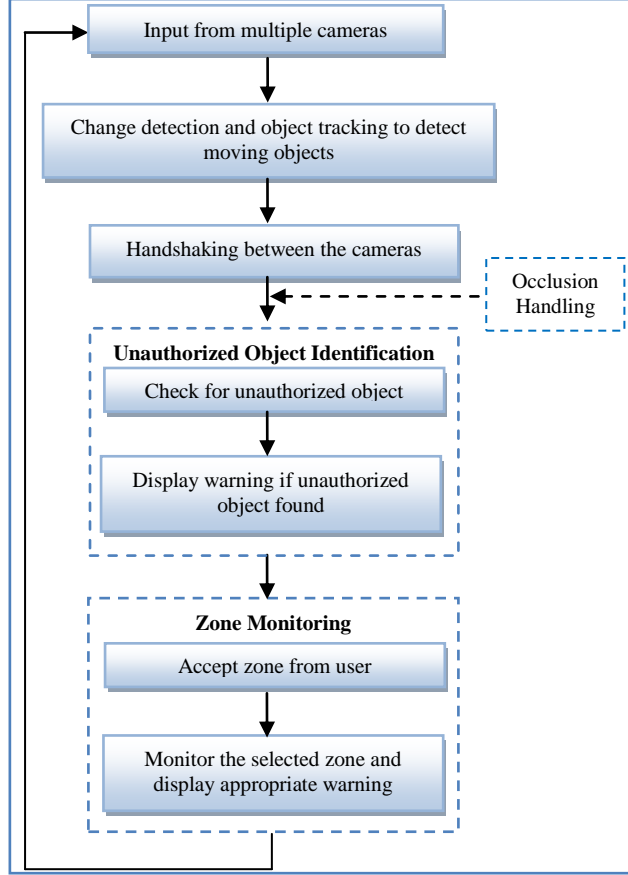


Figure 3. System-level block diagram

B. Tracking

A tracking algorithm is used to establish correspondence between the changes detected in consecutive frames. In this paper, particle-filtering method with histogram as a feature is used to achieve correspondence [20]. The algorithm retains changes (i.e. blobs) of previous frame in the form of a data structure called “state”. The elements of ‘State’ are histogram, position, center, height, and width of a changed blob. For each change, using the particle filtering approach, we consider certain regions in the neighborhood as probable new states. For instance, if there is only one change in the previous frame with center (13, 21), height 10 and width 6 then we can consider a region with center (12, 18), height 10 and width 6 as the probable state. In the proposed approach, Bhattacharya distance measure is used to compute the similarity between the histogram of any given change in the current and previous state.

$$\text{Bhattacharya distance (D)} = \sum_{u=1}^m \sqrt{p^{(u)} * q^{(u)}} \quad (11)$$

$$\text{weight} = (1 / \sqrt{2\pi\sigma}) * e^{-D^2 / 2\sigma^2} \quad (12)$$

Here, p and q are histograms of current and previous state of any given change respectively.

σ^2 = variance of previous state of a change

For each change, weighted mean of centers of its corresponding new probable states is considered as center of its new state. Histogram of the new state is computed using following equation where α is the learning rate.

histogram of new state = $[\alpha * \text{histogram of previous state}] + [(1 - \alpha) * \text{histogram of new state}]$

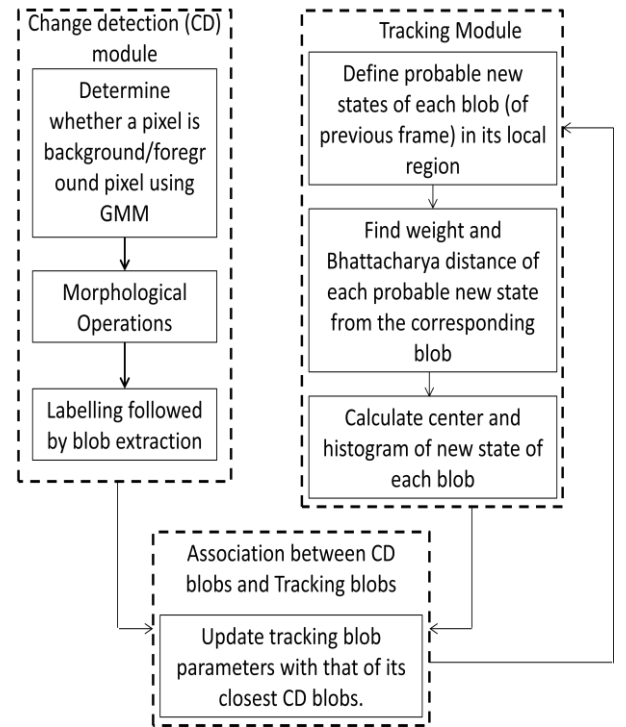


Figure 4. Change Detection and Tracking

C. Handshaking

In multi-camera environment, an object might be in the FOV of one or more cameras. The identity of the object needs to be consistent across the cameras, which is achieved through handshaking. A particular point in camera 1 is in the overlapping FOV of camera 2 if the homography transformation of that point in camera 2 yields a co-ordinate which is within the size limits of the given image. Whenever, a new object enters the overlapping FOV of a camera, all the other cameras having overlapping FOV region with the current camera are checked for handshaking. We have implemented two variants for handshaking as described in section a) and b).

Our Approach for handshaking:

- a) As shown in Fig. 5, the result of transformation of object 1' in camera 2 is H(obj 1') in camera 1. Object 1 in camera 1 is nearest to H(obj 1'). Hence, object 1 in camera 1 is the probable candidate to be associated with object 1' in camera 2.
- b) The distance between the center of the object and FOV lines is used in this section [21].
 - Consider that an object enters the FOV of camera 2 through line L_{24} (as shown in Fig. 5). The corresponding line L'_{24} of camera 1 should be taken into consideration for further calculations.
 - Consider the possible scenarios listed below:

Single object entering the FOV:

- i. The perpendicular distance of the objects that are in the overlapping zone of camera 1 and 2, in camera 1 from L'_{24} i.e., D_1 and D_1' is calculated as shown in Fig. 5.
- ii. The object with the minimum perpendicular distance is the probable candidate to be handshake with the new object i.e., object 1' have to be handshake with object 1 (Fig. 5).

More than one object entering the FOV:

- i. If more than one object enters the FOV at the same time, the perpendicular distances might be similar. In order to overcome this situation, an additional distance is calculated from another point on the FOV line.
- ii. If P_1' is equal to P_1 (Fig. 5), the distance P_2' and P_2 are compared with distance D_2' and D_2 .
- iii. $P_2' > P_2$ and $D_2' > D_2$ implies that object 1' and object 2' have to be handshake with object 1 and object 2 respectively.

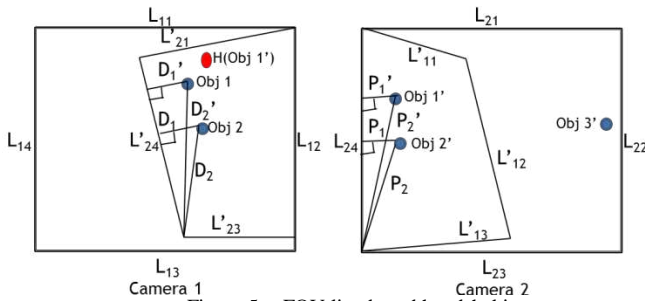


Figure 5. FOV line based handshaking

Fig. 6 shows the output after performing handshaking.

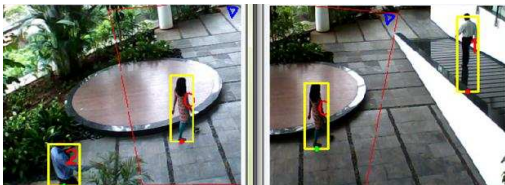


Figure 6. Result of handshaking

D. Occlusion Detection and Handling:

In real-time scenarios, where multiple objects are being monitored using surveillance systems, quite often overlap of some of these objects is observed. This is termed as occlusion. Consequently, there is loss of information of occluded objects. In order to prevent loss of information in such scenarios, we need to detect and handle occlusion.

Fig. 7(a) shows the situation before occlusion where two persons have different identities. However, during occlusion two persons are detected as a single object with same identity as shown in Fig. 7(b).

Occlusion can be detected by monitoring the difference between centers of any two objects in a frame using following equation:

$$X_{\text{difference}} = \left(\frac{W_1}{2} + \frac{W_2}{2} \right) - |cx_1 - cx_2| \quad (13)$$

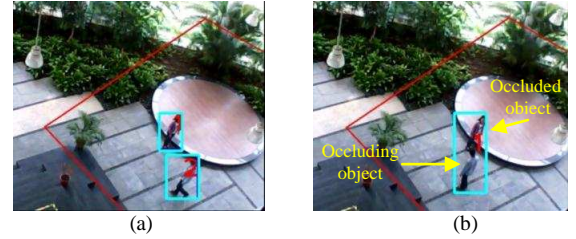


Figure 7. (a) Before occlusion (b) Situation of occlusion

$$Y_{\text{difference}} = \left(\frac{H_1}{2} + \frac{H_2}{2} \right) - |cy_1 - cy_2| \quad (14)$$

Where,

W_1, H_1 = width and height of first object respectively.

W_2, H_2 = width and height of second object respectively.

(cx_1, cy_1) = center of first object

(cx_2, cy_2) = center of second object

If $X_{\text{difference}}$ and $Y_{\text{difference}}$ are positive then we say that occlusion is detected and objects' information is stored. In an occlusion scenario, there are two types of objects as shown in Fig. 7(b) occluding object (the object which is clearly visible), occluded object (the object which is partly or completely invisible). Assume that the origin is at the top left corner of the image. The occluding object always has higher value of y-coordinate than that of the occluded object in the image (Because occluding object is near to the camera than the occluding object). Thereafter, percentage overlap between the two objects which are in occlusion is calculated using the following equation.

$$\text{Percentage overlap} = \frac{X_{\text{difference}} * Y_{\text{difference}}}{W_1 * H_1} \quad (15)$$

If the percentage overlap is less than 50%, then the objects are tracked separately otherwise those objects are tracked together considering them as a single object. The objects which are tracked together would be tracked separately again when percentage overlap becomes less than 50% using the information that we stored when occlusion was detected for the first time. Fig. 8 shows the situation

where occlusion is resolved and two persons get back the same identity which they had before occlusion.

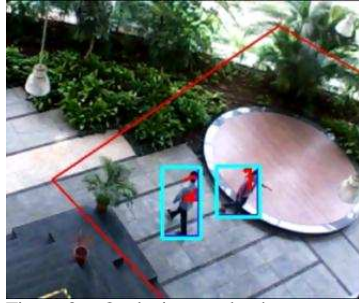


Figure 8. Occlusion resolved

E. Suspicious activity monitoring:

Merely tracking people continuously might not suffice in most of the surveillance systems. There might be some restricted events that have to be prevented from happening. This can be achieved by defining the restricted events at the first place. In case any such event is detected, a suitable audio/visual warning is issued.

1) Unauthorized object detection:

Whenever a person tries to leave a baggage/object, two separate objects are originated from a single object in the consecutive frame.

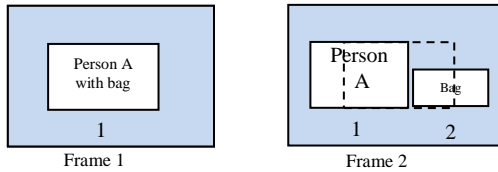


Figure 9. Splitting Condition

Fig. 9 illustrates the condition of splitting that is used in detection of unauthorized baggage.

The following steps illustrate the methodology:

1. Detecting the splitting of a single object into two objects [22].
2. Monitoring the split objects continuously to distinguish between the stationary object and the object in motion.
3. The stationary object is tagged as suspicious with the identity of the object in motion.

The situation of splitting of a single object into two objects also occurs in case of multiple persons moving together. This can be resolved using the fact that a suspicious object is in stationary condition. Fig. 10 shows that the person has left a baggage and is detected as suspect.



Figure 10. Unauthorized object detection

2) Restricted zone monitoring:

In certain high-security zones, there might be some areas where entry is prohibited. These areas have to be monitored continuously and an alarm is to be raised in case of trespassing.

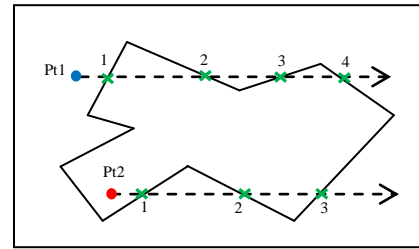


Figure 11. Detecting a point in polygon

As shown in Fig. 11, consider a zone of any arbitrary shape. Let us take any point Pt1 outside the zone. Draw a ray starting from point Pt1 and going in any fixed direction and calculate the number of times it intersects the edges of polygon. From Fig.11 it is clear that if point is inside the polygon, it intersects the edges of polygon for odd number of times. While if point is outside the polygon, number of intersection points are even. Based on this condition, it can be detected whether point is inside or outside the polygon. Fig. 12 explains the steps followed for zone monitoring.

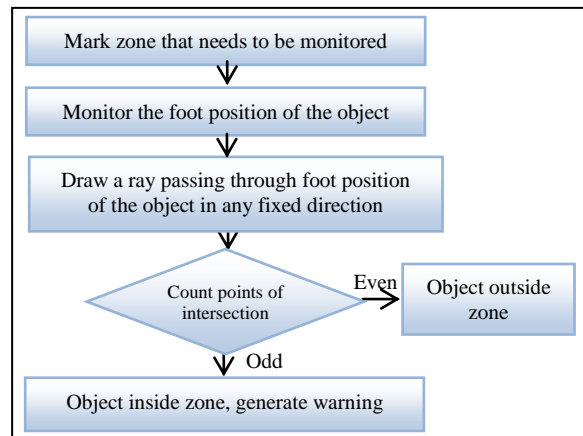


Figure 12. Flow chart of restricted zone monitoring

Fig. 13 shows the warning when person enters inside the restricted zone.

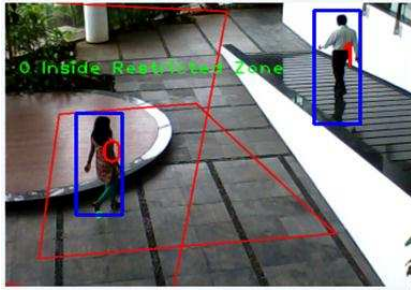


Figure 13. Person inside the restricted zone

IV. CONCLUSION

We have developed an intelligent real time multi-camera surveillance system that can detect and track people. This framework, by virtue of its simplicity, allows tracking people who are trespassing restricted zones and detect suspicious objects that are left unattended. In order to associate objects in multiple cameras, homography based transformation is used. Changes are detected using Gaussian Mixture Model (GMM) and tracking is done using particle filtering. Both the outputs are passed to the association module. This module links the change detection blob with corresponding tracking blob. The proposed system is also capable of detecting and handling occlusion.

The proposed system is implemented on CPU platform comprising of Intel dual core with 4 GB RAM on a 320*240 resolution image accessing four IP cameras simultaneously with a frame rate of 10fps. The processing time is approximately 100ms per instance. The system is also implemented on GPGPU (NVIDIA GeForce GTX 480) to improve performance of the system. It takes 20-40ms per instance at 25fps on a resolution of 320x240 pixels for eight camera inputs.

V. FUTURE WORK

The proposed framework allows an object to be viewed in multiple cameras, thereby providing multiple views of the same object. This would facilitate in creating a best view of any particular object by using specified characteristics or parameters of the object such as size, shape, resolution, etc. Moreover, by using pan-tilt-zoom (PTZ) camera, in addition to the proposed framework, a person detected as suspicious, can be tracked by the PTZ camera; which can zoom in to take a high-resolution image of the person's face. Other suspicious activities like a person suddenly changing his/her direction or group of persons moving in haphazard motion can also be detected. Implementation of pipelining architecture would further improve the system performance above 25 fps on the GPGPU platform.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, VOL. 22, NO. 8, pp-809-830, August 2000.

[2] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):747-757, August 2000.

[3] Liyuan Li, Weimin Huang, Irene Y.H. Gu and Qi Tian, "Foreground Object Detection from Videos Containing Complex Background," *Proceedings of the eleventh ACM international conference on Multimedia*, Berkeley, CA, USA, November 02-08, 2003.

[4] Chris Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[5] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Prentice-Hall, Inc., second edition, 2001.

[6] Fatih Porikli, Oncel Tuzel, "Multi-Kernel Object Tracking," *IEEE International conference on Multimedia and Expo(ICME)*, pp. 1234-1237, July 2005.

[7] Peter E. Hart, Richard O. Duda and David G. Stork, *Pattern Classification*, Wiley & Sons, Inc., second edition, 2001.

[8] W. Benjamin Justin, L. Post Benjamin, K. Estes Kyle, L. Milbert Randy, "TENTACLE: Multi-Camera Immersive Surveillance System", *Small Business Innovative Research (SBIR) Phase I Report*, Air Force Research Laboratory, 2011.

[9] Xiaogang Wang, "Intelligent Multi-Camera Video Surveillance: A Review", *Pattern Recognition Letters* 00 (2012), pp. 1-25, 2012.

[10] A. Shafer, J. Krumm, B. Brumitt, B. Meyers, M. Czerwinski, and D. Robbins, "The New EasyLiving Project at Microsoft," *Proc. DARPA/NIST Smart Spaces Workshop*, 1998.

[11] J. Rehg, M. Loughlin, and K. Waters, "Vision for a Smart Kiosk," *Computer Vision and Pattern Recognition*, 1997.

[12] A. Bobick, J. Davis, S. Intille, F. Baird, L. Cambell, Y. Irinov, C. Pinhanez, and A. Wilson, "Kidsroom: Action Recognition in an Interactive Story Environment," *Technical Report 398*, M.I.T. Perceptual Computing, 1996.

[13] T. Darrell, G. Gordon, M. Harville, J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *Computer Vision and Pattern Recognition*, 1998.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. Pattern Analysis and Machine Intelligence* vol. 19, no. 7, July 1997.

[15] T. Olson and F. Brill, "Moving Object Detection and Event Recognition Algorithms for Smart Cameras," *Proc. DARPA Image Understanding Workshop*, pp. 159-175, 1997.

[16] D. Beymer and K. Konolige, "Real-Time Tracking of Multiple People Using Stereo," *Proc. IEEE Frame Rate Workshop*, 1999.

[17] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving Target Detection and Classification from Real-Time Video," *Proc. IEEE Workshop Application of Computer Vision*, 1998.

[18] T. Boulton, "Frame-Rate Multibody Tracking for Surveillance," *Proc. DARPA Image Understanding Workshop*, 1998.

[19] Elan Dubrofsky, "Homography Estimation," 2009.

[20] Katza Nummiaro, Esther Koller-Meier, and Luc Van Gool, "An Adaptive Color based Particle Filter," *Elsevier Science*, September 2002.

[21] Sohaib Khan, Omar Javed, and Mubarak Shah, "Tracking in Uncalibrated Cameras with Overlapping Field of View", *Proc. 2nd IEEE Int. Workshop on PETS*, 2001.

[22] Tao Yang, Stan Z. Li, Quan Pan, and Jing Li, "Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 970-975, 2005.