# Development of New Algorithm for Detection of Terrorists-Robbers-Lost People and Suspicious People in Public Places

## SUMMARY/ ABSTRACT

In the area of computer vision, tracking human in public places using surveillance cameras is the present topic of research. The first step in any human tracking algorithm is to detect humans. Color and shape based automated human detection in surveillance video is presented. The surveillance video is first divided into individual frames and each frame is divided into three color components. In each component, background is modeled by taking the average of the corresponding past frame components. Individual backgrounds are removed from each color component and three binary images are retrieved. All the three binary images are united to get a composite binary image, which consists of clusters of white pixels called blobs. Depending on the width and height of each blob from its centroid, the blob is categorized whether it is human or non human. The other moving objects can be eliminated by taking the aspect ratio of the blob in to consideration. The top point from the centroid of the human blob is head point and the bottom point is the foot point. With these head and foot points, rectangles are formed with the aspect ratio of 1:3. These rectangles are given as input to the human tracking algorithm.

Three datasets namely AITAM1 (simple), AITAM2 (moderate) and AITAM3 (complex) are developed for testing of the algorithm. The persons in the datasets are combination of terrorists, robbers, suspicious people and lost people. Separate datasets for these four groups are also developed. The new algorithm Gaussian Mixture Model is verified on these datasets along with the standard dataset namely PETS 2009.

The rectangle regions extracted from the human detection algorithm are divided horizontally into four regions. HOG features of the top part are extracted separately and are given to the SVM classifier for face recognition. Based on the existing datasets of terrorists, robbers and lost people, the detected human is classified as whether the person is terrorists, robber, suspicious person or lost person.

## CHAPTER 1

# INTRODUCTION

## 1.1: MOTIVATION

Now a days, every public place like railway stations, bus stations, cinema halls, hospitals, grocery stores, air ports and even now the educational institutions are under surveillance cameras. We see every where that "you are under surveillance". If we can design a system where tracking is embedded in to the system, it may be useful to solve the problem of identification of persons up to some extent. Rather than simply recording the data, the system also tracks the object, if the object is in the vicinity of camera vision. Now the question may arise that what is the use of tracking the object unnecessarily without knowing the utility of tracking. To solve the above identified problems, it is desirable to design an intelligent video surveillance system, where it not only tracks the object but also gives the features of the object. The decision can be taken based on the extracted features of the object whether tracking the object to be continued or not.

## 1.2: IMAGE PROCESSING

An image is an array, or a matrix, of pixels (picture elements) arranged in columns and rows. In a (8-bit) grey scale image each picture element has an assigned intensity that ranges from 0 to 255. The name emphasizes that such an image will also include many shades of grey. The RGB color model relates very closely to the way we perceive color with the R, G and B receptors in our retinas. RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics.

## 1.3: COMPUTER VISION

The goal of computer vision is to let computers understand a scene using camera. In recent years, video analysis is becoming one of the most popular research in the area of computer vision. At any given moment, thousands of videos are uploaded to the database and each year, millions of surveillance cameras around the world are capturing trillions of hours of video. It is impossible for humans to process such large amounts of video data. Therefore, efficient computer vision algorithms for human detection, tracking, segmentation, action recognition, video retrieval, abnormal event detection and video summarization are becoming increasingly important. Object detection and tracking are the most fundamental tasks in computer vision. They have been widely used in many applications such as security and surveillance, human-computer interaction, video communication and compression, augmented reality, traffic control, and medical imaging. Object detection is the process of detecting the instances of a certain class of objects (e.g. humans, dogs, and bicycles) in images and videos. Object tracking is the process of locating moving objects in different frames of a video while maintaining the correct identities. In the context of video surveillance, object detection and tracking are usually coupled together to locate the objects of interest through the video. We need to first detect pedestrians in each video frame, and then track them across different frames. In this dissertation, we focus on the object class of humans, since humans are most likely to be the object of interest in applications such as visual surveillance, human computer interaction, and autonomous vehicle navigation. Object segmentation is also an important task in computer vision. Object Segmentation is the process of delineating the target object from the image. Human segmentation can benefit many computer vision applications, such as person recognition, pose estimation, body part tracking, motion analysis, action recognition, and autonomous driving system.

### 1.4: HUMAN DETECTION

Human detection for applications like video surveillance, autonomous driving vehicles, person recognition has become important. Human Detection is challenging because everyone is different in appearances and there are wide range of poses. There should be a robust method for feature extraction even when the background is cluttered. The cameras used for these applications make use of RGB cameras during night when there is deficiency of light and the images are not clear. This makes the changes in lighting conditions an important point as well. For this, many researchers have proposed different methods for detecting humans from any image. Different methods for extraction of features have been developed so that the features can be applied to SVM for classification. One of the proposed methods is Histogram of Oriented Gradients, which is extensively used in detection of human beings. The initial step towards prediction and analysis of the intention and behavior of human is their detection. Although detection of humans is a crucial work for many computer vision applications like gesture recognition, human tracking, video surveillance and action recognition, the time required for the task has always been a large overhead for real time processing. Researchers have proposed many methods for human detection from images and also from video frames. Lot of feature extraction algorithms have also been developed which can be applied to classifiers like SVM and others. This is extensively used for detection of human beings. There are two steps for human detection process. First is object detection and the second is classification. The first step object detection could be performed by optical flow, background subtraction and spatio-temporal filtering. Background subtraction is a most popular technique for moving object detection where it identifies the moving objects by taking the difference between the current frame and the background frame in a block-by-block or pixel-by-pixel fashion. There are so many methods to perform background subtraction. The most prominent ones are non-parametric background, Gaussian Mixture Model, temporal differencing, hierarchical background and warping background model. The optical flow-based human detection technique uses the moving object flow vector characteristics over time to detect the objects which are moving in the image frames. Though these techniques are susceptible to non-uniform lighting, color and image noise, all the optical flow computation techniques have requirement of computations and are vulnerable to motion discontinuities. In spatio-temporal filter methods which are based on motion detection, the motion is categorized by the 3-D spatio-temporal volume data occupied by the moving human in the image sequence. Process of simple implementation and low computational complexity are the two advantages of optical flow based human detection.

The usage of surveillance cameras is increasing significantly in urban areas which results in analysis of massive amounts of surveillance videos. There are three categories of object classification. Texture based, motion based and shape based. Shape based approaches describe the information of the shape of the motion regions such as boxes, points and blobs. Then it is considered commonly as issue of standard template matching. However, the human body articulation and the differences in the viewpoints observed tend to a huge number of probable appearances of the body, making it hard to distinguish accurately a moving object from other moving humans using this approach. The shape based approach alone cannot distinguish the humans from moving objects. This limitation could be overcome by combining the shape based approach with another thing like color based approaches. Methods based on texture such as Histograms of Oriented Gradients (HOG) use multi-dimensional features use Support Vector Machines for detecting humans.

### 1.5: FACE RECOGNITION

Face recognition [1] aims at identifying the person's distinctiveness by comparing the facial features with the available face data base features. The face data base, with known characteristics, is referred as the face gallery and the input face requiring determining the identity is the probe. One of the problems in face recognition is identification, and the other is the authentication (or verification). Of the two, face identification is more tricky as it cross verifies the gallery completely for minimum variance. As an active topic of research in computer vision, visual surveillance in active scenes attempts to identify, recognize and track objects in moving environment and in specific track human beings. It has extensive potential applications such as traffic surveillance in expressways, security issue in important organizations, to measure the crowd instability in shopping malls, airports and railway stations etc. Significant work has been done in surveillance systems by researchers. To track the human movements constantly, surveillance cameras are located in public places like super markets, grocery stores etc... There is a lot of scope for implementing these algorithms pertaining to the surveillance.

As one of the important research topics in the field of biometrics, face recognition has been favoured by its non-contact and non-stealing characteristics [1, 2]. With the machine learning, pattern recognition, artificial intelligence and computer vision technology continue to develop, face recognition technology has made great progress. The commonly used methods of face recognition are: Support Vector Machine (SVM), Principal Component Analysis (PCA), Histogram of Oriented Gradient (HOG), and so on. Among them, the improved SVM algorithm based on Optimization of kernel function [3] and classification algorithm for face recognition [4], due to the extraction of all features of the image, the computation is large and it is difficult to achieve the rapid identification of huge amounts of data; PCA is the feature extraction of the global feature, although the dimension of the feature is reduced, its recognition accuracy is affected by the illumination [5–7]; HOG method based on feature extraction [8], although not affected by illumination and geometric deformation, the calculation is still large, difficult to achieve rapid identification [9]. At present, most of the face recognition methods are based on restrictive conditions (illumination, gesture, expression and other specific circumstances) of the data classification and identification [10], and non-restrictive conditions (illumination, gesture, expression and other real state) of the face recognition is a difficult problem [11]

Face recognition methods mainly deal with images which are of large dimensions. This makes the task of recognition very difficult. Dimensionality reduction is a concept which is introduced for the purpose of reducing the image dimensions. PCA is the most widely used dimensionality reduction and also for subspace projection. PCA can supply the client with a lower-dimensional picture, a projection of this object when seen from its informative view point. This can be achieved by taking only the starting few principal components in such a way that the dimension of the transformed data is minimized. The linear combinations of pixel values here in PCA are called Eigen faces. PCA is an unsupervised and it ignores all the class labels. It treats the entire data as a whole. It uses SVD for dimensionality reduction.

### 1.6: HUMAN TRACKING

Implementation of human tracking functionality in computer vision systems has attracted researchers' interest for decades. This is because there are numerous applications in which human tracking is important. They range from civilian applications (human computer interaction, robotics, surveillance systems, crowd sourcing systems) to military applications. Tracking of humans [12] in video sequences is a particularly difficult task due to the following reasons: – varying illumination of the monitored scene, – loss of depth information in mono-camera image acquisition systems, – varying size and shape of the tracked human (due to changes in orientation and distance to the camera), – occlusions of the tracked humans, – motion of the tracking camera (i.e. both the tracked human and the background move in reference to the camera). The human tracking task can be subdivided into the three major steps:

1) Human detection in a scene and determining its location (i.e. applying methods for segmenting out the human of interest from the background).

2) Identifying human position changes in consecutive image frames, termed human tracking.

3) Analysis of the human tracking data (e.g. determination of the motion trajectory, path prediction, etc.).

Human tracking is a well-studied problem in robotics and related areas [13], [14]. Given a sequence of frames corresponding to moving crowds, the goal is to extract the trajectory of each human. As autonomous robots are increasingly used in the physical world inhabited by humans, it becomes more and more important to detect, track, and predict the actions of humans [15], [16]. We need real-time human detection capabilities for collision free navigation in dynamic environments. The human trajectories are used to predict future locations of people in order to compute appropriate routes for the robots, such as autonomous cars, mobile surveillance systems, wheelchairs, and museum guides. The problem of tracking objects and humans has been studied for almost two decades and remains a major challenge for crowded scenes [17]. Human tracking remains a challenge in part because humans tend to change their speed to avoid collisions with obstacles and other humans. Finally, in crowded scenes, the pair wise interactions between humans can increase at a super-linear rate. Many approaches have been proposed for online human tracking [18], [19], [20], [21], but they can't provide real time performance as the number and density of humans in the scene increase. Many other fast trackers have been proposed [22], [23], but they only provide good accuracy in some scenes.

Human tracking is used in video surveillance systems like railway stations, airports, any public places like malls and hospitals etc. The challenging and difficult task is to detect and track objects such as people. The difficulty increases as there are so many things need to be taken in to attention in human tracking. The complexity includes appearance of people, clothing, illumination changes and pose changes.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1: INTRODUCTION

Tracking objects in surveillance videos is one of the important ongoing exploration areas in many computer vision applications. The aim of any object tracking system is to follow the target objects through the frames of the video. In today's world, video surveillance system is the integrated part of any organization or even public places to track humans. Detecting, recognizing and then tracking the person in video surveillance is the present topic of research. In this chapter, different object tracking algorithms, human detection algorithms, background subtraction methods, human tracking algorithms, face recognition algorithms proposed by the researchers are presented.

## 2.2: OBJECT TRACKING

Object tracking has been a challenging topic of research in computer vision. Object tracking is defined as estimating the trajectory of an object over the video frames as the object moves around the scene. It has to deal with difficulties such as occlusions, dynamic illuminations, non-rigid motion and changing appearances. Motion detection gives useful information for tracking objects. Tracking requires extra segmentation of the corresponding motion parameters. There are several efforts made by the researchers dealing with the problem of tracking. Existing approaches are classified mainly in to two categories: model-based and motion-based approaches. Motion-based approaches mostly rely on robust methods for aligning visual motion consistencies over time. These methods are fast relatively but have significant difficulties in dealing with non-rigid objects. Model-based approaches also explore the utility of knowledge and high-level semantics of the objects. These techniques are more consistent compared to the tracking based on motion, but they have limitation of high cost of computations for complex models due to need for handle with rotation, translation, deformation and scaling of the objects. In this section, a wide variety of object tracking algorithms and its related research is reported.

**Gongyi Xia et al., (2016)** [5] combined particle swarm optimization with density estimation for object-tracking based on particle filter. There are various algorithms for object tracking based on particle filter. However this algorithm fails when the objects cross their paths. The target track initialization is too expensive and consumes lot of time in this algorithm.

**Bohyung Han et al., (2016)**[14] proposed appearance based human tracking algorithm which is dependent on the similarity between the observation in the image and the existing appearance model. This model is for finding the location of the human and also to identify the person. The algorithm is based on the Kinects (kalman fusion of multi sensors) which are installed at the entrance of the homes for recording purposes. Shapes, body colors and also the multi-view faces are recorded. Here, adapted MSA is performed both on captured image and the database simultaneously. The problem with this model is that there are two independent operations and there is no connection established. This may leads to occasional failures.

**Kalpana Goyal and Jyoti Singhai (2017)** [15] presented various background subtraction algorithms based on Gaussian Mixture Model and compares them on the basis of quantitative evaluation metrics. Their performance analysis is also presented to determine the most appropriate background subtraction algorithm for specific application of video surveillance systems. All the background subtraction algorithms based on Gaussian mixture model mentioned by the authors models each pixel either as a background or the foreground based on the all past pixels. However, it takes more number of iterations for convergence as the number of pixels considered is the entire history.

## 2.3: BACKGROUND SUBTRACTION

In this section, some of the main background subtraction methods addressed by the researchers are presented. The efficiency of the tracking algorithm depends more on modeling the background. To model the background efficiently, external parameters like cluttered background, scene complexity, illumination changes, and motion of static objects like tree branches are to be considered. Starting from modeling the background by simple average of past frames to developing the classifier by taking all other parameters in to consideration is presented.

**Liao S et al., (2010)** [12] proposed a novel background subtraction framework by taking the illumination variations and dynamic backgrounds in to consideration. First, a scale invariant local ternary pattern operator is proposed to tackle illumination variations. Second, pattern kernel density estimation technique is proposed to effectively model the probability distribution of local patterns in the pixel process. Third, multimodal background models with the above techniques and a multiscale fusion scheme for handling complex dynamic backgrounds are developed. However, the proposed techniques are for particular illumination variations and particular dynamic backgrounds, which is not the case in real-time scenarios.

## 2.4: HUMAN DETECTION

The first step in the process of human tracking is to detect the human of interest in the video sequence and to cluster pixels of these humans. A standard video surveillance system attempts to track and identify the interested objects in a video scenario. The classification of moving objects plays a crucial role in developing any tracking system. Considerable amount of research has already been done in the field of video surveillance; with each method make use of its own unique technique of classification. In this section, some of the main human detection techniques addressed by the researchers are presented.

**Komagal et al. (2014)** [16] developed a method for Human detection in hours of darkness (night time) using Gaussian Mixture Model (GMM) in real night environment employing an infrared radiation camera. And this method does not focus on tracking of person under severe conditions in light mode. This method is best suitable for human detection at night. This gives the insight of usefulness of GMM for human detection in video surveillance under different lightening conditions.

**Chen et al., (2014)** [18] developed an efficient method to deal with the face recognition problem. They exploited the property of facial expressions result from facial muscle movements or deformations. HOG is used for facial feature extraction and SVM for classification. The same has been adopted here for face recognition and incorporated in tracking human in surveillance videos. The algorithm is limited to standard facial expressions and not suitable for wild environment.

**Dollár et al. (2014)** [8] developed a new approach which uses a lightly samples image pyramid for feature approximation at intermediate scales. But he did not mention in detail how to confine the range of image scaling.

## 2.5: FACE RECOGNITION

Face recognition is one of the most important topics in the biometrics filed. It has been favoured by its non-stealing and non-contact characteristics. With the pattern recognition, machine learning, computer vision technology, artificial intelligence continue to develop, face recognition made a huge progress. The most common methods of face recognition are: Principal Component Analysis (PCA), Histogram of Oriented Gradients (HOG), Support Vector Machine (SVM), and so on. In this section, some of the face recognition methods addressed by the researchers are presented.

**Dalal et al. (2005)** [6] in his paper developed a human detection algorithm using HOG. These are similar to the features used in SIFT. The features of the HOG are found by taking the orientations of histograms at edges in the local region. This is simply the imitation of how the signals from the eye are processed by the brain. The algorithm is robust for local changes and position. The same is being used in this paper for face feature extraction. However the author is focused on the pedestrian detection and not on the tracking of human over a period of time.

**Conde et al., (2013)** [3] explained that HoGG gives better results when compared with local Gabor feature selection. Recognize the face by extracting face features and tracking the human is surely a challenging task for any video surveillance system. And all the algorithms quoted above are failed to do so either of these two or both of these tasks.

## 2.6: SUMMARY

This survey is made on different object tracking algorithms developed by researchers for various applications. Gaussian mixture model proved to be the best for object tracking in the context of complex scenarios. The number of iterations required by the GMM to make the decision whether the current pixel is foreground or background can be very much minimized by considering only the limited set of past pixels.

Recognizing the human before tracking makes the surveillance system more authentic. Researchers have addressed various algorithms for feature extraction and classification. HOG feature extraction and SVM classifiers have gain lot of interest because of their unique feature set. The face features are most useful when there are identity switches in multiple human tracking problems. By taking the face features into consideration along with tracking, identity switches can be eliminated.

## 2.7: PROBLEM STATEMENT

From the above survey, it is observed that the researchers addressed various object tracking algorithms. It is observed that, among the available tracking algorithms, tracking object by modeling each pixel as a mixture of Gaussians is found to be more accurate. Expectation Maximization (EM) finds the solution parameters namely the mean, variance and the weights of each gaussian function. While modeling the pixels, GMM algorithm takes the entire set of past pixels. Because of this, EM takes more number of iterations to come up with converging solution. In order to reduce the time taken by the EM, only few past pixels with different wieghtages are considered in this work. To overcome the identity switches both in single object and multiple object tracking problems, face features are also considered. The main aim of this work is to develop

***"Development of new algorithm for detection of terrorists-robbers-lost people and suspicious people in public places".***

## 2.8: AIM & OBJECTIVES

### AIM

To detect, identify and classify the people as terrorists, robbers, lost people and suspicious people in public places using surveillance cameras.

### OBJECTIVES

- To develop Color and shape based human detection algorithm for detection of humans while tracking.
- Develop Gaussian Mixture Model based pedestrian tracking algorithm.
- Develop HOG based face feature extractor.
- Based on the existing datasets of terrorists, robbers and lost people, classify the detected human as whether the person is terrorists, robber, suspicious person or lost person.

# CHAPTER – 3

# COLOR AND SHAPE BASED AUTOMATIC DETECTION OF PEDESTRIANS IN SURVEILLANCE VIDEOS

**3.1: INTRODUCTION:** Pedestrian detection is the first step in any human tracking system. There are so many methods for human detection which use background subtraction principle. Background subtraction is the principle behind developing this algorithm.

In this, a combination of color and shape attributes are considered for automatic detection of human in surveillance videos.

**3.2: DATASETS**

For experimental purpose PETS dataset is taken. PETS (Performance Evaluation of Tracking and Surveillance) is a standard datasets which will be released every year. Here PETS 2009 datasets are taken. The same scenario which is of 1 minute 53.43 seconds duration and at a rate of 7 frames per second with a resolution of 576X720 pixels is taken from seven viewpoints. The frames from these videos are named as PETS 1 to PETS 7. Figure 3.1(a) to (l) shows 12 sample frames from PETS 4 dataset. The number of humans in the video are 8. There are isolate humans, partially occluded humans and also fully occluded humans. Both static and dynamic occlusions are present in the video.

(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig: 3.1 (a) – (l) 12 Sample frames from PETS 4 Dataset which contains 794 frames with each frame having the resolution of 576X720 pixels.



(a) (b) (c)



(d) (e) (f)

(g)

Fig: 3.2 (a) – (g) Frame number 15 from PETS Datasets 1, 2, 3, 4, 5, 6 and 7 respectively. All the datasets have same frame count and with same resolution. The frames shown are the same instant of time with different view angles.

Figure 3.2 shows the frame number 15 from all the datasets. The frames are from the same instant of time from different views. Frame number 15 from PETS 1 to PETS 7 are shown in figure (a) – (g). Because of different views, the human count and also the human orientation seem different.

## 3.3: METHODOLOGY

1. Divide the color frames into three individual components (R, G and B) for all the past 100 frames.
2. Find the three color back ground images separately by taking the average of all individual past frame color components.
3. Find the three color components of the current frame and take the difference between the color component and the corresponding color component of the back ground image there by getting three binary images for the three colors.
4. Pass the red component binary image through a median filter of mask size 5X5 in order to remove all the small clusters of white pixels.
5. Find the area of all the clusters and filter out all the small clusters of area size 500 pixels.
6. From the centroid, find the height and width of each cluster.
7. Filter out the non humans by keeping the aspect ratio WXH as 1X3.
8. Fix the bottom point as the foot point and the top point as the head point of the human.
9. Form a rectangle with measured height and width of all the blobs in the frame and call the binary image as A.
10. Repeat the steps from 3 to 9 for green and blue colors and call the binary images as B and C.
11. Find A U B U C and form the binary image.
12. Repeat the steps from 1 to 11 for all the frames there on.

Background images are formed by taking the average of all the frames in the image individually. If the platform if real time, the background images are formed by taking the average of 100 past frames.

## 3.4: EXPERIMENTAL RESULTS

(a)

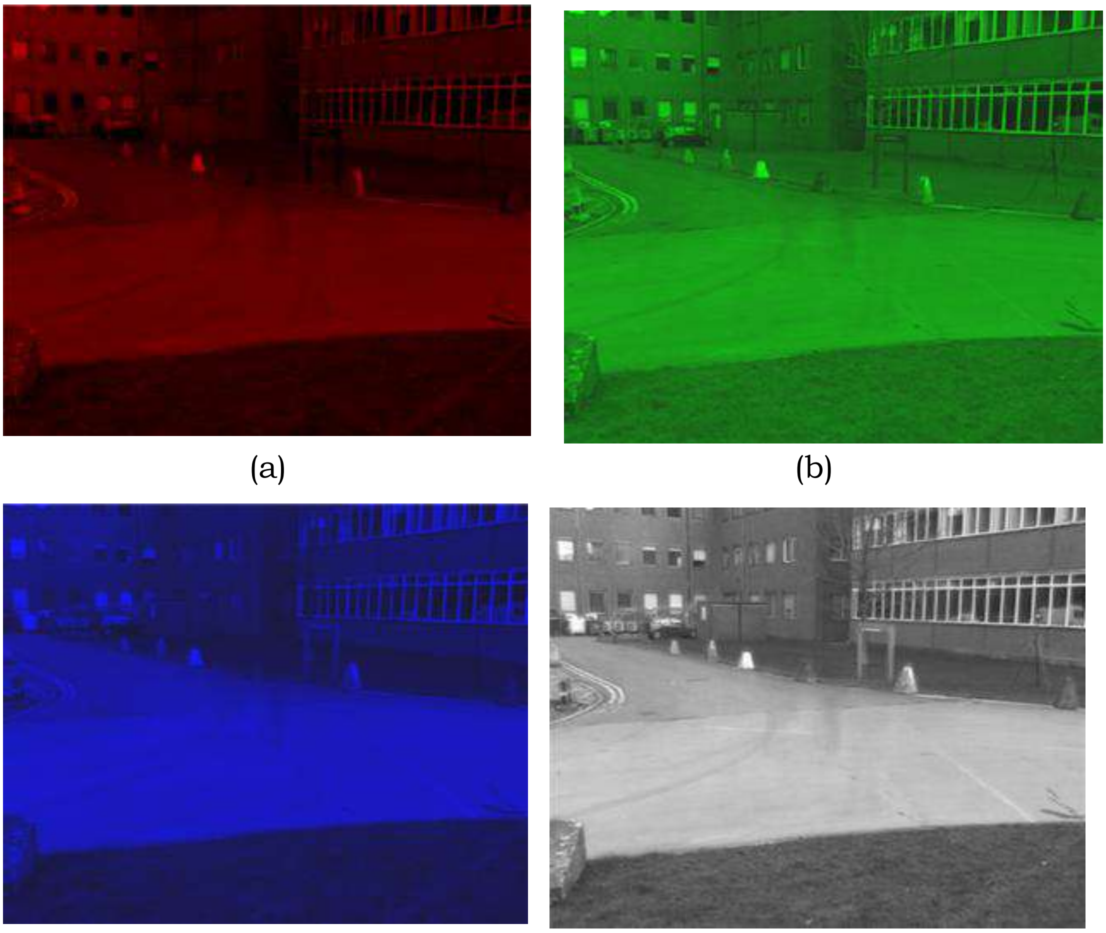
(b)

(c)

(d)

Fig: 3.3 Background modeled image from (a) red (b) green (c) blue and (d) gray components. It is formed by taking the average of all the respective components of all the images.

Figure 3.3 shows the background model images for the three components and also by using the gray images. The background model of red component is formed by taking the average of the red components of all images and similarly for the green and blue background model images.



(a) (b) (c) (d)

Fig: 3.4 Low pass filtered background modeled image from (a) red (b) green (c) blue and (d) gray components.

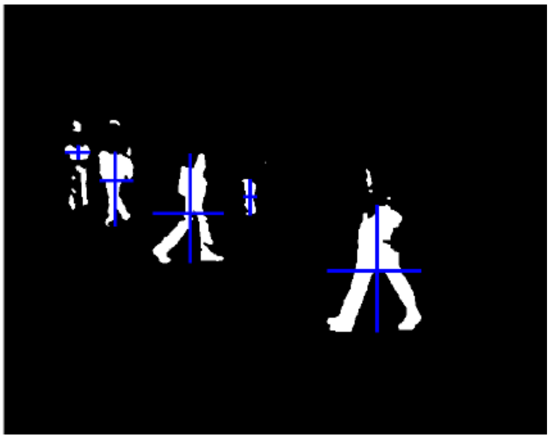
The background model images for different components are passed through the low pass filter in order to remove the shadowlike structures formed because of averaging principle. Figure 3.4 shows the low pass filtered background model images.

(a) (b) (c)

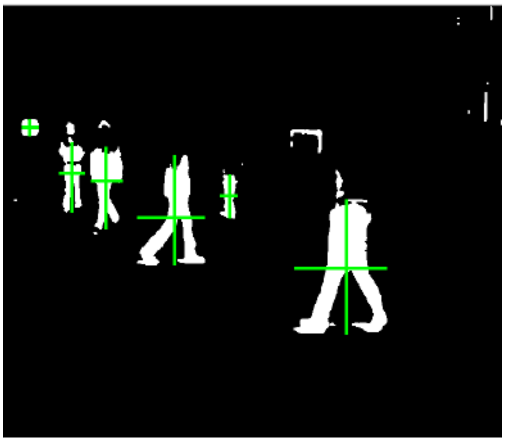
Fig: 3.5 (a) Frame number 500 of PETS 4 dataset; (b) result of background subtraction done on the sample frame number 500 with back ground model gray image; (c) Detection of objects using gray component background subtraction.

Figure 3.5 (a) shows the gray image of frame number 500 from PETS 4 dataset. There is only one human in the image. By frame differencing from the background model image, two objects are found. Figure 3.5 (b) shows the binary image and the detected objects. Figure 3.5 (c) shows the detected objects on the video frame. It is observed that by using the gray level back ground modeling; only a portion of the human has been detected. Since the background of the face part is the green garden, with the gray level background modeling it is not possible to be identified in the difference image. In order to overcome such problems, the color components are separated. The face part may not appear in frame differencing image when it is gray level. But when it is either green or red or blue based the same face part will appear in the difference image. Therefore the color components are separated.

Figures 3.6 shows the width and height measurements based on the blue and green component based background subtraction.

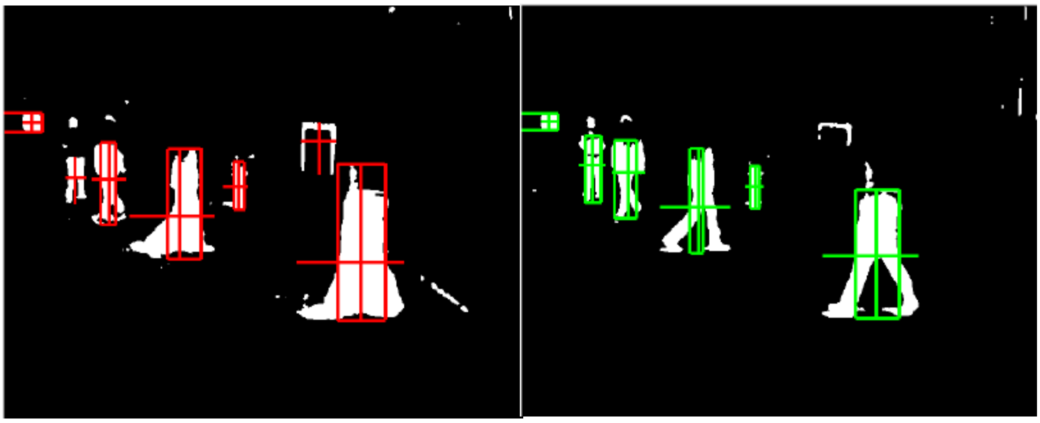


(a)



(b)

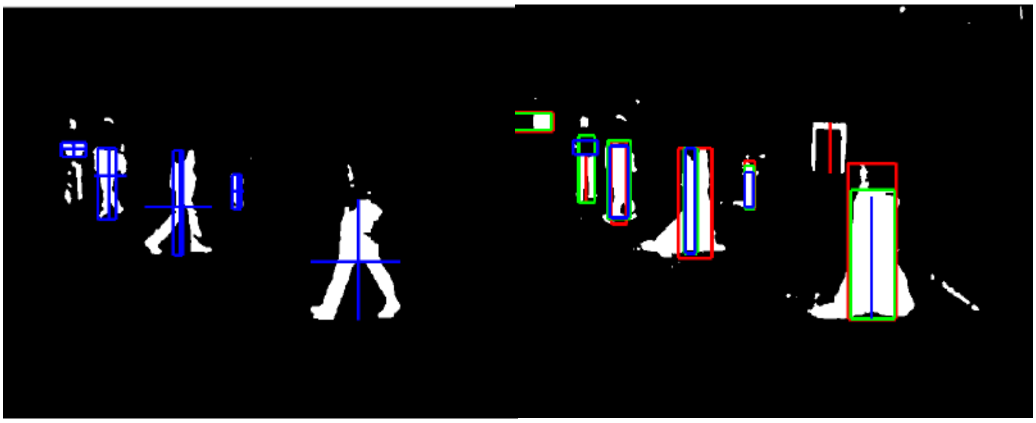
Fig: 3.6 (a) Detection of objects and their height and width measurements using blue component; (b) Detection of objects and their height and width measurements using green component.



(a) (b)

Fig: 3.7 (a) boundary boxes on objects with modified width measurement using red component; (b) boundary boxes on objects with modified width measurement using green component;

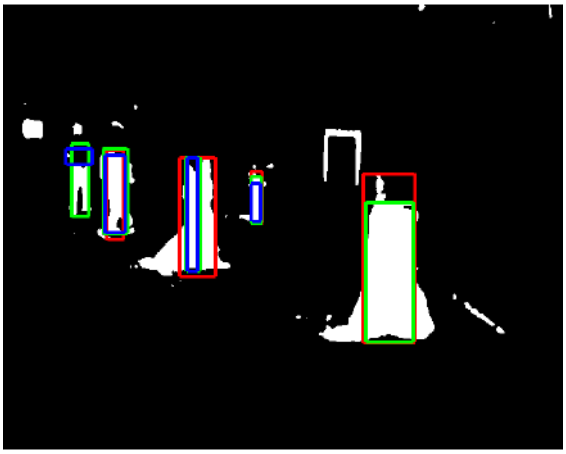
Figures 3.7 show the boundary box formation based on the height and modified width of the object blob of red, green and the blue background models respectively.



(a) (b)

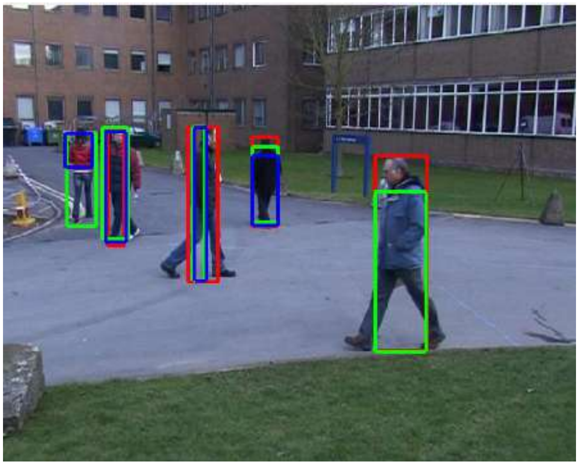
Fig: 3.8 (a) boundary boxes on objects with modified width measurement using blue component; (b) overlapping of all the three bounding boxes on the objects.

Figure 3.8 shows that the overlapping of all the three bounding boxes on the objects. Some blobs have all the three color boxes. Some blobs have only one or two boxes. Here the one with the bigger box is considered. The aspect ratio is calculated for all the bounding boxes. The height must be at least 3 times than the width is considered. The W/H ratio of at least 1/3 is taken. The blobs are filtered out based on this principle. Figure shows only human blobs are detected and others are filtered out as they are not maintaining the aspect ratio.

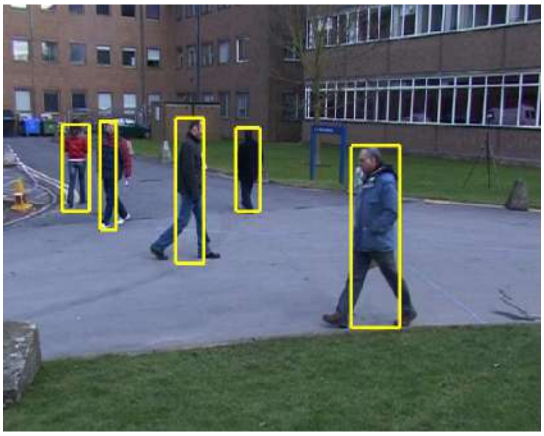


(a)

Fig: 3.9 filtering the humans from the non-humans using aspect ratio criterion.



(a)

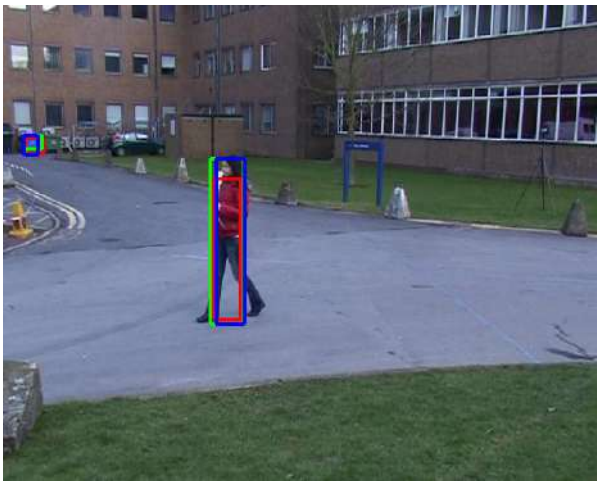


(b)

Fig: 3.10 (a) detection of humans using the combination of three color bounding boxes; (b) Human detection using proposed color and shape based human detection algorithm.

Figure 3.19 shows the overlapping of bounding boxes on the humans. Figure 3.10 shows the human detection algorithm based on color and shape based human detection algorithm.

(a)



(b)

Fig: 3.11 (a) Frame number 500 from PETS 4 dataset; (b) detection of humans using the combination of three color bounding boxes.

Figures 3.11 (a) show the frame number 500 from PETS 4 dataset and figure 3.13 (a) shows the frame number 700 from PETS 1 dataset respectively. Figure 3.11 has only one human and figure 3.13 has 8 humans. Figures 3.11 (b) and figure 3.13 (b) show the overlapping of bounding boxes on the humans. All the humans are detected exactly using this algorithm. The complete human is detected using our proposed algorithm.

Figure 3.13 (b) shows the bounding boxes of all the three colors are formed on each human. The maximum of the bounding box is taken and are shown as yellow boxes in the figure 3.12 and 3.13.

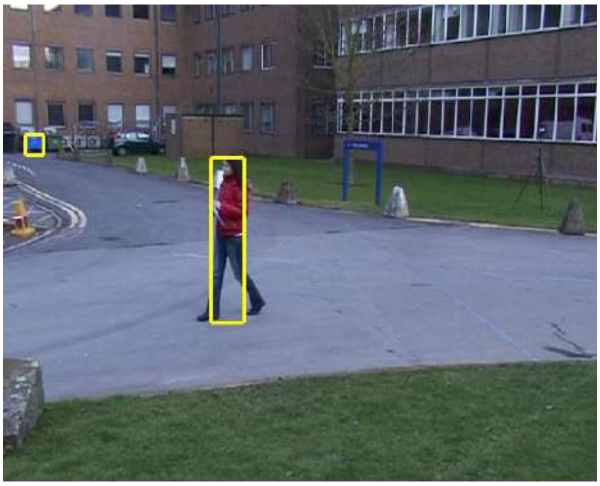
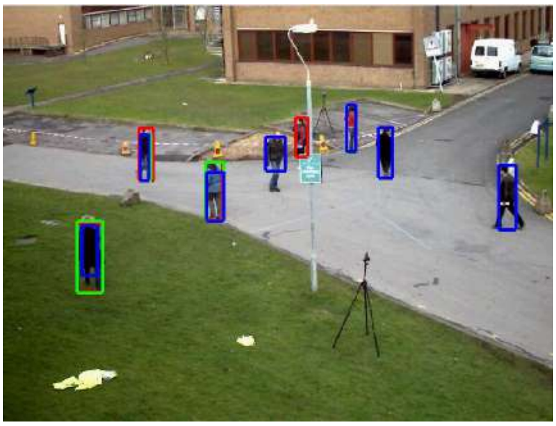


Fig: 3.12 Human detection using proposed color and shape based human detection algorithm.



(a) (b)

Fig: 3.13(a) Frame number 700 from PETS 1 dataset; (b) detection of humans using the combination of three color bounding boxes.

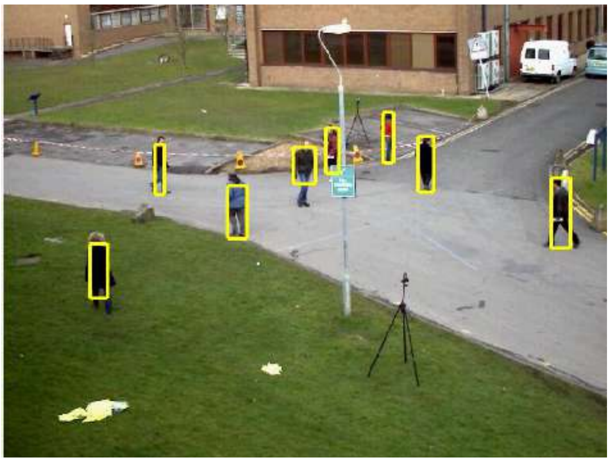


Fig: 3.14 Human detection using proposed color and shape based human detection algorithm.

| | |
|---|---|
| HUMAN DETECTED AS HUMAN | TP |
| HUMAN DETECTED AS NOT HUMAN | FN |
| NOT HUMAN DETECTED AS HUMAN | FP |
| NOT HUMAN DETECTED AS NOT HUMAN | TN |

TABLE 3.1 Test statistics of human detection

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

Recall = True Positives / (True Positives + False Positives).

Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.

Precision = True Positives / (True Positives + False Negatives)

Annotations of PETS 2009 View 1 are calculated manually as:

| Persons | From (Frame No.) | To (Frame No.) | Total Frames |
|---|---|---|---|
| 3 | 1 | 12 | 12 |
| 4 | 13 | 16 | 4 |
| 5 | 17 | 39 | 23 |
| 6 | 40 | 80 | 41 |
| 7 | 81 | 115 | 35 |
| 8 | 116 | 132 | 17 |
| 9 | 133 | 147 | 15 |
| 8 | 148 | 171 | 24 |
| 7 | 172 | 207 | 36 |
| 6 | 208 | 228 | 21 |
| 7 | 229 | 287 | 59 |
| 6 | 288 | 362 | 75 |
| 5 | 363 | 374 | 12 |
| 4 | 375 | 395 | 21 |
| 3 | 396 | 403 | 8 |
| 2 | 404 | 423 | 20 |
| 3 | 424 | 462 | 39 |
| 4 | 463 | 502 | 40 |
| 7 | 503 | 520 | 18 |
| 6 | 521 | 571 | 51 |
| 7 | 572 | 608 | 37 |
| 7 | 609 | 617 | 9 |
| 6 | 618 | 656 | 39 |
| 7 | 657 | 691 | 35 |
| 8 | 692 | 743 | 52 |
| 7 | 744 | 794 | 51 |
| Ground truth | | | 4853 |

TABLE 3.2 ground truth of PETS 4 2009 Dataset

| | |
|---|---|
| True Positives | 4531 |
| False Negatives | 254 |
| False Positives | 118 |
| True Negatives | Not Applicable |

TABLE 3.3 Confusion matrix of human detection test.

The true negatives are not applicable in this experiment because we want to find the human. Therefore there is no question of "not human detected as not human".

| Metrics / Algorithm | Data Set | No. of Persons | No. of frames | Frame rate | Duration | Recall | Precision |
|---|---|---|---|---|---|---|---|
| Automated color and shape based human detection algorithm | PETS4 | 8 | 794 | 25 | 33.08sec | 97.46% | 94.67% |

TABLE 3.4 Performance metrics of PETS 4 2009 dataset

| Metrics / Algorithm | Data Set | False Alarm per Frame (Fa/F) | Ground Truth (GT) | False Positives (FP) | False Negatives (FN) |
|---|---|---|---|---|---|
| Automated color and shape based human detection algorithm | PETS4 | 0.4656 | 4853 | 118 | 254 |

TABLE 3.5 Performance metrics of PETS 4 2009 dataset (Contd..)

### 3.5: CONCLUSIONS

Automatic human detection using combination of color and shape in human tracking surveillance videos is presented in this work. Both color and shape are used in detecting the humans. By adapting the color component in the algorithm, the human can be detected more accurately. By using the shape component and by using the aspect ratio, humans are detected properly. This algorithm can separate out all the non-moving humans. These automated annotations can be used for any human tracking surveillance systems. Experimental results show that humans are detected in PETS 2009 view 1 datasets. The algorithm is having the limitation of not detecting the humans in the case of severe occlusions.

# CHAPTER – 4

# FACE RECOGNITION USING HOG AND SVM IN SURVEILLANCE VIDEOS

## 4.1: HISTOGRAM OF ORIENTED GRADIENTS (HOG)

HOG was proposed by Dalal and Triggs. The Histogram of Oriented Gradients is a popular dense feature extraction method for images. Dense means that it extracts features for all locations in the image or a region of interest in the images. Applications of HOG features were proposed for face recognition, thumb recognition and vehicle detection. As said by Dalal and Triggs to obtain the HOG features, can be implemented by four phases including Gradient Computation, Orientation Binning, Descriptor Blocks and Block Normalization.

### 4.2.1: Procedure of HOG

HOG is a feature extraction algorithm converts an image of fixed size to a feature vector of fixed size. A feature descriptor is a representation of an image or part of an image that simplifies the image by extracting useful information and throwing away extraneous information. Typically, a feature descriptor converts an image of size width x height x 3 (channels) to a feature vector or array of length n. In the case of pedestrian detection, the HOG feature descriptor is calculated for a 64×128 piece of an image and it returns a vector of size 3780. Notice that the original dimension of this image piece was 64 x 128 x 3 = 24576 which is reduced to 3780 by the HOG descriptor. The feature vector is not useful for the purpose of viewing the image. But, it is very useful for tasks like image recognition and object detection. Typically images at multiple sizes are analyzed at many image datasets. To demonstrate this we considered 4.88 MB size dataset and shown in below. From this we have selected a piece of image size with 256 X 256 for calculating HOG feature descriptor.

HOG can be implemented based on four stages include Gradient calculation, histogram of gradients, block normalization and feature vector. The following stages for calculating the HOG descriptor for a 256×256 image are listed below.

**Steps in finding the total HOG feature vector of the face image.**

1. The face image extracted from the top patch of the human detection region is resized to 32X32.

2. The 32X32 matrix is divided into 16 number of 8X8 cells.

3. Each 8X8 matrix is converted in to vector of length 9.

4. The vectors of first 4 number of 8X8 cells are concatenated to form a vector of length 36.

5. This 36 length vector is block normalized by using L2-Normalization.

### 4.3: SUPPORT VECTOR MACHINE

#### 4.3.1: Introduction

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate.

### 4.4: DATA SETS

Three data sets namely AITAM1, AITAM2 and AITAM3 are used for conducting experiments.

#### 4.4.1: Description of the Data Set AITAM1 (Simple)

In AITAM1 dataset, there are two persons. The frame rate is 25 frames per second. The camera used is of 8 mega pixel and the size of the frame is 232x454. The duration of the video is 20.24 seconds. The head movement is up to $5^{\circ}$. There are no occlusions among the persons. All are faced towards the camera. There are no shadows.


(a) (b)

Fig: 4.1 (a) and (b) are the sample frames from AITAM1 dataset


(a) (b)

Fig: 4.2. (a) and (b)   are the sample frames from AITAM2 dataset

### 4.4.2: Description of the Data Set AITAM2 (Moderate)

In AITAM2 dataset, there are three persons. The frame rate is 25 frames per second.

The camera used is of 16 mega pixel and the size of the frame is 768X1024. The duration of the video is 35 seconds. There are only partial occlusions among the persons. The persons are moving towards the camera and also away from the camera. There are no shadows.

### 4.4.3: Description of the Data Set AITAM3 (Complex)

In AITAM3 dataset, there are five persons. The frame rate is 25 frames per second.


(a)


(b)

Fig: 4.3. (a) and (b) are the sample frames from AITAM3 dataset

The camera used is 16 mega pixel and the size of the frame is 768X1024. The duration of the video is 68 seconds. There are partial and full occlusions among the persons. There are no shadows. The persons are moving towards, away and also across the camera.

## 4.5: HUMAN TRACKING USING GMM

The annotations of the human are taken for the video. These annotations are used for creating the regions. In this work, the template has been divided into four regions. The top region is for face and the remaining regions are for chest, waist and the legs. By using the GMM algorithm for individual regions and making connection among the regions assuming that all regions move together for human. Here, 10% of horizontal variation is given between the two regions.

## 4.6: FACE RECOGNITION USING HOG AND SVM

The topmost region from the template is taken for face recognition. Every $10^{th}$frame is considered for taking the face regions of all the faces in the frame. The first 10 face regions are taken and extracted the HOG features. These HOG features are given to the SVM classifier for training purpose. Ones the classifier is prepared, the remaining frames are used for testing purpose. The block diagram of SVM classifier trainer and tester are shown in the figure below. The first 10 face regions of all the three faces in the video are shown below. These face regions are used for training the classifier. The remaining frames are used for testing purpose. All three AITAM data sets have been taken for testing.

There are three people in the video of AITAM2. The sample face regions which are separated for the extraction of HOG features are shown in figure 4.6. These face regions are taken from every tenth frame of the dataset.

(a) (b) (c)

Fig: 4.4(a) Sample frame showing the detection and tracking of person 1 in the video; (b) Sample frame showing the detection and tracking of person 2 in the video; (c) Sample frame showing the detection and tracking of person 3 in the video

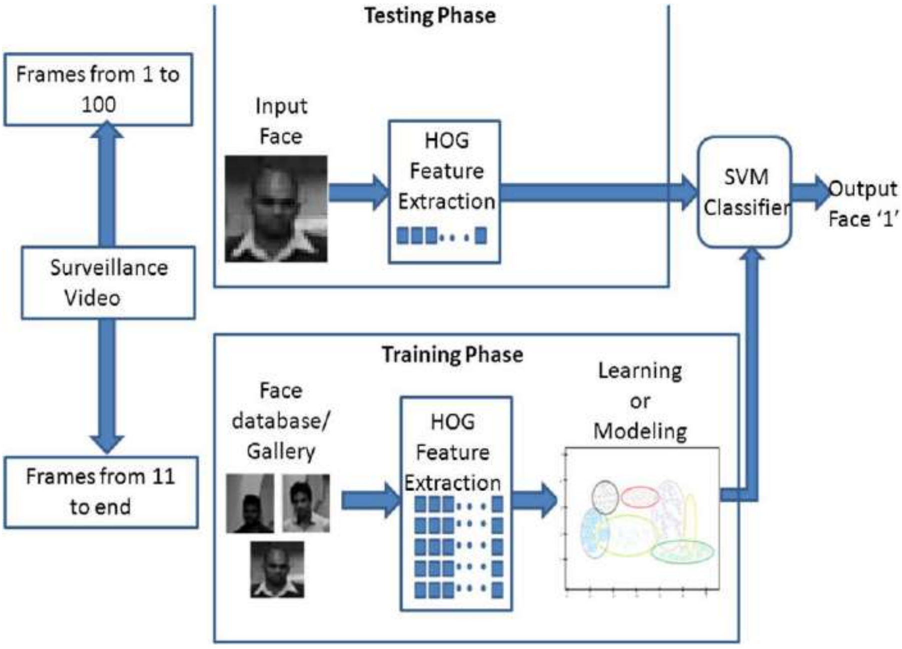



Fig: 4.5 Testing and training blocks of SVM classifier

(a)

(b)

(c)

Fig: 4.6 (a)Sample faces of person 1 from the AITAM2 database for testing; (b) Sample faces of person 2 from the AITAM2 database for testing; (c) Sample faces of person 3 from the AITAM2 database for testing and given to the SVM classifier.

**4.7: EXPERIMENTAL RESULTS**

Recognition and tracking of human in surveillance video are done. GMM algorithm performs well for tracking the human. Connection formed among the regions gives best results for human tracking. The HOG feature extraction of the faces is best for face recognition. SVM classifier used in this work performed well for human classification.

HOG features are extracted for the faces of every $10^{th}$ frame of first 100 frames. These HOG features are given to the SVM classifier for training. For recognition, depending on the requirement, the corresponding frame in the video is taken and the faces of that frame are extracted and given to the classifier for testing. In this way the face recognition is done. The same experiment has been done by taking every $5^{th}$ frame and every $3^{rd}$ frame from the first hundred frames for extracting HOG features for training. However the number of testing frames is same for all the three experiments for all the three datasets. The testing frame is taken from every $10^{th}$ frame of the video.

Table 4.1 shows the comparison of face recognition rate for different training sets of the same dataset AITAM1. The number of training faces considered for each person are 33 in case 1, 20 in case 2 and 10 in case 3. Table 4.2 shows the results of dataset AITAM2 and Table 4.3 for the dataset AITAM3.

| Dataset | Person | Case I 33 training faces for each face | Case II 20 training faces for each face | Case III 10 training faces for each face |
|---|---|---|---|---|
| | | Face Recognition Rate | | |
| AITAM1 (Simple) | ROBBER | 86.21% | 62.07% | 53.45% |
| | SUSPICIOUS PERSON | 79.31% | 68.97% | 60.34% |

TABLE 4.1: Face Recognition Rate for AITAM1 Dataset for different cases

| Dataset | Person | Case I 33 training faces for each face | Case II 20 training faces for each face | Case III 10 training faces for each face |
|---|---|---|---|---|
| | | Face Recognition Rate | | |
| AITAM2 (Moderate) | ROBBER | 60.00% | 50.67% | 46.67% |
| | SUSPICIOUS PERSON | 54.67% | 48.00% | 41.33% |
| | LOST PERSON | 52.00% | 44.33% | 38.67% |

TABLE 4.2: Face Recognition Rate for AITAM2 Dataset for different cases

| Dataset | Person | Case I 33 training faces for each face | Case II 20 training faces for each face | Case III 10 training faces for each face |
|---|---|---|---|---|
| | | Face Recognition Rate | | |
| AITAM3 (Complex) | ROBBER | 54.97% | 50.94% | 47.80% |
| | SUSPICIOUS PERSON | 39.62% | 38.36% | 37.11% |
| | LOST PERSON | 44.65% | 37.11% | 32.08% |
| | TERRORIST | 34.59% | 32.70% | 29.56% |
| | SUSPICIOUS PERSON | 57.86% | 48.43% | 44.65% |

TABLE 4.3: Face Recognition Rate for AITAM3 Dataset for different cases

## 4.8: CONCLUSIONS

Face recognition in video surveillance system is introduced in this work. Video surveillance system for human tracking is complete when it has these two components namely recognition of human and tracking. The first hundred frames are used for gathering data for person identification. To track the human, face features are extracted using HOG. These features are given to the SVM classifier. The results show that HOG and SVM combination works better for face recognition. Three different datasets are taken with different degrees of complexity. The results show that the algorithm works well for all three datasets. Finally, the results in all these datasets shows that the more the number of training faces taken for classification, the more the face recognition rate. The same experiment can also be extended to multiple human tracking.

# CHAPTER – 5

# HUMAN TRACKING USING GMM

### 5.1.1: Number of Pixels

The number of past pixels considered for modeling the current pixel are limited in this algorithm. Rather than considering the entire history of pixels in that position, here only a set of past pixels are considered for background modeling.

Following are the list of reasons for considering the limited number of pixels for background modeling:

1. GMM is known to diverge and find solution with infinite likelihood if the entire history of pixels are considered.

2. The object crosses a pixel completely within 25 to 50 frames in pedestrian tracking. (Considering the speed of movement of the pedestrian to be average and the resolution of the camera to be average).

3. Limited number of pixels means limited number of data points for each distribution. The number of iterations taken by the EM algorithm also reduces therefore finds quick solution.

### 5.1.2: Tracking Human Using GMM

The algorithm developed in this work is used for human tracking. The head and foot points of the human are first detected and given as input for this algorithm. Color and shape based automatic human detection algorithm given in chapter two is used for extracting the annotations. It is assumed here that the pedestrian is walking. Pixel based point tracking is used here. Figure 6.1 shows how the object is tracked using GMM algorithm.

Fig: 5.1 Human tracking using GMM algorithm

#### 5.1.2.1: Number of Regions

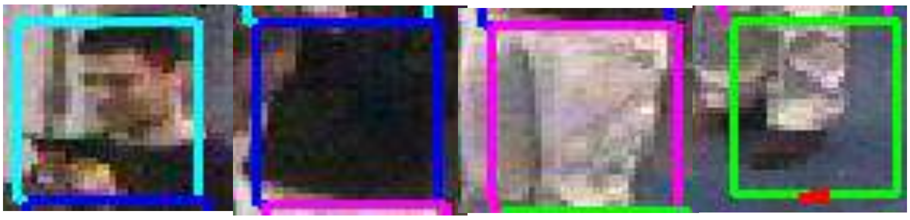
Based on the aspect ratio of pedestrian, the box has been divided into four regions. The first three regions from the bottom are of rectangle shape. The top region could be either rectangle shape or the circle shape. For the face, circle is the better shape to fit. But here only the rectangle shape is used because the features of the face are also to be extracted for recognition of the face. Therefore only rectangle shape for all the regions are taken. Figure 5.2 (a), (b) and (c) are the examples of pedestrians body parts assignment in the algorithm.

(a) (b) (c)

Fig: 5.2 four region assignment for humans

### 5.1.2.2: Region Assignment

These four rectangular shaped regions are assigned to four parts of the human body. The first rectangle from the top is assigned to the face of the human. The second rectangle is assigned to the chest, the third region is assigned to the top part of the legs and the fourth region is assigned to the legs from the knees. The rectangular box has been divided into four regions in order to have maximum likelihood. Figure 5.3 shows the rectangles formed on the pedestrian.

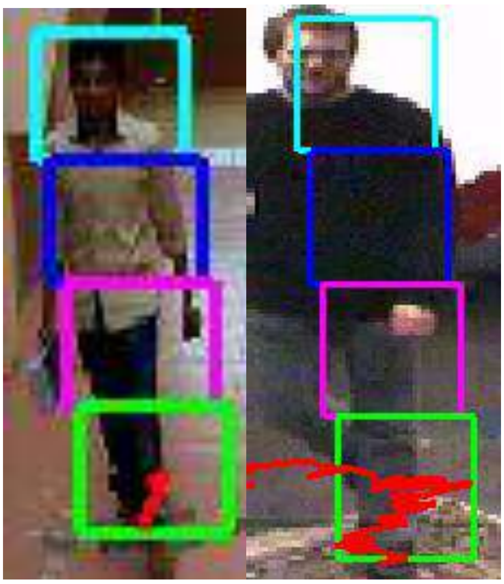


(a) (b) (c) d)

Fig: 5.3 (a) First rectangle assigned to the face of the region; (b) Second rectangle assigned to the chest and stomach region; (c) third rectangle is assigned to the upper legs region and; (d) fourth rectangle is assigned to the legs region.

### 5.1.2.3: Region Connection

The vertical connection has been formed among the regions. All four regions are one above the other. While selecting the rectangle in the present frame, three considerations are to taken. First thing is selecting box with maximum number of foreground pixels. Second thing is that the position of the rectangle box is such that there should be maximum of 10% horizontal deviation. The last consideration is that the vertical connection among these regions is always intact. Figure 5.4 shows the horizontal variation between the two adjacent rectangles.



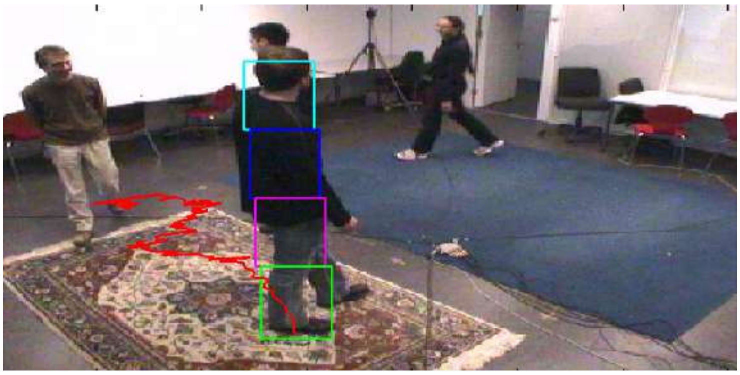
(a) (b)

Fig: 5.4 Maximum of 10% horizontal variation allowed between the two successive rectangles

**5.1.2.4: Region Based Human Tracking**

The human is tracked by dividing the head to the foot rectangle in to four regions. The middle point of the bottom horizontal line of the last rectangle is chosen for drawing the tracking line. The line shows the path of the human. The size of the four rectangles also changes according to the position of the human from the camera. If the human is far from the camera, the size of the rectangles decreases and when the human is coming closure to the camera, the size increases. By vertical binding all the rectangles, the human tracking is preserved the most. Human is tracked as long as the human is in the vicinity of the camera. If the pedestrian goes and comes back in to the vicinity, he will not be recognized as the same person. The pedestrian is only tracked but not remembered in this algorithm. Figure 5.5 shows the change of size of the rectangles according to the size of the human.

(a)



(b)

Fig: 5.5: (a) Small size rectangles for human being distant from the camera; (b) bigger size rectangle for human being far away from camera.

#### 5.1.2.5: Tracking Path

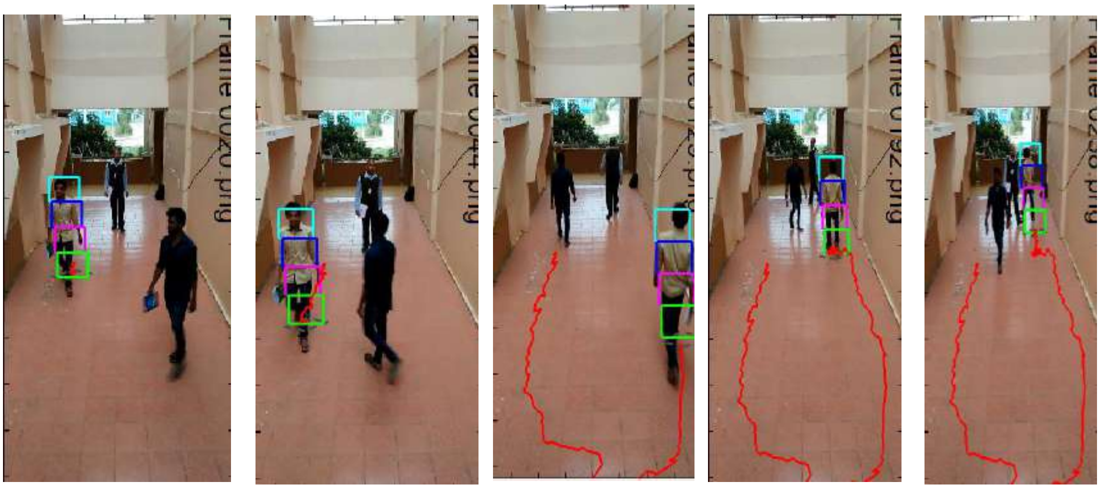
Tracking algorithm is complete only when the tracking path is shown. The middle point of the bottom line of the lower rectangle is remembered for every frame. The continuous line between these points of two successive frames is drawn by using plot function in MATLAB. The line shows the tracking path of the human. The same is developed both for single human tracking and also multi human tracking algorithms. Figure 5.6 shows how the human tracking path is formed both single human as well as multi human tracking.

(a) (b)

Fig: 5.6 (a) Single tracking path shown in the PETS 1 2009 dataset; (b) Multi tracking path shown in PETS 1 2009 dataset.

### 5.2: Experimental Results

GMM algorithm is tested by taking standard pedestrian datasets. PETS (Performance Evaluation of Tracking Systems) is a standard dataset for pedestrian tracking. Every year they release a dataset for evaluating the pedestrian tracking algorithm. PETS 2009 datasets are taken for experimentation. The other editions of the PETS are for criminal detection and tracking, crowd tracking, quarrel tracking datasets. The other datasets taken for this experimentation are LAB dataset, SOCCER datasets, CAVIAR datasets, Town Centre dataset and AITAM dataset. The annotations of pedestrians (the head and the foot points of the pedestrians) are taken from the algorithm developed for human detection in this work. The color and shape based human detection algorithm in pedestrian tracking is given in chapter 3.



(a) (b) (c) (d) (e)

Fig: 5.7 (a), (b), (c), (d) and (e) are the sequence of frames with numbers 20, 44, 129, 192, 238 respectively from AITAM database showing the tracking of pedestrian 1.

Figure 5.7 shows sequence of frames from AITAM database. AITAM database is a simple pedestrian database with three persons moving in the corridor. The dataset is of 688 frames and frame resolution of 231X452. The bit depth is 24 bits. The frame rate is 24 frames per second. The proposed human tracking algorithm is applied on all the three pedestrians.

## 5.8: CONCLUSIONS

Gaussian Mixture Model algorithm is best suited for tracking the humans. By taking the aspect ratio of the pedestrian into consideration, the region of interest is divided into four regions. The regions are assigned to different parts of the human body. The GMM algorithm is applied on each and every pixel in the region. By considering only the finite number of pixels from the past frames in deciding whether the present pixel belongs to foreground or background, the algorithm takes less number of iterations to converge. The proposed algorithm is compared with the existing Gaussian Mixture Model algorithm. The performance metrics shows that the proposed algorithm performing well for human tracking in terms of time taken for tracking.

# CHAPTER – 6

# PERFORMANCE METRICS AND EXPERIMENTAL RESULTS OF HUMAN TRACKING USING GMM ALGORITHM

## 6.1: QUANTITATIVE METRICS

There are various quantitative metrics available for measuring the performance of tracking algorithm. All the metrics can be found by first forming the confusion matrix.

### 6.1.1: Confusion Matrix

Confusion matrix is a table which contains two rows and two columns that reports the number of true positives, true negatives, false negatives and false positives.

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">True Condition</td></tr>
<tr><td>Condition Positive</td><td>Condition Negative</td></tr>
<tr><td rowspan="2">Predicted Condition</td><td>Predicted Condition Positive</td><td>TRUE POSITIVE</td><td>FALSE POSITIVE</td></tr>
<tr><td>Predicted Condition Negative</td><td>FALSE NEGATIVE</td><td>TRUE NEGATIVE</td></tr>
</table>

#### 6.1.1.1: True Positives

True positive means the overlap between the ground truth and the algorithm output is greater than the threshold. If the threshold value is less, it means the confidence is high. If the thresholding value is more, it means that the confidence is less. A specific threshold value has to be fixed for comparison of the tracking algorithms. In this experiment, the value chosen for threshold is 0.25.

#### 6.1.1.2: False Positives

False positive means that there is no entry in the ground truth database, but still there is a result showing by the algorithm.

#### 6.1.1.3: False Negatives

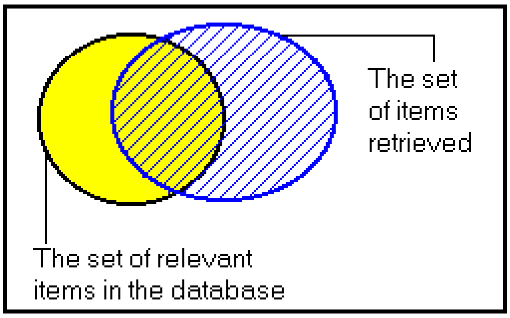
False negative means even though there is ground truth entry, still there is no result by the algorithm.

#### 6.1.1.4: True Negatives

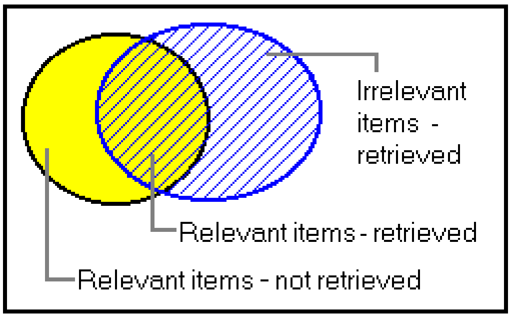
If there is no entry in the ground truth and also there is no entry by the algorithm, it is counted as true negative.

When the overlap is less than the fixed threshold value, it is counted as both false negative and false positive.

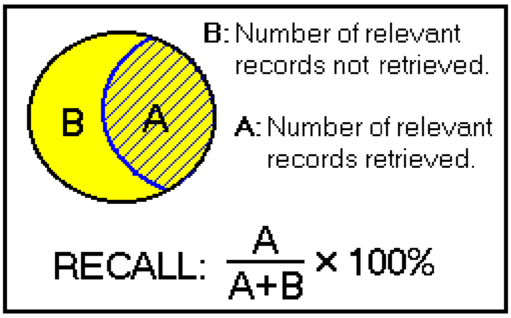
### 6.1.2: Recall




(a)




(b)




(c)

Fig. 6.1 (a), (b) and (c) Venn diagrams of Recall

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

### 6.1.3: Precision

Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.

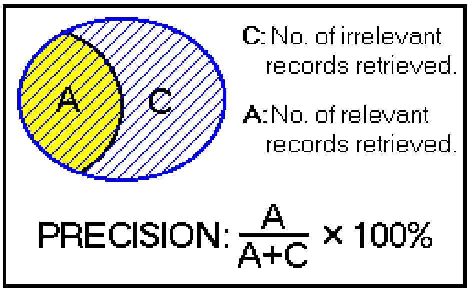



Fig: 6.2 Venn diagram for Precision

### 6.1.4: False Alarm per Frame

The average number of false alarms per frame.

### 6.1.5: Mostly Lost

The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.

| Metrics / Algorithm | Data Set | No. of Persons | No. of frames | Frame rate | Duration | Recall | Precision |
|---|---|---|---|---|---|---|---|
| GMM with 4 Regions | AITAM1 | 2 | 688 | 25 | 28sec | 48.68% | 27.27% |
| | AITAM2 | 3 | 857 | 25 | 35sec | 31.55% | 21.79% |
| | AITAM3 | 5 | 1691 | 25 | 68sec | 21.10% | 20.37% |

TABLE 6.1: Performance results for tracking the human in the AITAM Datasets.

| Metrics / Algorithm | Data Set | False Alarm per Frame (Fa/F) | Ground Truth (GT) | False Positives (FP) | False Negatives (FN) | Mostly Lost (ML) |
|---|---|---|---|---|---|---|
| GMM with 4 Regions | AITAM1 | 0.59 | 1365 | 344 | 136 | 21.10% |
| | AITAM2 | 1.55 | 2502 | 402 | 243 | 47.64% |
| | AITAM3 | 3.19 | 8291 | 692 | 662 | 71.10% |

TABLE 6.2: Performance results for tracking the human in the AITAM Datasets (Contd.)

## 6.2: CONCLUSIONS

Qualitative and quantitative performance metrics for human tracking algorithms are introduced in this work. For the human tracking algorithms namely GMM, quantitative metrics like precision, recall, false alarm per frame, mostly lost, false positives and false negatives are calculated. AITAM1, 2 and 3 datasets are taken for finding the performance of the algorithms. Qualitative metrics like occlusion complexity, clutter complexity, illumination complexity and contrast complexity are calculated and found that GMM algorithm is performing well for tracking humans.

# CHAPTER - 7

# CONCLUSIONS

The proposed color and shape based automated human detection algorithm is producing good human detection metrics for standard datasets namely PETS 2009 and also for the developed datasets in this work. The detected human's face features are extracted and given to the SVM classifier for training. After the human is detected and recognized either as the robber, suspicious person, terrorists or lost person, the human is tracked and the path is formed. Three datasets namely AITAM1, AITAM2 and AITAM3 with different levels of complexity are developed. In each dataset, combination of all four category people are considered. In AITAM1 dataset has a robber and a suspicious person. AITAM2 dataset has a robber, a suspicious person and a lost person and AITAM3 dataset has a robber, lost person, terrorist and two suspicious persons. The proposed algorithm is producing better face recognition rate. The proposed algorithm reduced the processing time, without compromising the other performance metrics. Quantitative metrics are calculated for GMM algorithm. Performance results show that the algorithm is working well, if the dataset environment is either occlusive or cluttered or contrast or illuminative. The combination of recognizing and tracking the person makes the human tracking systems intelligent and more authenticated.

## FUTURE SCOPE

The proposed algorithm can be extended further for multiple human tracking. Handling the occlusions in multiple tracking without missing the tracking path and without identity switches are some of the challenges in this area of work.

# REFERENCES


1. Gavrila D.M. “Pedestrian detection form a moving vehicle”, in: VernonD(ed) ECCV 2000, LNCS, vol 1843. Springer, Heidelberg, pp 37–49
2. Papageorgiou C, Oren M, Poggio T (1998) A general framework for object detection. In: Proceedings of international conference on computer vision.
3. Conde C, Moctezuma D, De Diego IM, Cabello E (2013) HoGG: Gabor and HoG-based human detection for surveillance in non-controlled environments. Neurocomputing 100(16):19–30.
4. Csaba Beleznai, Bernhard Fr¨uhst¨uck, “Multiple Object Tracking Using Local PCA”, Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, 20-24 Aug. 2006.
5. Gongyi Xia, Simone A. Ludwig, “Object-tracking based on particle filter using particle swarm optimization with density estimation”, Evolutionary Computation (CEC), 2016 IEEE Congress on, 24-29 July 2016.
6. Dalal N, Triggs B (2005) Histogram of oriented gradients for human detection. In: IEEE international conference on computer vision and pattern recognition.
7. Wu Gang, Zhang Haofeng, “Appearance-based subspace learning model using incremental PCA in object tracking”, Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference on, 12-13 Oct. 2013.
8. P. Dollár, R. Appel, S. Belongie, P. Perona, "Fast feature pyramids for object detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532-1545, Aug. 2014.
9. Amith R, V.N. ManjunathAradhya, S.K. Niranjan, “The application of cosine transform and principal components for foreground detection in video”, Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, 8-10 Oct. 2015.

10. Shreenandan Kumar, Suman Kumari, SuchetaPatro, Tushar Shandilya, Anuja Kumar Acharya, “Adaptive visual tracking on Euclidean space using PCA”, Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, 10-13 Aug. 2015.
11. HadiFirouzi, HomayounNajjaran, “Robust PCA-Based Visual Tracking by Adaptively Maximizing the Matching Residual Likelihood”, Computer and Robot Vision (CRV), 2013 International Conference on, 28-31 May 2013.
12. Liao S, Zhao G, Kellokumpu V, Pietikäinen M & Li SZ. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. Proceedings CVPR2010: 1301–1306.
13. Kai Zhou, Rui-xia Fan, Wei-xing Li, A SIFT-Kalman information fusion tracking algorithm based on PCA”, Intelligent Control and Automation (WCICA), 2010 8th World Congress on, 7-9 July 2010.
14. Bohyung Han, L. Davis, “Object tracking by adaptive feature extraction”, Image Processing, 2004. ICIP '04. 2004 International Conference on, 24-27 Oct. 2004.
15. Kalpana Goyal, Jyoti Singhai, “Review of background subtraction methods using Gaussian mixture model for video surveillance systems”, Artificial Intelligence Review, Springer online on January 2017
16. Komagal E, Seenivasan V, Anand K, Anand raj CP (2014) Human detection in hours of darkness using Gaussian mixture model algorithm. Int J InfSci Tech: IJIST 4(3):84–89.
17. Fei Feng, Xiao-Jun Wu, Tianyang Xu, “Object tracking with kernel correlation filters based on mean shift”, Smart Cities Conference (ISC2), 2017 International, 14-17 Sept. 2017.

18. Kun Chen, ChunLei Liu, Yongjin Xu, "Face Detection and Tracking Based on AdaboostCamShift and Kalman Filter Algorithm", International Conference on Life System Modeling and Simulation and International Conference on Intelligent Computing for Sustainable Energy and EnvironmentLSMS/ICSEE 2014: Computational Intelligence, Networked Systems and Their Applications pp 149-158.

19. JaehyunIm, Jaehoon Jung, Joonki Paik, "Single Camera-Based Depth Estimation and Improved Continuously Adaptive Mean Shift Algorithm for Tracking Occluded Objects", Pacific Rim Conference on MultimediaPCM 2015: Advances in Multimedia Information Processing -- PCM 2015 pp 246-252.

20. Dawei Li, Lihong Xu, Yang Wu, "Improved CAMShift object tracking based on Epanechnikov Kernel Density Estimation and Kalman filter, Control And Decision Conference (CCDC), 2017 29th Chinese, 28-30 May 2017.

21. Zheng Han, Rui Zhang, Linru Wen, XiaoyiXie¸Zhijun Li, "Moving Object Tracking Method Based on Improved Camshift Algorithm", Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016 International Conference on, 3-4 Dec. 2016.

22. Yumei Zhu, Bo Xu, "A robust object tracking method combining shape descriptor and adaptive background camshift", Natural Computation (ICNC), 2015 11th International Conference on, 15-17 Aug. 2015.

23. Manjunath Kulkarni, Paras Wadekar, HareshDagale, "Block Division Based CAMShift Algorithm for Real-Time Object Tracking Using Distributed Smart Cameras", Multimedia (ISM), 2013 IEEE International Symposium on, 9-11 Dec. 2013

ANNEXURE – 1

## MATLAB CODE USED FOR GENERATING THE FINAL RESULTS

**main_pets_single()**

```
clc; close all; clear;
load anno_pets;
format long;
i=1;
patch=4;
options.feature_type   = 'color_hist';
options.n_bins = 64;
options.walking_speed = 1.5;
options.relaxation_parameter = 0.5;
options.world_search_sample_step = 2;
options.show_frames = true;
options.save_frames = false;
options.image_pref      =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src\datasets\pets1\Crowd_PETS09\S2\L1\Time_12-
34\View_001\';
options.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options.d_mask          = 6;
options.file_ext        = 'jpg';
options.world_unit      = 'mm';
options.min_patch_height = 20;
options.video_fps       = 7;
options.begin_frame    = anno(i);
options.end_frame      = 794;
options.head_point     = [anno(i,2); anno(i,3)];
options.foot_point     = [anno(i,4); anno(i,5)];
options.patch_width    = 28;
options = validate_options(options);
disp(options);
run_tracker1(options,patch);
```

**main_pets_multi()**

```
load anno_pets;
format long;
i1=4; i2=5dataset = 'pets_s01';
patch=4;
options1.feature_type   = 'color_hist';
options1.n_bins = 64;
options1.walking_speed = 1.5;
options1.relaxation_parameter = 0.5;
options1.world_search_sample_step = 2;
options1.show_frames = true;
options1.save_frames = false;
options1.out_path = 'out/';
options1.out_filename = ['results_', dataset, '.mat'];
options1.image_pref      =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src/datasets/pets1/Crowd_PETS09/S2/L1/Time_12-
34/View_001/';
options1.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options1.d_mask          = 6;
options1.file_ext        = 'jpg';
options1.world_unit      = 'mm';
options1.min_patch_height = 20;
options1.video_fps       = 7;
options1.begin_frame    = anno(i1);
options1.end_frame      = 794;
options1.head_point     = [anno(i1,2); anno(i1,3)];
```

```
options1.foot_point     = [anno(i1,4); anno(i1,5)];
options1.patch_width    = 28;
options1 = validate_options(options1);
options2.feature_type   = 'color_hist';
options2.n_bins = 64;
options2.walking_speed = 1.5;
options2.relaxation_parameter = 0.5;
options2.world_search_sample_step = 2;
options2.show_frames = true;
options2.save_frames = false;
options2.out_path = 'out/';
options2.out_filename = ['results_', dataset, '.mat'];
options2.image_pref       =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src/datasets/pets1/Crowd_PETS09/S2/L1/Time_12-
34/View_001/';
options2.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options2.d_mask           = 6;
options2.file_ext         = 'jpg';
options2.world_unit       = 'mm';
options2.min_patch_height = 20;
options2.video_fps        = 7;
options2.begin_frame    = anno(i2);
options2.end_frame      = 794;
options2.head_point     = [anno(i2,2); anno(i2,3)];
options2.foot_point     = [anno(i2,4); anno(i2,5)];
options2.patch_width    = 28;
options2 = validate_options(options2);
run_tracker2(options1,options2,patch);
```

**main_aitam3_single()**

```
clc; close all; clear;
load AITAM_2_36;
format long;
 i=2;
patch=4;
options.feature_type   = 'color_hist';
options.n_bins = 64;
options.walking_speed = 1.5;
options.relaxation_parameter = 0.5;
options.show_frames = true;
options.save_frames = false;
options.image_pref       =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src\datasets\AITAM_2_PERSONS_1_117-235\Frame ';
options.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options.d_mask           = 4;
options.file_ext         = 'png';
options.world_unit       = 'mm';
options.min_patch_height = 20;
options.video_fps        = 7;
options.begin_frame    = AITAM_2_36(i,1);
options.end_frame      =235;
options.head_point     = [AITAM_2_36(i,2); AITAM_2_36(i,3)];
options.foot_point     = [AITAM_2_36(i,4); AITAM_2_36(i,5)];
options.patch_width    = 28;
options = validate_options(options);
tStart = tic;
run_tracker1_face(options,patch);
tElapsed = toc(tStart);
```

**MAIN_AITAM3_MULTI**

```
load anno_AITAM3_1356FRAME;
format long;
i1=1; i2=2;
dataset = 'pets_s01';
 patch=4;
options1.feature_type   = 'color_hist';
options1.n_bins = 64;
options1.walking_speed = 1.5;
options1.relaxation_parameter = 0.5;
options1.world_search_sample_step = 2;
options1.show_frames = true;
options1.save_frames = false;
options1.out_path = 'out/';
options1.out_filename = ['results_', dataset, '.mat'];
options1.image_pref      =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src\datasets\AITAM3_875outof1811\Frame ';
options1.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options1.d_mask          = 4;
options1.file_ext        = 'png';
options1.world_unit      = 'mm';
options1.min_patch_height = 20;
options1.video_fps       = 7;
options1.begin_frame     = a1(4,1);
options1.end_frame       = a1(4,2);
options1.head_point      = [a1(i1,1); a1(i1,2)];
options1.foot_point      = [a1(i1,3); a1(i1,4)];
options1.patch_width     = 28;
options1 = validate_options(options1);
options2.feature_type   = 'color_hist';
options2.n_bins = 64;
options2.walking_speed = 1.5;
options2.relaxation_parameter = 0.5;
options2.world_search_sample_step = 2;
options2.show_frames = true;
options2.save_frames = false;
options2.out_path = 'out/';
options2.out_filename = ['results_', dataset, '.mat'];
options2.image_pref      =
'F:\work\RRM3\GMM__HUMAN_TRACKING\src\datasets\AITAM3_875outof1811\Frame ';
options2.calib_filename   = '~/datasets/pets/cparsxml/View_001.xml';
options2.d_mask          = 4;
options2.file_ext        = 'png';
options2.world_unit      = 'mm';
options2.min_patch_height = 20;
options2.video_fps       = 7;
options2.begin_frame     = 175;
options2.end_frame       = 794;
options2.head_point      = [a1(i2,1); a1(i2,2)];
options2.foot_point      = [a1(i2,3); a1(i2,4)];
options2.patch_width     = 28;
options2 = validate_options(options2);
run_tracker2(options1,options2,patch);
```

**run_tracker_face1()**

```
function run_tracker1_face(options,patch)
```

```
im_first_frame = get_frame(options, options.begin_frame);
tracker = init_tracker(options, im_first_frame,patch);
if options.show_frames
    figure;
    imshow(im_first_frame); hold on;
   face_dimen= plot_patches_face(tracker.patches); hold off;
    pause(0.1);
    if options.save_frames
        out_frame = sprintf('%strack%s.png', options.out_path, format_int(options.d_mask, options.begin_frame));
        export_fig(out_frame, '-a2');
    end

end
kk=0;
tracer_row=[];
tracer_col=[];
i_result = 1;
k=1;
predicted = false;
for f = options.begin_frame:1:options.end_frame
    fprintf('Frame %d of %d ', f, options.end_frame); tic;
    im_frame = get_frame(options, f);
    if options.show_frames
        imshow(im_frame); hold on;
    end

    if ~isempty(tracker.last_displacement)
        tracker = motion_prediction(tracker);
        predicted = true;
    end
    if predicted
        tracker = track_patches(tracker, im_frame, options, tracker.predicted_vector);
    else
        tracker = track_patches(tracker, im_frame, options);
    end
    r_patches{i_result}.n_candidates = tracker.n_candidates;
    tracker.patches = compute_world_d_vectors(tracker.patches, options);
    patches1=tracker.patches;
            roi1 = patches1(patch).roi;
    tracer_row=[tracer_row, (roi1(1,2)+ roi1(1,1))/2];
    tracer_col=[tracer_col (roi1(2,2))];
    if predicted
        tracker.filtered_vector = wvmf(tracker.patches, options, tracker.predicted_vector, tracker.best_matchings);
    else
        tracker.filtered_vector = wvmf(tracker.patches, options);
    end
    tracker = move_w_vector(tracker, tracker.filtered_vector, options);
    tracker.wd_vector = tracker.filtered_vector;
    tracker.gpoint    = tracker.gpoint + tracker.wd_vector;

    tracker = scale_w_homography(tracker, options.H);
    tracker = compute_motion_information(tracker, options);
    if options.show_frames
      face_dimen=  plot_patches_face(tracker.patches); hold on;
plot(tracer_row, tracer_col,'Color','r','LineWidth',2);
        pause(0.1); hold off;
       facepart= imcrop(im_frame,face_dimen);
       kk1='F:\work\RRM3\GMM__HUMAN_TRACKING\src\FaceParts data set\AITAM5_FACE1\';
       kk=sprintf('%4.4d.png', k);
       kk2=strcat(kk1,kk);
```

```
            imwrite(facepart, kk2);
            kk3= imread(kk2);
            kk4= imresize(kk3,[32 32]);
            imwrite(kk4,kk2);
        if options.save_frames
            out_frame = sprintf('%strack%s.png', options.out_path, format_int(options.d_mask, f));
            export_fig(out_frame, '-a2');
        end
    end
    k=k+1;
    r_patches{i_result}.patches = tracker.patches;
    r_patches{i_result}.c_point = extract_central_point(tracker.patches);
    i_result = i_result + 1;
    toc;
end
disp(tracer_row);
disp(tracer_col);
frames = [options.begin_frame:options.end_frame];
```

**run_tracker2()**

```
function run_tracker2(options1, options2,patch)
tracer_row1=[];
tracer_col1=[];
tracer_row2=[];
tracer_col2=[];
im_first_frame1 = get_frame(options1, options1.begin_frame);
im_first_frame2 = get_frame(options2, options2.begin_frame);
 tracker1 = init_tracker(options1, im_first_frame1,patch);
tracker2 = init_tracker(options2, im_first_frame2,patch);
 if options1.show_frames
    figure;
     imshow(im_first_frame1);
hold on;
    plot_patches(tracker1.patches);
      plot_patches(tracker2.patches);
 end
 predicted1 = false;
predicted2 = false;
 for f = options1.begin_frame+1:options1.end_frame
     fprintf('Frame %d of %d \n', f, options1.end_frame);
    im_frame1 = get_frame(options1, f);
    if options1.show_frames
        imshow(im_frame1); hold on;
    end
    if ~isempty(tracker1.last_displacement)
        tracker1 = motion_prediction(tracker1);
        predicted1 = true;
    end
    if predicted1
        tracker1 = track_patches(tracker1, im_frame1, options1, tracker1.predicted_vector);
    else
        tracker1 = track_patches(tracker1, im_frame1, options1);
    end
    tracker1.patches = compute_world_d_vectors(tracker1.patches, options1);
    patches1=tracker1.patches;
    roi1 = patches1(patch).roi;
    patches2=tracker2.patches;
    roi2 = patches2(patch).roi;
```

```
    if predicted1
        tracker1.filtered_vector = wvmf(tracker1.patches, options1, tracker1.predicted_vector,
tracker1.best_matchings);
    else
        tracker1.filtered_vector = wvmf(tracker1.patches, options1);
    end
    tracker1 = move_w_vector(tracker1, tracker1.filtered_vector, options1);
    tracker1.wd_vector = tracker1.filtered_vector;
    tracker1.gpoint    = tracker1.gpoint + tracker1.wd_vector;
    tracker1 = scale_w_homography(tracker1, options1.H);
    tracker1 = compute_motion_information(tracker1, options1);
    tracer_row1=[tracer_row1, (roi1(1,2)+ roi1(1,1))/2];
    tracer_col1=[tracer_col1 (roi1(2,2))];
    tracer_row2=[tracer_row2, (roi2(1,2)+ roi2(1,1))/2];
    tracer_col2=[tracer_col2 (roi2(2,2))];
    if f>=options2.begin_frame
        im_frame2 = get_frame(options2, f);
    if ~isempty(tracker2.last_displacement)
        tracker2 = motion_prediction(tracker2);
        predicted2 = true;
    end
    if predicted2
        tracker2 = track_patches(tracker2, im_frame2, options2, tracker2.predicted_vector);
    else
        tracker2 = track_patches(tracker2, im_frame2, options2);
    end

    tracker2.patches = compute_world_d_vectors(tracker2.patches, options2);
    if predicted2
        tracker2.filtered_vector = wvmf(tracker2.patches, options2, tracker2.predicted_vector,
tracker2.best_matchings);
    else
        tracker2.filtered_vector = wvmf(tracker2.patches, options2);
    end
    tracker2 = move_w_vector(tracker2, tracker2.filtered_vector, options2);
    tracker2.wd_vector = tracker2.filtered_vector;
    tracker2.gpoint    = tracker2.gpoint + tracker2.wd_vector;
    tracker2 = scale_w_homography(tracker2, options2.H);
    tracker2 = compute_motion_information(tracker2, options2);
   plot_patches(tracker2.patches);
   plot(tracer_row1, tracer_col1,'Color','r','LineWidth',2);
   plot(tracer_row2, tracer_col2,'Color','g','LineWidth',2);
    end
        plot_patches(tracker1.patches);
        pause(0.1);
end
save([options1.out_path, options1.out_filename], 'r_patches', 'frames', 'options');
disp('Done!');
end
```

**validate_options()**

```
function options = validate_options(options)
if ~isfield(options, 'min_patch_height'), options.min_patch_height = 20; end;
if ~isfield(options, 'search_window_size'), options.search_window_size = 30; end;
if ~isfield(options, 's_temporal_window'), options.s_temporal_window = 30; end;


if ~isfield(options, 'head_point') || ~isfield(options, 'foot_point')
```

```
    error('You need to specify points for the head and foot!');
end

if ~isfield(options, 'feature_type')
    warning('No feature type was informed! Using covariance...');
    options.feature_type = 'covariance';
end

if ~ismember(options.feature_type, {'color_hist', 'covariance'})
    error('Invalid feature type!');
else
    if strcmp(options.feature_type, 'color_hist')
        options.wvmf_gamma = 0.25;
        if ~isfield(options, 'n_bins'), options.n_bins = 64; end
    elseif strcmp(options.feature_type, 'covariance')
        options.wvmf_gamma = 0.15;
    end
end

if ~isfield(options, 'calib_filename')
    error('You need to inform the calibration xml file');
end
options.world_search_radius = compute_wsearch_radius(options);
[options.K, options.Rt] = parse_xml_calibration_file(options.calib_filename);
options.H = Rt2homog(options.Rt, options.K);
```

**plot_patches()**

```
function plot_patches(patches, varargin)

if nargin == 2
    color = varargin{1};
else
    color = 'g';
end

i=length(patches);
if i>=1
    roi = patches(1).roi;
plot([roi(1,1), roi(1,2), roi(1,2), roi(1,1), roi(1,1)], ...
        [roi(2,1), roi(2,1), roi(2,2), roi(2,2), roi(2,1)], '-', 'Color', 'c', 'LineWidth', 2); hold on;
end


if i>=2
    roi = patches(2).roi;
    plot([roi(1,1), roi(1,2), roi(1,2), roi(1,1), roi(1,1)], ...
        [roi(2,1), roi(2,1), roi(2,2), roi(2,2), roi(2,1)], '-', 'Color', 'b', 'LineWidth', 2); hold on;
end

if i>=3
    roi = patches(3).roi;
    plot([roi(1,1), roi(1,2), roi(1,2), roi(1,1), roi(1,1)], ...
        [roi(2,1), roi(2,1), roi(2,2), roi(2,2), roi(2,1)], '-', 'Color', 'm', 'LineWidth', 2); hold on;
end
if i>=4
    roi = patches(4).roi;
    plot([roi(1,1), roi(1,2), roi(1,2), roi(1,1), roi(1,1)], ...
```

```
    [roi(2,1), roi(2,1), roi(2,2), roi(2,2), roi(2,1)], '-', 'Color', color, 'LineWidth', 2); hold on;
end
```

**compute_motion_information()**

```
function tracker = compute_motion_information(tracker, options)

best_d = get_best_matching_distance(tracker.patches, tracker.filtered_vector);
tracker.best_matchings = [tracker.best_matchings best_d];
if size(tracker.best_matchings, 2) > options.s_temporal_window
   tracker.best_matchings = tracker.best_matchings(end-options.s_temporal_window+1:end);
end

tracker.last_displacement = tracker.wd_vector;
```

**compute_patches_height()**

```
function patches = compute_patches_height(patches, options)

original_roi  = patches(end).roi;
fpoint = [(original_roi(1,1)+original_roi(1,2))/2; original_roi(2,2)];
wf_point = ics2wcs([fpoint; 1], options.H);

P = options.K*options.Rt;

X = wf_point(1);
Y = wf_point(2);

for i=1:size(patches,2)
   cp = extract_central_point(patches(i));
   u = cp(1);
   v = cp(2);
   H = (P(2,1)*X + P(2,2)*Y + P(2,4) - P(3,1)*X*v - P(3,2)*Y*v - P(3,4)*v) / (P(3,3)*v - P(2,3));
   patches(i).height = H;
end
```

**create_patches()**

```
function patches = create_patches(hpoint, fpoint, np, patch_w,patch)

min_y = min(hpoint(2), fpoint(2));
max_y = max(hpoint(2), fpoint(2));

size_y = max_y - min_y;
n_y_patches=patch;
if n_y_patches == 0
   n_y_patches = 1;
   new_np_y = size_y;
else
  new_np_y = floor(size_y/n_y_patches);
end
rmd_y = mod(size_y, new_np_y);


l = cross([hpoint; 1], [fpoint; 1]); i_patch = 1;
for i=1:n_y_patches
   cyi = min_y + ((i-1)*new_np_y) + new_np_y/2;
   cxi =  (-l(2)*cyi - l(3))/l(1);
   init_y = min_y + ((i-1)*new_np_y);
   end_y  = min_y + (i*new_np_y);
```

```
    if i == n_y_patches
        end_y = end_y + rmd_y;
    end
    init_x = cxi - (patch_w/2);
    end_x  = cxi + (patch_w/2);
    patches(i_patch).roi = [init_x end_x; init_y end_y];
    i_patch = i_patch + 1;

end
 fprintf('%d patches created.\n', i_patch-1);
```

**describe_initial_patches()**

```
function patches = describe_initial_patches(patches, image, options)

if strcmp(options.feature_type, 'color_hist')
    c = makecform('srgb2lab');
    image = applycform(image, c);
    image = image(:,:,2:3);
end

for i=1:size(patches,2)
    roi = int16(patches(i).roi);
    roi_image = image(roi(2,1):roi(2,2), roi(1,1):roi(1,2), :);
  if strcmp(options.feature_type, 'color_hist')
        patches(i).hists = image2color_hist(roi_image, options);
    elseif strcmp(options.feature_type, 'covariance')
        [patches(i).mean, patches(i).covariance] = image2covariance(roi_image);
    end
end
```

**gmmtest()**

```
function [theta_final, J_test, probJ, S_final, final_moments,...
         final_moments_grad, bandw, var_theta, std_theta,conf_inter] = gmmest(options, data, popmom, startval,
We, varargin);

if nargin<5, error('The first four inputs (data, popmom, stval, W) must be provided by the user');end
[stvr,stvc] = size(startval);
if stvc~=1, error('The starting values must be a column vector');end
pmc=feval(popmom, startval,data, varargin{:});
[~,q]=size(pmc);
if stvr>q, error('The system is under-identified. You must supply at least as many moment conditions as
parameters.');end
center    = optget('gmmest','center',0);
method    = optget('gmmest','method','SerUnc');
bandw     = optget('gmmest','bandw',[]);
itergmm   = optget('gmmest','itergmm',50);
tol       = optget('gmmest','tol',1e-006);
theta = fminunc('gobj', startval, options, popmom, data, We, varargin{:});
for i=2:itergmm
    pmc = feval(popmom, theta,data, varargin{:});
    S = longvar(pmc, center, method, bandw);
    invS = S\eye(size(S,1)); %
    thetanew = fminunc('gobj', theta,...
        options, popmom, data, invS, varargin{:});
    if norm(abs(theta - thetanew)) < tol
```

```
        result = sprintf('The algorithm converged to a solution. The optimal estimator was achieved in iteration
%2.0f .', i);
        disp(result);
        break
    end
    theta = thetanew;
end
if exist('result','var')==0 & itergmm~=1
    disp('The algorithm didn''t converged to a solution.' );
end

if itergmm == 1
    thetanew = theta;
    pmc = feval(popmom, thetanew, data, varargin{:});
    S = longvar(pmc, center, method, bandw);
    disp('One step GMM estimation: Completed');
    disp('J-test has a non-standard distribution; p-value is not calculated');
end

theta_final = thetanew;
if nargout>1
    [pmc,dpmc] = feval(popmom, theta_final,data, varargin{:});
    W_final = S\eye(size(S,1));
    [S_final, bandw] = longvar(pmc, center, method, bandw);
    [N, pc] = size(pmc);
    df = pc - stvr;
    final_moments = pmc;
    final_moments_grad = dpmc;
    if itergmm==1
        J_test = gobj(theta_final, popmom, data, We, varargin{:});
        [VAR,SD,CI] = varest(dpmc, S_final, theta_final, N, We,1);
        probJ = [];
    else
        [VAR,SD,CI] = varest(dpmc, S_final, theta_final, N,We,itergmm);
        J_test = gobj(theta_final, popmom, data, inv(S_final), varargin{:});
        probJ = 1-chi2cdf(J_test, df);
    end
    var_theta = VAR;
    std_theta = SD;
    conf_inter = CI;
end
if isempty(optget('gmmest', 'bandw')) & lower(optget('gmmest', 'method'))~='serunc'
    message = sprintf('The optimum bandwidth, has been set to %4.0f', bandw);
    disp(message);
end
```

**meanshift()**

```
function [Y1,C1]= meanshift(I,Ht,Y0,width,height,Kernel,CH1,CH2);

    step = inf;
    simThresh = 0.05;
    nstep=0;
    while( step > eps)
        nstep=nstep+1;
        [Cand Y0] = cropAt(I, Y0, width, height);
        Hc = pHist2D( Cand(:, :, CH1), Cand(: ,: , CH2), Kernel, 1);
        C0 = bhatCoef(Ht, Hc);
        numerator   = 0;
        denominator = 0;
```

```
        H = round([width/2; height/2]);
        for x = 1:width,
            for y = 1:height,
                X = ([x;y] - H);
                ht = histvalue(Ht, Cand(y, x, CH1), Cand(y, x, CH2));
                hc = histvalue(Hc, Cand(y, x, CH1), Cand(y, x, CH2));
                if hc ~= 0,
                    w = sqrt( ht / hc);
                else
                    w = 0;
                end
                numerator   = numerator + (X .* w);
                denominator = denominator + (w );
            end
        end
        Y1 = round(numerator ./ denominator);
        Y1 = Y0 + Y1;

        [Cand Y1] = cropAt(I, Y1, width, height);
        Hc = pHist2D( Cand(:, :, CH1), Cand(: ,: , CH2), Kernel, 1);
        C1 = bhatCoef(Ht, Hc);
        while (C1 + simThresh) < C0 & Y0~=Y1,
            Y1 = floor(1/2 * (Y0 + Y1));
            [Cand Y1]= cropAt(I, Y1, width, height);
            Hc = pHist2D( Cand(:, :, CH1), Cand(: ,: , CH2), Kernel, 1);
            C1 = bhatCoef(Ht, Hc);
        end
        step = sqrt((Y0(1) - Y1(1))^2 + (Y0(2) - Y1(2))^2);
        Y0 = Y1;
        if (nstep==20)
            break;
        end;
    end
end
```

**parse_xml_calibration_file()**

```
function [K, Rt] = parse_xml_calibration_file(filename)
 tree = parseXML(filename);
 focal = 0;
sx    = 0;
cx    = 0; cy    = 0;
dpx   = 0; dpy   = 0;
 t       = [0; 0; 0];
r_angles = [0; 0; 0];
 for c = tree.Children
    if strcmp(c.Name, 'Geometry')
        for a = c.Attributes
            if strcmp(a.Name, 'dpx');
                dpx = str2num(a.Value);
            elseif strcmp(a.Name, 'dpy');
                dpy = str2num(a.Value);
            end
        end
    elseif strcmp(c.Name, 'Intrinsic')
        for a = c.Attributes
            if strcmp(a.Name, 'focal');
                focal = str2num(a.Value);
            elseif strcmp(a.Name, 'cx');
```

```
            cx = str2num(a.Value);
        elseif strcmp(a.Name, 'cy');
            cy = str2num(a.Value);
        elseif strcmp(a.Name, 'sx');
            sx = str2num(a.Value);
        end
    end
  elseif strcmp(c.Name, 'Extrinsic')
    for a = c.Attributes
        if strcmp(a.Name, 'tx')
            t(1) = str2num(a.Value);
        elseif strcmp(a.Name, 'ty')
            t(2) = str2num(a.Value);
        elseif strcmp(a.Name, 'tz')
            t(3) = str2num(a.Value);
        elseif strcmp(a.Name, 'rx')
            r_angles(1) = str2num(a.Value);
        elseif strcmp(a.Name, 'ry')
            r_angles(2) = str2num(a.Value);
        elseif strcmp(a.Name, 'rz')
            r_angles(3) = str2num(a.Value);
        end
    end
  end
end
R = angle2matrix(r_angles(1), r_angles(2), r_angles(3));
Rt = [R t];
K = [(focal*sx)/dpx          0      cx; ...
              0     focal/dpy     cy; ...
              0             0      1];
```

**parseXML()**

```
function theStruct = parseXML(filename)
try
   tree = xmlread('pets.xml');
catch
   error('Failed to read XML file %s.',filename);
end
try
   theStruct = parseChildNodes(tree);
catch
   error('Unable to parse XML file %s.',filename);
end
function children = parseChildNodes(theNode)
children = [];
if theNode.hasChildNodes
   childNodes = theNode.getChildNodes;
   numChildNodes = childNodes.getLength;
   allocCell = cell(1, numChildNodes);
    children = struct(           ...
      'Name', allocCell, 'Attributes', allocCell,    ...
      'Data', allocCell, 'Children', allocCell);

    for count = 1:numChildNodes
        theChild = childNodes.item(count-1);
        children(count) = makeStructFromNode(theChild);
    end
end
function nodeStruct = makeStructFromNode(theNode)
```

```
nodeStruct = struct(                    ...
   'Name', char(theNode.getNodeName),       ...
   'Attributes', parseAttributes(theNode),  ...
   'Data', '',                      ...
   'Children', parseChildNodes(theNode));

if any(strcmp(methods(theNode), 'getData'))
   nodeStruct.Data = char(theNode.getData);
else
   nodeStruct.Data = '';
end
function attributes = parseAttributes(theNode)
attributes = [];
if theNode.hasAttributes
   theAttributes = theNode.getAttributes;
   numAttributes = theAttributes.getLength;
   allocCell = cell(1, numAttributes);
   attributes = struct('Name', allocCell, 'Value', ...
                      allocCell);
    for count = 1:numAttributes
      attrib = theAttributes.item(count-1);
      attributes(count).Name = char(attrib.getName);
      attributes(count).Value = char(attrib.getValue);
   end
end
```

**sample_quadrilateral()**

```
function pts = sample_quadrilateral(points, sample_step)
 if nargin == 1
    sample_step = 2;
end
 p1 = points(:,1); p2 = points(:,4);
n_points = floor(norm(p1 - p2)/sample_step);

if n_points == 1
   pts = points;
else
   step_t = 1/(n_points-1);
    pts_r = [];
    for t=0:step_t:1
        pts_r = [pts_r [t*p1 + (1 - t)*p2]];
    end
   p1 = points(:,2); p2 = points(:,3);
    pts_l = [];
    for t=0:step_t:1
        pts_l = [pts_l [t*p1 + (1 - t)*p2]];
    end
    pts = [];
    p1 = points(:,3); p2 = points(:,4);
   n_v_points = floor(norm(p1 - p2)/sample_step);
    step_t = 1/(n_v_points-1);
    for i=1:n_points
        p1 = pts_l(:,i);
        p2 = pts_r(:,i);
        for t=0:step_t:1
            pts = [pts [t*p1 + (1 - t)*p2]];
        end
    end
```

```
end
pts = round(pts);
```

**scale_patches()**

```
function tracker = scale_patches(tracker, scale)
for i = 1:size(tracker.patches,2);
    roi = tracker.patches(i).roi;
    min_x = roi(1,1); max_x = roi(1,2);
    min_y = roi(2,1); max_y = roi(2,2);
    cx = (max_x + min_x)/2;
    cy = (max_y + min_y)/2;
    w     = max_x - min_x;
    new_w = scale*w;
    hw = new_w/2;
    min_x = cx - hw;
    max_x = cx + hw;
    h     = max_y - min_y;
    new_h = scale*h;
    min_y = max_y - new_h;
    roi = [min_x, max_x; min_y max_y];
    tracker.patches(i).roi = roi;

end
tracker.w = new_w;
for i=size(tracker.patches,2)-1:-1:1
    roi      = tracker.patches(i).roi;
    next_roi = tracker.patches(i+1).roi;
    min_y = roi(2,1); max_y = roi(2,2);
    n_min_y = next_roi(2,1);
    dy = n_min_y - max_y;
    min_y = min_y + dy;
    max_y = max_y + dy;
    tracker.patches(i).roi(2,1) = min_y;
    tracker.patches(i).roi(2,2) = max_y;
end
```

**sstest()**

```
function [theta1, theta2, Wss, LMss, Dss, O, O1, O2] = ...
    sstests(options, data, popmom, thetaf, Sfull, Df, bp, varargin)

center   = optget('gmmest','center');
method   = optget('gmmest','method');
bandw    = optget('gmmest','bandw');
itergmm  = optget('gmmest','itergmm');
tol      = optget('gmmest','tol');
if nargin<7
    error('The first 7 inputs must be provided')
end
 [pmc,dpmc] = feval(popmom, thetaf,data, varargin{:});
T = size(pmc,1);
pit = bp/T;
p = size(thetaf, 1);
data1  = data(1:bp,:);
data2  = data(bp+1:T,:);
if nargin == 8
    INSTR_USED  = varargin{1};
    INSTR_USED1 = INSTR_USED(1:bp,:);
```

```
    INSTR_USED2 = INSTR_USED(bp+1:T,:);
    disp('First step estimates:');
    theta1  = gmmest(options, data1, popmom, thetaf, Sfull, INSTR_USED1);
    [m1u,D1u] = feval(popmom, theta1, data1, INSTR_USED1);
    [m1r,D1r] = feval(popmom, thetaf, data1, INSTR_USED1);
    S1u = longvar(m1u, center, method, bandw);
    invS1u = inv(S1u);
    Q1u = gobj(theta1, popmom, data1, invS1u, INSTR_USED1);
    Q1r = gobj(thetaf, popmom, data1, invS1u, INSTR_USED1);
    disp('Second step estimates:');
    theta2  = gmmest(options, data2, popmom, thetaf, Sfull, INSTR_USED2);
    [m2u,D2u] = feval(popmom, theta2, data2, INSTR_USED2);
    [m2r,D2r] = feval(popmom, thetaf, data2, INSTR_USED2);
    S2u = longvar(m2u, center, method, bandw);
    invS2u = inv(S2u);
    Q2u = gobj(theta2, popmom, data2, invS2u, INSTR_USED2);
    Q2r = gobj(thetaf, popmom, data2, invS2u, INSTR_USED2);
else
    disp('First step estimates:');
    theta1  = gmmest(options, data1, popmom, thetaf, Sfull);
    [m1u,D1u] = feval(popmom, theta1, data1);
    [m1r,D1r] = feval(popmom, thetaf, data1);
    S1u = longvar(m1u, center, method, bandw);
    invS1u = inv(S1u);
    Q1u = gobj(theta1, popmom, data1, invS1u);
    Q1r = gobj(thetaf, popmom, data1, invS1u);
    disp('Second step estimates:');
    theta2  = gmmest(options, data2, popmom, thetaf, Sfull);
    [m2u,D2u] = feval(popmom, theta2, data2);
    [m2r,D2r] = feval(popmom, thetaf, data2);
    S2u = longvar(m2u, center, method, bandw);
    invS2u = invS2u;
    Q2u = gobj(theta2, popmom, data2, invS2u);
    Q2r = gobj(thetaf, popmom, data2, invS2u);
end
VW1 = D1u'*invS1u*D1u;
VW2 = D2u'*invS2u*D2u;
VW  = (1/pit)*inv(VW1)+(1/(1-pit))*inv(VW2);
c   = theta1-theta2;
Wss.value = T*c'*inv(VW)*c;
Wss.prob = 1-chi2cdf(Wss.value,p);
g1  = pit*(sum(m1r)/size(m1r,1))';
invS  = inv(Sfull);
LMv = Df'*invS*g1;
sc  = T/(pit*(1-pit));
LMss.value = sc*LMv'*inv(Df'*invS*Df)*LMv;
LMss.prob = 1-chi2cdf(LMss.value,p);
Dss.value   = (Q1r-Q1u)+(Q2r-Q2u);
Dss.prob = 1 - chi2cdf(Dss.value,p);
q = size(m1r, 2);
O1.value = Q1u;
O2.value = Q2u;
O.value = Q1u + Q2u;
dfo = 2*(q-p);
O.prob = 1-chi2cdf(O.value,dfo);
O1.prob = 1-chi2cdf(O1.value,dfo/2);
O2.prob = 1-chi2cdf(O2.value,dfo/2);
```

# ACKNOWLEGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am highly indebted to the University Grant Commission-SERO, Hyderabad for considering my proposal and providing financial assistance.

I am highly indebted to Prof.V.V. Nageswra Rao, Director of Aditya Institute of Technology and Management (A) and Dr. A. Srinivasa Rao, Principal for the facilities provided to accomplish this project. I am obliged to Dr. K.B.Madhu Sahu, Director R&D and R&D team for their encouragement.

I am thankful to Dr. G. Sateesh Kumar, Head of the Department, Department of ECE for providing necessary facilities in the department. I am also thankful to all the senior members of the department for their constructive criticism throughout the project.

My heartful thanks are due to all my colleagues in the department for their keen interest and great support throughout the project.

I express my gratitude to nonteaching staff of our department for their cooperation throughout the project.

Finally, I express my heartful thanks to all of my family members who helped me in successful completion of this project.

Mr. Harihara Santosh Dadi

Principal Investigator

Technical report submitted to UGC on

# Development of New Algorithm for Detection of Terrorists-Robbers-Lost People and Suspicious People in Public Places

UGC Minor Research Project sanction **No.F MRP- 6950/16 (SERO/UGC)-dated 02-08-2017**

**Link No: 6950**

Submitted by

**Mr. Harihara Santosh Dadi**

Principal Investigator

Assoc. Professor, Dept. of Electronics and Communication Engineering

**Department of Electronics and Communication Engineering**

**Aditya Institute of Technology and Management (A)**

**Approved by AICTE, Permanently Affiliated to JNTUK Kakinada, Accredited by NBA, Accredited by NAAC (UGC) with A+, Recognized by UGC u/s 2(f) & 12 (B), Recognized as SIRO by DSIR**

**Tekkali, Srikakulam-532201, A.P., India.**

## INDEX

# CERTIFICATE

This is to certify that the present work titled "**Development of New Algorithm for Detection of Terrorists-Robbers-Lost People and Suspicious People in Public Places**" is carried out by me and was not submitted for full/ partial financial assistance to any other funding agency.

Mr. Harihara Santosh Dadi

Principal Investigator

Place: AITAM, TEKKALI.

Date: 24/07/2019

**Annexure – VII**

**UNIVERSITY GRANTS COMMISSION**

**BAHADUR SHAH ZAFAR MARG**

**NEW DELHI – 110 002**

**PROFORMA FOR SUBMISSION OF INFORMATION AT THE TIME OF SENDING THE**

**FINAL REPORT OF THE WORK DONE ON THE PROJECT**

**1. Title of the Project**: Development of new algorithm for detection of terrorists – robbers – lost people and suspicious people in public places

**2. NAME AND ADDRESS OF THE PRINCIPAL INVESTIGATOR:**

Mr. Harihara Santosh Dadi,

Assoc. Professor, Department of ECE),

Aditya Institute of Technology and Management (A),

K.Kotturu (Vi), Tekkali – 532201. Srikakulam District. Andhra Pradesh.

**3. NAME AND ADDRESS OF THE INSTITUTION:**

Aditya Institute of Technology and Management (A),

K.Kotturu (Vi), Tekkali – 532201. Srikakulam District. Andhra Pradesh.

**4. UGC APPROVAL LETTER NO. AND DATE**:

MRP-6950/16 (SERO/UGC) Dated 02/08/2017

**5. DATE OF IMPLEMENTATION** **:** 18 – 08 -2017

**6. TENURE OF THE PROJECT** **:** 02 Years

**7. TOTAL GRANT ALLOCATED** **:** Rs. 2,61,500/-

**8. TOTAL GRANT RECEIVED** **:** Rs. 2,36,500/-

**9. FINAL EXPENDITURE** **:** Rs. 2,72,956/-

**10. TITLE OF THE PROJECT:**

Development of new algorithm for detection of terrorists – robbers – lost people and suspicious people in public places

**11. OBJECTIVES OF THE PROJECT :**

- To develop Color and shape based human detection algorithm for detection of humans while tracking.

- Develop Gaussian Mixture Model based pedestrian tracking algorithm.
- Develop HOG based face feature extractor.
- Based on the existing datasets of terrorists, robbers and lost people, classify the detected human as whether the person is terrorists, robber, suspicious person or lost person.

**12. WHETHER OBJECTIVES WERE ACHIEVED :** Yes. Details are available in the report.
**(GIVE DETAILS)**

**13. ACHIEVEMENTS FROM THE PROJECT:**

- Color and Shape based human detection algorithm is developed.
- HOG and SVM based face recognition algorithm is applied for AITAM1, AITAM2 and AITAM3 data sets with robbers, suspicious persons, terrorists and lost persons.
- By the combination of human detection algorithm and face recognition algorithm, high face recognition rate was achieved in public places in surveillance cameras.

**14. SUMMARY OF THE FINDINGS: ( IN 500 WORDS )**

The proposed color and shape based automated human detection algorithm is producing good human detection metrics for standard datasets namely PETS 2009 and also for the developed datasets in this work. The detected human’s face features are extracted and given to the SVM classifier for training. After the human is detected and recognized either as the robber, suspicious person, terrorists or lost person, the human is tracked and the path is formed. Three datasets namely AITAM1, AITAM2 and AITAM3 with different levels of complexity are developed. In each dataset, combination of all four category people are considered. In AITAM1 dataset has a robber and a suspicious person. AITAM2 dataset has a robber, a suspicious person and a lost person and AITAM3 dataset has a robber, lost person, terrorist and two suspicious persons. The proposed algorithm is producing better face recognition rate. The proposed algorithm reduced the processing time, without compromising the other performance metrics. Quantitative metrics are calculated for GMM algorithm. Performance results show that the algorithm is working well, if the dataset environment is either occlusive or cluttered or contrast or illuminative. The combination of recognizing and tracking the person makes the human tracking systems intelligent and more authenticated.

**15. CONTRIBUTION TO THE SOCIETY: (GIVE DETAILS )**

- Terrorists – robbers – lost people and suspicious people can be easily identified though they are wandering in huge crowded areas/ hidden places.
- Presently developed algorithm is useful to Police and defense personnel.

**16. WHETHER ANY PH.D. ENROLLED/PRODUCED OUT OF THE PROJECT:**

Principal Investigator Submitted his Ph.D thesis in JNTUH Hyderabad in an allied area.

**17. NO. OF PUBLICATIONS OUT OF THE PROJECT:( PLEASE ATTACH)**

Results were received very recently. Publication is under process.

**( PRINCIPAL INVESTIGATOR )** **(PRINCIPAL)**

**(Seal)**

**Annexure -VI**

**UNIVERSITY GRANTS COMMISSION**

**BAHADUR SHAH ZAFAR MARG**

**NEW DELHI – 110 002.**

**Annual/Final Report of the work done on the Minor Research Project.**

**(Report to be submitted within 6 weeks after completion of each year)**

1. Project report No. 1st /Final : Year - 1

2. UGC Reference No.F. : **MRP-6950/17 (SERO/UGC)** dated 02/08/2017

3. Period of report: from : 18/08/2017 to 17/08/2018

4. Title of research project : Development of new algorithm for detection of terrorists-robbers-lost people and suspicious people in public places

5. (a) Name of the Principal Investigator : Mr. Harihara Santosh Dadi

(b) Deptt. : Electronics and Communication Engineering

(c) College where work has progressed: **ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, TEKKALI. SRIKAKULAM DIST. ANDHRA PRADESH.**

6. Effective date of starting of the project : 18/08/2017

7. Grant approved and expenditure incurred during the period of the report:

a. Total amount approved Rs. 2,61,500/-

b. Total expenditure Rs. 2, 43, 594.97/-

c. Report of the work done: (Please attach a separate sheet)

i. Brief objective of the project:

- ✓ To detect, Identify, classify and track the robbers, suspicious people and thieves in public places using surveillance cameras.

ii. Work done so far and results achieved and publications, if any, resulting from the work (Give details of the papers and names of the journals in which it has been published or accepted for publication________________

- Developed HOG based face feature extractor. These features are given to the SVM classifier for classification. AT&T dataset is taken and testing. The face recognition rate is considerably high when compared with the other existing face recognition algorithms.
- Developed Gaussian Mixture Model based pedestrian tracking algorithm. PETS 2009 dataset is taken for testing. There is a substantial improvement in the Performance metrics of the developed GMM based object tracking algorithm.
- Paper publication is under progress.

iii. Has the progress been according to original plan of work and towards achieving?
the objective. if not, state reasons

- The progress is according to the original plan of work and towards achieving the objective.

iv. Please enclose a summary of the findings of the study. One bound copy of the
final report of work done may also be sent to the concerned Regional Office of the
UGC.

The proposed color and shape based automated human detection algorithm is producing good human detection metrics for all kinds of datasets. The detected human's face features are extracted and given to the SVM classifier for training. After the human is detected and recognized, the human is tracked and the path is formed. Three datasets namely AITAM1, AITAM2 and AITAM3 with different levels of complexity are developed. The proposed algorithm is producing better face recognition rate. Weighted Running Window Background model based Gaussian Mixture Model is developed for human tracking. Instead of considering the entire history of pixels for modeling the pixel in the current frame, only a window of past pixels with different weights are considered. Performance results show that the algorithm reduced the process time for tracking. AITAM2 dataset considered for this experiment has 688 frames and the time taken for processing these frames is only 1 minute and 15 seconds. Whereas, the GMM algorithm takes 1 minute 57 seconds. The proposed algorithm reduced the processing time, without compromising the other performance metrics. Both qualitative and quantitative metrics are calculated for WRWGMM algorithm and compared with GMM algorithm. Performance results show that the algorithm is working well, if the dataset environment is either occlusive or cluttered or contrast or illuminative. The proposed algorithm is also working well on non stationary camera videos. The combination of recognizing and tracking the person makes the human tracking systems intelligent and more authenticated.

v. Any other information: No.

**SIGNATURE OF THE PRINCIPAL INVESTIGATOR** **PRINCIPAL**

**(Seal)**

# Quantitative Performance Metrics for Human Tracking Algorithms

Dr. Harihara Santosh Dadi

Associate Professor, ECE Department, AITAM, Tekkali

***Abstract*—Now a days, human tracking algorithms getting wide attention in the field of computer vision. In this paper, different human tracking algorithms are studied and compared its performance metrics. Quantitative performance metrics like FP, FN, Precision, Recall, IDSW, FaF are considered. Performance results show that WRWGMM algorithm is working better on PETS 2009 dataset.**

***Index Terms*—False Positives, False Negatives, Identity Switches, Weighted Running Window Background Model based GMM.**

## I. Introduction

The aim of computer vision is to allow computers understand a scene by using digital cameras. In current years, video evaluation is becoming one of the utmost popular studies in the field of computer vision. At any given second, thousands of videos are uploaded to the database and every year, millions of surveillance cameras around the arena are capturing trillions of hours of video. It is impossible for humans to comprehend such massive quantities of video information. Consequently, efficient vision based algorithms for human detection, tracking, segmentation, motion popularity, video retrieval, strange occasion detection and video summarization are getting increasingly more crucial. In any tracking system, detection of object and object tracking are two fundamental tasks. These are used widely in numerous applications like video surveillance, video communication, medical imaging, traffic control, security, augmented reality and human-computer interaction.

Object detection is the method of detecting the times of a certain class of entities (e.g. people, dogs, and bicycles) in pictures and surveillance videos.

The methodology of locating the stationary objects in various frames of video while keeping the identities of the objects properly is called object tracking. In video surveillance or object tracking systems, detection of object and tracking of object are usually combined together to find the trajectory of the object in the video. Pedestrians in every video frame are to be identified first, after which, tracking them across specific frames is done. We know that the object of recognition is human beings, for the reason that human beings are maximum probable to be the interested objects in applications inclusive of visual surveillance, human laptop interplay, and self sufficient vehicle navigation. Object segmentation is also a vital undertaking in computer vision. Object segmentation is the method of defining the target object from the image.

The existing GMM algorithm track the object based on the pixel history over the frames. Every pixel in the region of interest is modeled either as a background or foreground and the decision is taken based on the cluster of foreground and background pixels.

## II. Literature Review

Elgammal et al., [1] presented a novel non-parametric background subtraction approach. The model proposed has the advantage of handling situations where there are cluttered background scenes, when the background is not completely static and contains small variations like movement of tree branches etc.. The proposed model can able to estimate the observing pixel probabilities by the intensities of every pixel in the image. The model quickly adapts to the variations in the environment that enables detection of sensitive moving objects. This method uses the information regarding the color content to eliminate the shadows. This technique reaches very minute detection with less number of false alarms per frame. Based on the monochrome intensity values of the pixels in the image, the proposed method can also estimate the probability of the intensity of the pixel under observation. However, in modeling the pixel, the method takes all the past history of pixels in to consideration which takes longer time to estimate the probability.

P. Kaew Tra Kul Pong et al., [2] proposed an improved adaptive background mixture model is presented in this paper. This system learns faster, more accurately and adapt effectively to changing environment. This algorithm discriminates the shadows and moving objects. The authors reinvestigated the update equation and also used different equations at different phases. It overcomes the limitations of multi-color background model proposed by Grimson et al [3]. By incorporating the shadow detection, the proposed method produces better results when compared with Grimson's method. However, the method is suitable only for single object tracking. When comes to multiple object tracking, it learns very slowly.

Z. Zivkovic et al., [4] proposed pixel-level approach for modeling the background. Each pixel is modeled as mixture of Gaussian probability density functions. Recursive equations are used to constantly update the mean, covariance matrix and also the number of Gaussian components. However, there is no limit in considering the number of past pixels in modeling the present pixel. This makes the algorithm diverge with no solution.

J. McHugh et al, [5] proposed a threshold value which is adaptive which depends on the video statistics. It switches between the nonparametric and mixture of Gaussians models. A foreground model based on small spatial neighborhood to improve discrimination sensitivity is also introduced. However, finding the video statistics and choosing the threshold value by means of the two statistical models is expensive. For simple environments like pedestrian tracking videos, instead of adaptive threshold value, either mixture of Gaussians or any nonparametric model is recommended.

Liao S et al., [6] proposed a novel background subtraction framework by taking the illumination variations and dynamic backgrounds in to consideration. The initial step is, scale independent kernel is developed for handling changes in the lighting conditions. After that, the mask based on the shape observation is developed to efficiently handle the possibility allocation of shapes in the processing of intensity values. The last method is, development of multitude background based techniques for tackling the adaptive critical background models. However, the proposed techniques are for particular illumination variations and particular dynamic backgrounds, which is not the case in real-time scenarios

## III. QUANTITATIVE METRICS

There are various quantitative metrics available for measuring the performance of tracking algorithm. All the metrics can be found by first forming the confusion matrix.

### A. *Confusion Matrix*

Confusion matrix is a table which contains two rows and two columns that reports the number of true positives, true negatives, false negatives and false positives shown in Fig: 7.1.

| | | True Condition | |
|---|---|---|---|
| | | Condition Positive | Condition Negative |
| Predicted Condition | Predicted Condition Positive | TRUE POSITIVE | FALSE POSITIVE |
| | Predicted Condition Negative | FALSE NEGATIVE | TRUE NEGATIVE |

Fig: 1 Confusion matrix

### B. *True Positives*

True positive means the overlap between the ground truth and the algorithm output is greater than the threshold. If the threshold value is less, it means the confidence is high. If the thresholding value is more, it means that the confidence is less. A specific threshold value has to be fixed for comparison of the tracking algorithms. In this experiment, the value chosen for threshold is 0.25.

### C. *False Positives*

False positive means that there is no entry in the ground truth database, but still there is a result showing by the algorithm.

### D. *False Negatives*

False negative means even though there is ground truth entry, still there is no result by the algorithm.

### E. *True Negatives*

If there is no entry in the ground truth and also there is no entry by the algorithm, it is counted as true negative.

When the overlap is less than the fixed threshold value, it is counted as both false negative and false positive.

### F. *Recall*

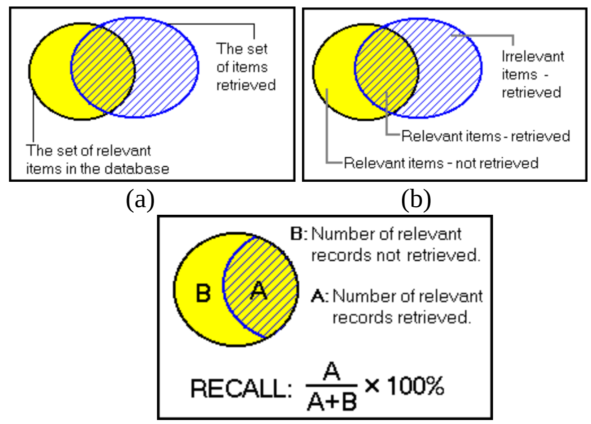



Fig. 2 (a), (b) and (c) Venn diagrams of Recall

Recall is calculated as the amount of information retrieved which is relevant to the total amount of retrieved information present in the database. It is also called as sensitivity.

### G. *Precision*

Precision is calculated as the amount of information retrieved which is relevant to the total amount of relevant and irrelevant information present in the database. It is also called as positive predictive value.

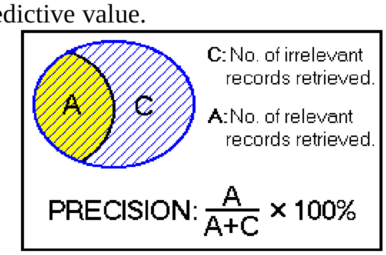



Fig: 3 Venn diagram for Precision

### H. *False Alarm per Frame*

It is the average of the false alarms that appear per frame. The ideal value of false alarm per frame is zero. If there is any identity switch going to happen for sufficient number of frames, it is termed as false alarms. It is a complete switch from one object to the other. It is only an alarm.

## IV. RESULTS

Fig.s 4 – 10 and the table 1 shows the algorithm proposed outperforming in all the parameters.

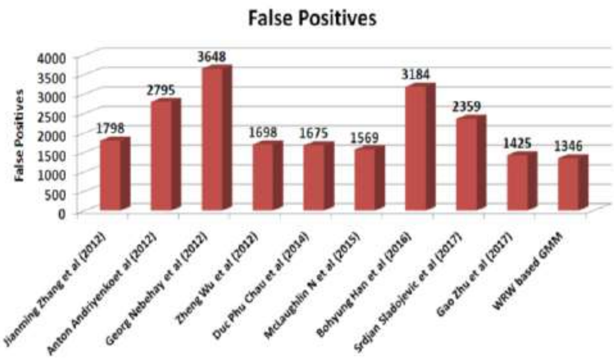



Fig: 4 Comparison of false positives with the state-of-the-art tracking algorithms on PETS 2009 dataset

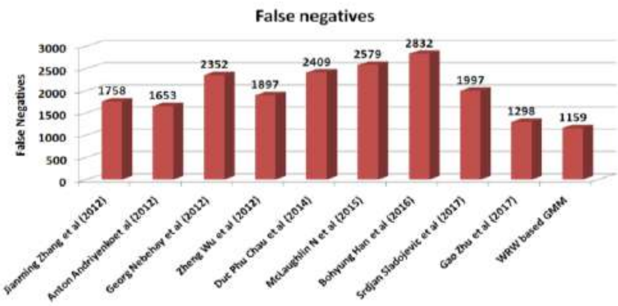



Fig: 5 Comparison of false negatives with the state-of-the-art tracking algorithms on PETS 2009 dataset

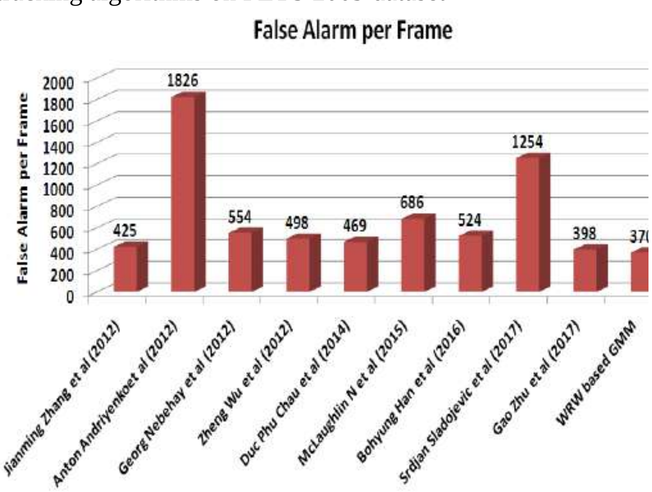



Fig: 6 Comparison of false alarm per frame with the state-of-the-art tracking algorithms on PETS 2009 dataset

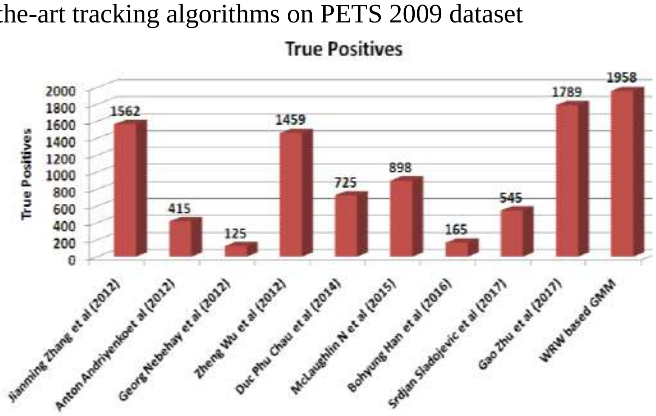



Fig: 7 Comparison of true positives with the state-of-the-art tracking algorithms on PETS 2009 dataset

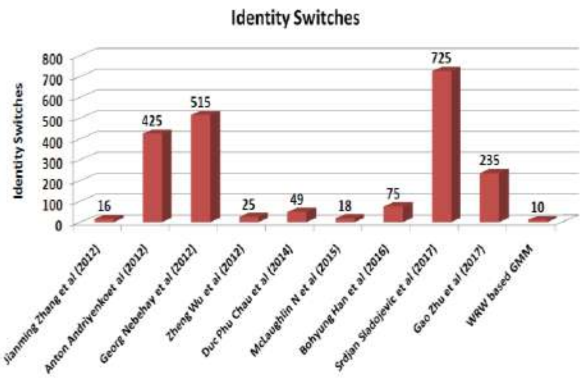



Fig: 8 Comparison of identity switches with the state-of-the-art tracking algorithms on PETS 2009 dataset

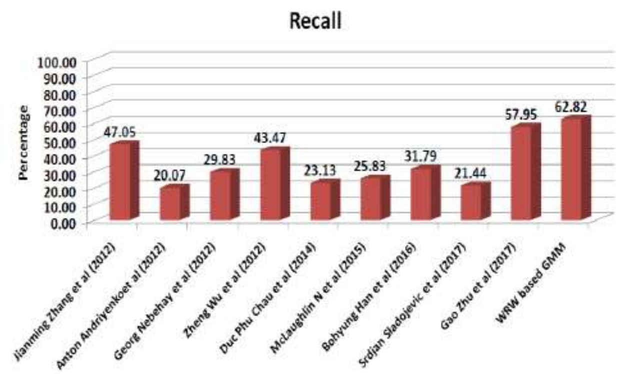



Fig: 9 Comparison of recall with the state-of-the-art tracking algorithms on PETS 2009 dataset

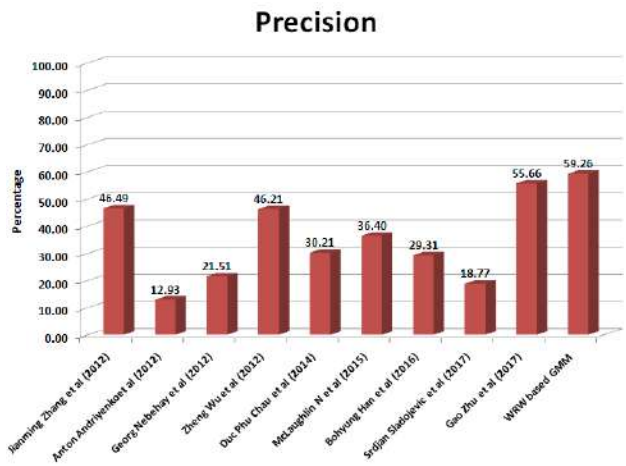



Fig: 10 Comparison of precision with the state-of-the-art tracking algorithms on PETS 2009 dataset

Table: 1 Comparison of performance metrics of state-of-the-art tracking algorithms on PETS 2009 dataset

| S. No. | Author | TP | FP | FN | PRECISION (%) | RECALL (%) | FAF | IDSW |
|---|---|---|---|---|---|---|---|---|
| 1 | Jianming Zhang et al [7] | 1562 | 1798 | 1758 | 46.49 | 47.05 | 425 | 16 |
| 2 | Anton Andriyenkoet al [8] | 415 | 2795 | 1653 | 12.93 | 20.07 | 1826 | 425 |
| 3 | Georg Nebehay et al [9] | 125 | 3648 | 2352 | 21.51 | 29.83 | 554 | 515 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | Zheng Wu et al [10] | 1459 | 1698 | 1897 | 46.21 | 43.47 | 498 | 25 |
| 5 | Duc Phu Chau et al [11] | 725 | 1675 | 2409 | 30.21 | 23.13 | 469 | 49 |
| 6 | McLaughlin N et al [12] | 898 | 1569 | 2579 | 36.40 | 25.83 | 686 | 18 |
| 7 | Bohyung Han et al [13] | 165 | 3184 | 2832 | 29.31 | 31.79 | 524 | 75 |
| 8 | Srdjan Sladojevic et al [14] | 545 | 2359 | 1997 | 18.77 | 21.44 | 1254 | 725 |
| 9 | Gao Zhu et al [15] | 1789 | 1425 | 1298 | 55.66 | 57.95 | 398 | 235 |
| 10 | WRW based GMM | 1958 | 1346 | 1159 | 59.26 | 62.82 | 370 | 10 |


Fig. 11 Sample frame from online multi-person tracking by tracker hierarchy

Fig. 12 Sample frame from discrete-continuous optimization for multi-target tracking

Fig. 13 Sample frame from robust object tracking based on tracking-learning-detection

Fig. 14 Sample frame from coupling detection and data association for multiple object tracking

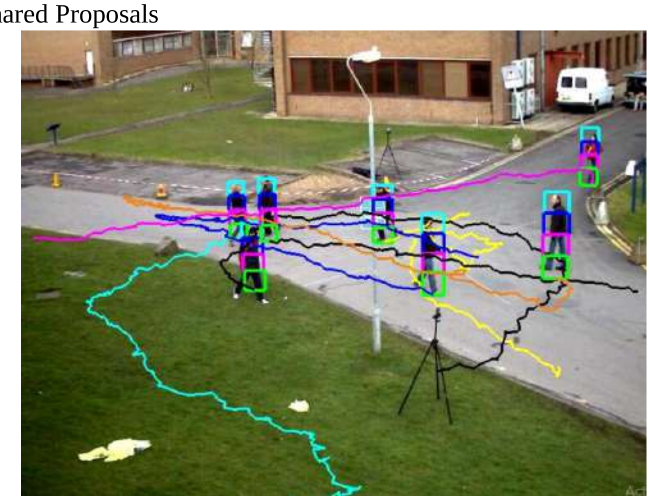
Fig. 15 Sample frame from Online Parameter Tuning for Object Tracking Algorithms

Fig. 16 Sample frame from Enhancing Linear Programming with Motion Modelling for Multi-target Tracking

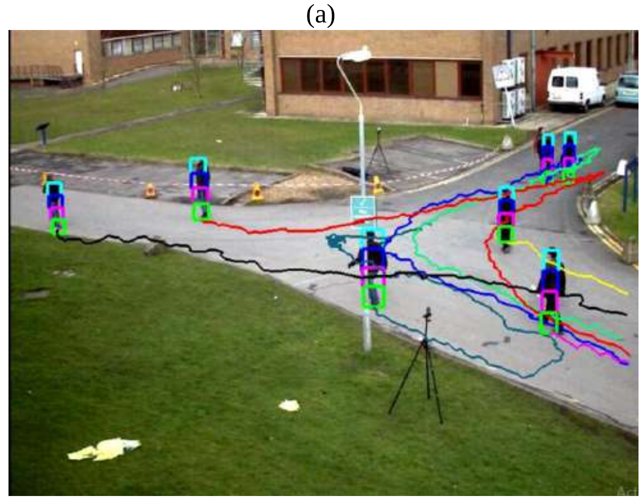
Fig. 17 Sample frame from Appearance-Based Human Tracking


Fig. 18 Sample frame from Integer arithmetic approximation of the HOG algorithm used for pedestrian detection

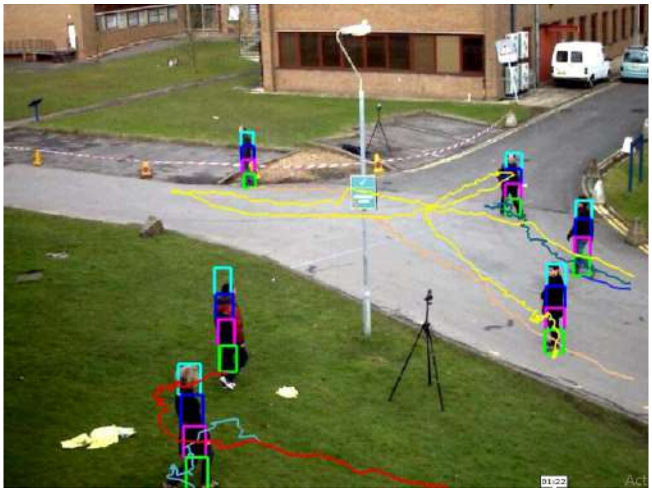
Fig. 19 Sample frame from Multiple Object Tracking with shared Proposals



(a)



(b)



(c)

Fig. 20 (a), (b) and (c) Sample frame from proposed algorithm

Fig. 11 to Fig. 19 shows the sample frames from the video outputs run on the PETS datasets. Fig. 20 show the sample frames from the proposed algorithm. Tracking results show that the algorithm is outperforming when compared with other existing algorithms.

## V. Conclusions

The existing state-of-the-art algorithms for human tracking in surveillance videos is compared quantitatively in this work. PETS 2009 dataset is taken for analysis. The performance parameters considered for comparison are false positives, false negatives, precision, recall, identity switches and false alarm per frame. Weighted Running Window Background model based GMM (WRWGMM) outperforms among all the existing human tracking algorithms.

## VI. Acknowledgments

The author gratefully acknowledges University Grants Commission SERO for their financial support. This work is a part of Minor Research Project sanctioned in the year 2017 with file no. UGC/SERO/6950. The author gratefully acknowledges Prof. Golpala Krishna Mohan Pillutla and Prof. Madhava Latha Makkena for guiding and mentoring for successfully completion of the project.

## VII. References

[1]
[2] Elgammal, A., Harwood, D., Davis, L. “Non-parametric model for background subtraction”, European Conference of Computer Vision, pp. 751–767 (2000).
[3] KaewTraKul Pong P, Bowden R (2001), “An improved adaptive background mixture model for realtime tracking with shadow detection”, proceedings of 2nd European workshop on advanced video based surveillance systems, AVBS01. Sept 2001. Video based surveillance systems: computer vision and distributed processing, Kluwer Academic Publishers, pp 1–5.
[4] W.E.L. Grimson, C. Stauffer, R. Romano and L. Lee, "Using adaptive tracking to classify and monitor activities in a site", Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 25-25 June 1998.
[5] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” Proc. ICPR, 2004.
[6] J. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin “Foreground-adaptive background subtraction, “ IEEE Signal Process. Lett., vol. 16, no. 5, pp. 390–393, May 2009.
[7] Liao S, Zhao G, Kellokumpu V, Pietikäinen M & Li SZ. (2010) “Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes”, Proceedings CVPR: 1301–1306.
[8] Jianming Zhang, Liliana Lo Presti, Stan Sclaroff, Online Multi-person Tracking by Tracker Hierarchy, 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, 18-21 Sept. 2012.
[9] Anton Andriyenko, Konrad Schindler, Stefan Roth, Discrete-continuous optimization for multi-target tracking, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 16-21 June 2012.
[10] Georg Nebehay, Robust Object Tracking Based on Tracking-Learning-Detection, Faculty of Informatics, TU, Vienna, 2012.
[11] Zheng Wu, Ashwin Thangali, Stan Sclaroff and Margrit Betke, Coupling Detection and Data Association for Multiple Object Tracking, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 16-21 June 2012.
[12] Duc Phu Chau, Monique Thonnat, François Brémond, Etienne Corvée, Online Parameter Tuning for Object Tracking Algorithms, Image and Vision Computing, Volume 32, Issue 4, April 2014, Pages 287-302.
[13] McLaughlin, N., Martinez Del Rincon, J., & Miller, P. (2015). Enhancing Linear Programming with Motion Modeling for Multi-target Tracking. In 2015 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 71-77). Institute of Electrical and Electronics Engineers Inc..
[14] Han B. (2014) Appearance-Based Human Tracking. In: Ikeuchi K. (eds) Computer Vision. Springer, Boston, MA
[15] Srdjan Sladojevic, Andras Anderla, Dubravko Culibrk, Darko Stefanović, Bojan Lalic, Integer arithmetic approximation of the HoG algorithm used for pedestrian detection, Computer Science and Information Systems 14(2):329-346.
[16] Zhu G., Porikli F., Li H. (2017) Model-Free Multiple Object Tracking with Shared Proposals. In: Lai SH., Lepetit V., Nishino K., Sato Y. (eds) Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science, vol 10112. Springer, Cham.
[17] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., “Performance Analysis of WRW Background model based GMM with other algorithms for tracking humans in surveillance videos” in Journal on Software Engineering, Volume 12, Issue 4, (April – June 2018) pp. 32 – 42.
[18] Gonzalez, R. C., Eddins, S. L., & Woods, R. E. (2004), Digital Image Publishing Using MATLAB. Prentice Hall.
[19] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., “Qualitative Performance Metrics for Object Tracking Algorithms” in Journal on Image Processing, Volume 4, Issue 4, (October - December 2017), pp. 1 – 15.
[20] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., “Human Tracking Under Severe Occlusions” in Journal on Software Engineering, Volume 12, Issue 1, (July - September 2017), pp.29 – 37.
[21] Hager, G. D., Dewan, M., & Stewart, C. V. (2004, June). Multiple kernel tracking with SSD. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.
[22] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., “Color and Shape Based Automatic Detection of Pedestrians in Surveillance Videos” in Journal on Image Processing, Volume 5, Issue 1, (January – March 2018) pp. 1 – 11.
[23] Birchfield, S. T., & Rangarajan, S. (2005, June). Spatiograms versus histograms for region-based tracking. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 1158-1163). IEEE.
[24] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., “Human Tracking Using Weighted Running Window Background Model Based GMM” in Journal on Software Engineering, Volume 12, Issue 2, (October - December 2017), pp. 44 – 54.

## VIII. Biographies

Mr. D. Harihara Santosh obtained his B.Tech., M.Tech and Ph.D., Degrees from JNT University, Hyderabad in the year 2005, 2010 and 2020. He is presently working as Associate Professor at Aditya Institute of Technology and Management, Tekkali. He has 22 publications in both International and National Journals and presented 20 papers at various International and National Conferences. His areas of interests are Image and Video Processing.

# Formulation of WRWGMM Algorithm for Tracking Humans

Harihara Santosh Dadi

Associate Professor, Dept. of ECE,

AITAM, Srikakluam

***Abstract*—Human tracking is the most challenging task in the field of computer vision. As the surveillance cameras are becoming the integrated part of humans, the need for human tracking is increasing for upgrading the existing surveillance systems. In the past decade, Gaussian Mixture Model (GMM) based human tracking proves to be the commanding because of its unique feature of pixel based background subtraction. The limitations of existing GMM based human tracking algorithm are presented in this work. A novel algorithm named Weighted Running Window based Gaussian Mixture Model to overcome the limitations of GMM is presented. The complete formulation of WRWGMM algorithm is also presented and compared with existing GMM algorithm on some frames of PETS 2009 dataset. Results of modeling of pixel (41,43) in 270$^{th}$ frame of PETS 2009 dataset shows that the number of iterations required are reduced from 78 to 2 and for pixel (41,184) from 33 to 3.**

***Index Terms*—Gaussian Mixture Model, Weighted Running Window based Gaussian Mixture Model, Back ground, Running Window.**

## I. Introduction

The aim of computer vision is to allow computers understand a scene by using digital cameras. In current years, video evaluation is becoming one of the utmost popular studies in the field of computer vision. At any given second, thousands of videos are uploaded to the database and every 12 months, millions of surveillance cameras around the arena are capturing trillions of hours of video. It is impossible for humans to comprehend such massive quantities of video information. Consequently, efficient vision based algorithms for human detection, tracking, segmentation, motion popularity, video retrieval, strange occasion detection and video summarization are getting increasingly more crucial. In any tracking system, detection of object and object tracking are two fundamental tasks. These are used widely in numerous applications like video surveillance, video communication, medical imaging [7], traffic control, security, augmented reality and human-computer interaction.

Object detection is the method of detecting the times of a certain class of entities (e.g. people, dogs, and bicycles) in pictures and surveillance videos.

The methodology of locating the stationary objects in various frames of video while keeping the identities of the objects properly is called object tracking. In video surveillance or object tracking systems, detection of object and tracking of object are usually combined together to find the trajectory of the object in the video. Pedestrians in every video frame are to be identified first, after which, tracking them across specific frames is done. We know that the object of recognition is human beings, for the reason that human beings are maximum probable to be the interested objects in applications inclusive of visual surveillance, human laptop interplay, and self sufficient vehicle navigation. Object segmentation is also a vital undertaking in computer vision. Object segmentation is the method of defining the target object from the image.

The existing GMM algorithm track the object based on the pixel history over the frames. Every pixel in the region of interest is modeled either as a background or foreground and the decision is taken based on the cluster of foreground and background pixels.

## II. Literature Review

Elgammal et al., [1] presented a novel non-parametric background subtraction approach. The model proposed has the advantage of handling situations where there are cluttered background scenes, when the background is not completely static and contains small variations like movement of tree branches etc.. The proposed model can able to estimate the observing pixel probabilities by the intensities of every pixel in the image. The model quickly adapts to the variations in the environment that enables detection of sensitive moving objects. This method uses the information regarding the color content to eliminate the shadows. This technique reaches very minute detection with less number of false alarms per frame. Based on the monochrome intensity values of the pixels in the image, the proposed method can also estimate the probability of the intensity of the pixel under observation. However, in modeling the pixel, the method takes all the past history of pixels in to consideration which takes longer time to estimate the probability.

P. Kaew Tra Kul Pong et al., [2] proposed an improved adaptive background mixture model is presented in this paper. This system learns faster, more accurately and adapt effectively to changing environment. This algorithm discriminates the shadows and moving objects. The authors reinvestigated the update equation and also used different equations at different phases. It overcomes the limitations of multi-color background model proposed by Grimson et al. By incorporating the shadow detection, the proposed method produces better results when compared with Grimson's method. However, the method is suitable only for single object tracking. When comes to multiple object tracking, it learns very slowly.

Z. Zivkovic et al., [3] proposed pixel-level approach for modeling the background. Each pixel is modeled as mixture of Gaussian probability density functions. Recursive equations are used to constantly update the mean, covariance matrix and also the number of Gaussian components. However, there is no limit in considering the number of past pixels in modeling the present pixel. This makes the algorithm diverge with no solution.

J. McHugh et al, [4] proposed a threshold value which is adaptive which depends on the video statistics. It switches between the nonparametric and mixture of Gaussians models. A foreground model based on small spatial neighborhood to improve discrimination sensitivity is also introduced. However, finding the video statistics and choosing the threshold value by means of the two statistical models is expensive. For simple environments like pedestrian tracking videos, instead of adaptive threshold value, either mixture of Gaussians or any nonparametric model is recommended.

Liao S et al., [5] proposed a novel background subtraction framework by taking the illumination variations and dynamic backgrounds in to consideration. The initial step is, scale independent kernel is developed for handling changes in the lighting conditions. After that, the mask based on the shape observation is developed to efficiently handle the possibility allocation of shapes in the processing of intensity values. The last method is, development of multitude background based techniques for tackling the adaptive critical background models. However, the proposed techniques are for particular illumination variations and particular dynamic backgrounds, which is not the case in real-time scenarios

## III. OBJECTIVES

1. To develop an efficient human tracking algorithm.
2. To reduce number of iterations required for modeling the pixel and thereby reducing the time for tracking.

## IV. DATASETS

For experimental purpose PETS dataset is taken. PETS (Performance Evaluation of Tracking and Surveillance) is a standard dataset which will be released every year. The scenario is of 1 minute 53.43 seconds duration and at a rate of 7 frames per second with a resolution of 576X720 pixels. Fig: 1 a sample frames from PETS dataset. The number of humans in the video are 8. There are isolate humans, partially occluded humans and also fully occluded humans. Both static and dynamic occlusions are present in the video.

Fig. 1: Simple human

## V. HUMAN DETECTION

There are many applications of human detection in the real-time. They are human recognition, tracking of human in surveillance videos and so on. Detection of humans in surveillance videos is a great challenge, because every human is unique in appearance and pose variations. Therefore, there should be a strong technique for extracting features even when there are occlusions, cluttered background, illumination changes and low contrast frames. The video monitoring techniques use color cameras at some point of night when there may be no brightness and additionally when the frames are not clear due to dark luminosity. This makes the modifications in lighting conditions which is an essential point. To consider, many researchers have proposed various techniques for detecting humans in video frames. Many strategies for human characteristics extraction had been proposed with the aid of the researchers as they can observe these functions for classifiers like SVM (Support Vector Machine).

One of the proposed techniques is Histogram of oriented Gradients (HOG) that is considerably utilized in detection of humans. The initial step towards prediction and analysis of the intention and behavior of human is human detection. Although detection of humans is a crucial work for many computer vision applications like gesture recognition, human tracking, video surveillance and action recognition, the time required for the task has always been a large overhead for real time processing. Researchers have proposed many methods for human detection from images and also from video frames. Many of feature extraction algorithms have also been developed which can be applied to classifiers like SVM and others. This is extensively used for detection of human beings. There are two steps for human detection process. The initial step is detecting objects and the next one is classification. The first step of detecting the object might be achieved by way of optical waft, background subtraction and spatio-temporal filtering. Background subtraction is a popular method for detection of moving objects where it identifies the moving objects by way of taking the difference among the present frame and the average frame in a pixel based or block based style. There are lots of techniques to achieve subtracting the background.

The most prominent ones are Non-Parametric Background (NPB), Gaussian Mixture Model, Temporal Differencing (TD), Hierarchical Background (HB) and Warping Background Models (WBM). The optical flow-based human detection technique uses the moving object flow vector characteristics over time to detect the objects which are moving in the image frames. Though these techniques are susceptible to non-uniform lighting, color and image noise, all the optical flow computation techniques have requirement of computations and are vulnerable to motion discontinuities. In spatio-temporal filter methods which are based on motion detection, the motion is categorized by the 3-D spatio-temporal volume data occupied by the moving human in the image sequence. Process of simple implementation and low computational complexity are the two advantages of optical flow based human detection.

There are three categories of object classification. There are three types of template matching’s, based on the movement, color and silhouette/shape. Method based on silhouette gives the knowledge of the structure of the objects in movement as rectangles, clusters and so on. If that is the case, it is identified as the problem of general template matching. But the bodily structures and the variations in the observed points implies a large count of likely appearances of the frame; building it solid to differentiate accurately affecting nonhuman from different moving human beings using this method. The silhouette based method by itself cannot differentiate the persons from affecting things. This inadequacy could be conquered by mixing the silhouette based method with one more thing like texture based methods. Techniques dependent on quality such as HOG use multiple features along with SVM for finding the humans in the videos.

## VI. Human Tracking

Performance of person tracking method in computer vision systems has fascinated many authors from the last few years. The reason for this could be as there are many areas where identification of person plays vital role. There are so many difficulties in identifying a person in surveillance videos. Overlapping is another major reason for difficulty in person identification in the video scenes.

In this method, the steps involved are:

1) Identification of a person in a scenario and finding its position.

2) Finding the person location variations in adjacent frames called as person trajectory.

3) Analyzing the person trajectory information.

Finding the person trajectory is a noted predicament in surveillance videos and allied scenes. If the total frames of the surveillance video are available, the aim is to find the tracking path of every person in the video. As the present world is adversely using the robots, determining the location, finding the trajectory, anticipating the movements of the persons are becoming more important. For accident free movement in the lively environment, the person identification and ability of finding the trajectory is becoming so important.

The person paths/trajectories are required to anticipate the upcoming positions of the humans for the purpose of calculating the proper paths for the robots like, unmanned vehicles, traffic monitoring systems. The task of finding the trajectory of the humans and things has been learned for almost twenty years and still exist like a significant issue in the human scenarios. Finding the trajectory of the person still exist like a major issue. The reason behind it is the persons adapted to the situations and modify their attributes like movement to overcome the accidents.

There is a lot of unpredictability in the visibility of the lighting. Because of this, it becomes difficult for texture dependant trajectory techniques to continuously find the trajectory. The other problem that could arise in finding the trajectories is occlusions. There is always a coincidence of two persons and there by overlapping takes place. This is the major issue in any human trajectory techniques. This has been addressed up to some degree by online person trajectory algorithms; still there is a lot to work on multiple person trajectories.

Finding the trajectory of the person is a vital area in surveillance videos like grocery stores, cinema halls, bus stands etc. the complicated activity is to identify and find the trajectory of the things like persons. The complexity increases because various issues need to be addressed and take into account in finding the trajectory of the person. The complication involves presence of people, clothing, illumination changes and pose changes.

## VII. Weighted Running Window Background Model (Wrw)

It is a modification of the existing GMM algorithm. The history of past pixels in modeling the present pixels is modified [6].

Each pixel is modeled either as a background or the foreground. This is a simple binary classification problem. Here, the weights of the Gaussian functions tell the contribution of the pixel towards the foreground and background. The weights are the probabilities of the Gaussian distribution functions. The pixels belong to either foreground or background. Therefore the total probability of the pixel belonging to foreground or background is one. This is given in the below equation.

$$p(F \mid \bar{x}^i) + p(B \mid \bar{x}^i) = 1 \qquad (6.2)$$

Where F is the foreground class and B is the background class.

### A. Number of Pixels

The number of past pixels considered for modelling the current pixel are limited in this algorithm. Rather than considering the entire history of pixels in that position, here only a set of past pixels are considered for background modeling [10 – 12].

Following are the list of reasons for considering the limited number of pixels for background modelling:

GMM is known to diverge and find solution with infinite likelihood if the entire history of pixels is considered.

The object crosses a pixel completely within 25 to 50 frames in pedestrian tracking. (Considering the speed of movement of the pedestrian to be average and the resolution of the camera to be average).

Limited number of pixels means limited number of data points for each distribution. The number of iterations taken by the EM algorithm also reduces therefore finds quick solution.

### B. Weight Values

In the Gaussian mixture model algorithm, the number of pixels considered in modelling the present pixel are from the beginning of the first frame. All these past pixels are given equal weights. For the case of human tracking, the pedestrian usually moves at a speed of around five kilo meters per hour and the size of the video frame is 560X720, the number of past pixels to be considered should not be greater than 50. Therefore limited number of frames are considered for modelling the present pixel in this algorithm. Among these limited set of past pixels, the immediate past pixels are given more weightage as they contribute more in modelling the present pixel. The last pixel in the window of pixels is given the least weight [13 – 15]. The weights for these past pixels are linearly decreasing from the immediate past pixel to the last pixel in the window. Weighing the past pixel by 'N' means considering the pixel 'N' number of times.

### C. Formulation of WRW Based GMM

The image is modeled as a Gaussian mixture distribution in the spatial and color space, i.e. each pixel is represented as

$\bar{x}^i \in [I]$

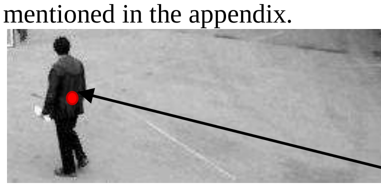
The past pixels considered here in modeling the pixel $\bar{x}^i$ are N.

$\bar{x}^i$ is dependent on $\{\bar{x}_1^i, \bar{x}_2^i \ldots\ldots \bar{x}_N^i\}$

Using Expectation Maximization, the parameters of the Gaussian functions namely mean, variance and the weight of the Gaussian function are calculated. Most of the learning takes place in the initial iterations of the EM algorithm [8 – 9]. The convergence occurs in the initial iterations. After some iteration, it is only refinement of the parameters and the estimates. The changes in the refinement process are almost negligible [16 – 18].

## VIII. Results and Discussions

The pixel can be modeled whether the pixel belongs to foreground or background by using the full history of frames. The pixel modeling can also be achieved by using a set of past frames. This simplifies the modeling process and also reduces the number of iterations taken by the EM algorithm for convergence. To illustrate this, two experiments are conducted. To model the pixel present in the current frame (270th frame), in the first experiment, all the 269 past frames are considered. The pixel values are mentioned in the appendix. In the second experiment, only 55 frames are considered. Here, the assumption is made that the speed of the pedestrian is 3 to 5kmph. The frame rate of the PETS dataset is 7 frames per second. Therefore, to model the pixel, it is sufficient to have past 55 frames. The pixel values are mentioned in the appendix.



The image dimension is 106X260.

Fig. 2: Current frame (270th frame) marked with pixel to be modeled

Modeling 270th frame pixels of PETS 2009 Dataset using full history.

In the current frame, at the position (41, 43), the pixel value is 21.

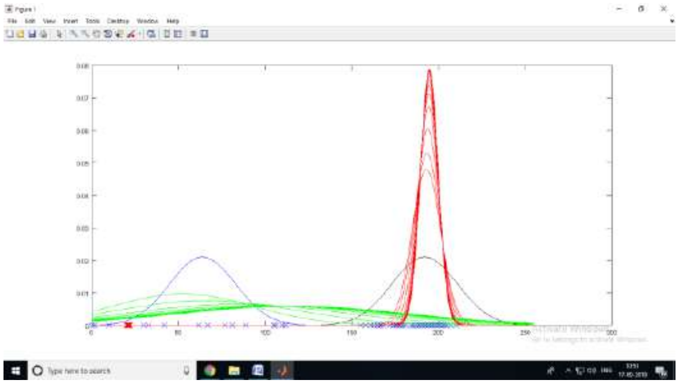


Fig. 3: Different Gaussian functions for modeling the distribution of pixels

The image dimension is 106X260.

Fig. 4: Current frame (270th frame) marked with pixel to be modeled

Fig: 2 and Fig: 4 shows the current frame. The red dot shows the position of the current pixel to be modeled.
In the current frame, at the position (41, 184), the pixel value is184.

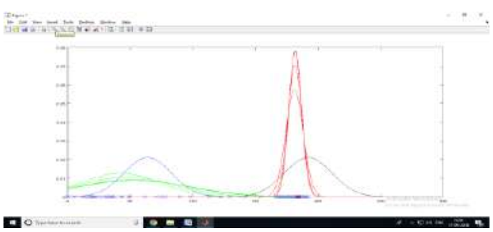


Fig. 5: Different Gaussian functions for modeling the distribution of pixels

Fig: 3 and Fig: 5 shows the modeling of pixel value 21 as foreground by using GMM algorithm. The present pixel is marked as a dark red cross mark and the remaining pixels are marked as blue cross marks.

TABLE 1: MODELING OF PIXEL USING FULL HISTORY OF FRAMES

| Number of Past Pixels Considered for Modeling | Modeling | Number of Iterations Taken |
|---|---|---|
| 269 | Foreground pixel with position (41, 43) | 78 |
| 269 | Background pixel with position (41, 184) | 33 |

The number of iterations taken for modeling the pixel are given in table 1.
In the second experiment, the number of past frames are confined to be 55. A window of length 55 is kept and as the video runs to the next frame, the window of past frames moves. Therefore this window is called running window.
Modeling 270th frame pixels of PETS 2009 Dataset using rectangular window of length 55. The pixel values are mentioned in the appendix.
In the current frame, at the position (41, 43), the pixel value is 21.

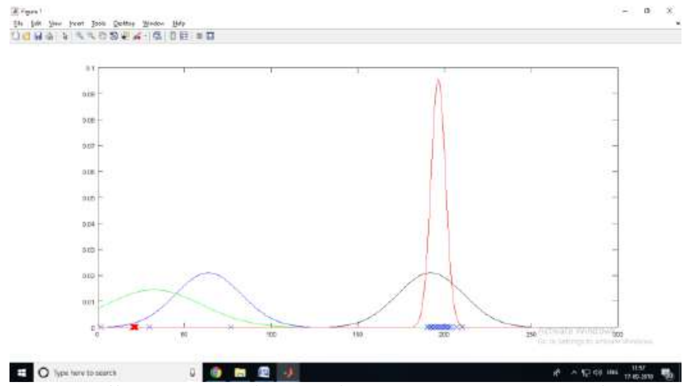
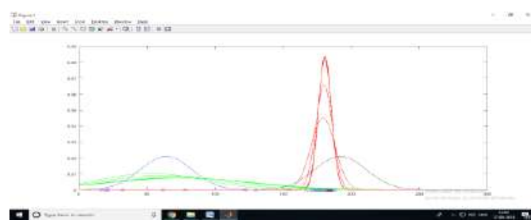


Fig. 6: Different Gaussian functions for modeling the distribution of pixels using 55 past frames

In the current frame, at the position (41, 43), the pixel value is 184.



nctions for

Fig. 7: Different Gaussian fu modeling the distribution of pixels using 55 past frames

Fig: 6 and 7 show that the pixel is modeled using GMM. In Fig: 6, the red pixel is more occupied by the green Gaussian function which is foreground. In Fig: 7, the red pixel is more occupied by the red Gaussian function which is a background. The number of iterations taken for modeling the pixel are given in table 2.

TABLE 2: MODELING OF PIXEL USING A WINDOW OF FRAMES

| Number of Past Pixels Considered for Modeling | Modeling | Number of Iterations Taken |
|---|---|---|
| 55 | Foreground pixel with position (41, 43) | 4 |
| 55 | Background pixel with position (41, 184) | 17 |

The third experiment is also conducted. Instead of considering all the 55 frames, here only 10 past frames are considered. The pixel values are mentioned in the appendix. In modeling any pixel, the dependency of the current pixel is more on the immediate past pixel and more weightage is given to the immediate past pixel and as the window goes away from the current frame, the weightage is getting decreased linearly. The immediate past frame pixel is given a weight of 10 and the last frame pixel given a weight of one. Here, weight 10 means, considering the pixel value ten number of times and so on. There are two major advantages of this:

Though the number of past pixels considered are only ten, still because of weights, the total number of pixels used for modeling are 55. The pixel values are mentioned in the appendix.

The diversity of distribution of the pixels is reduced because of considering the same pixel more number of times.

The EM algorithm takes less number of iterations for convergence as the diversity of the distribution is reduced.

Modeling 270th frame pixels of PETS 2009 Dataset using weighted window of length 10.

Past pixel values at the position (41, 43) from 269th frame to 261st frame with weights shown in the Fig: 8.

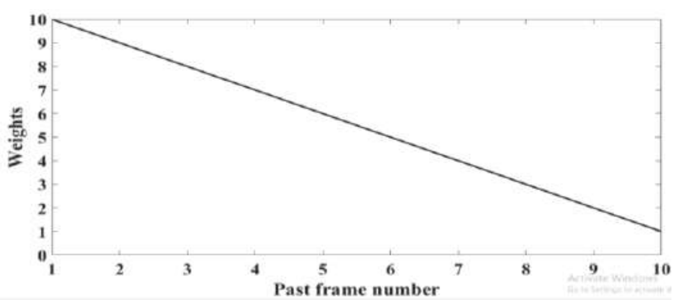



Fig. 8: Window used for modeling the current frame pixel based on the past frame pixels

In the current frame, at the position (41, 43), the pixel value is 21.

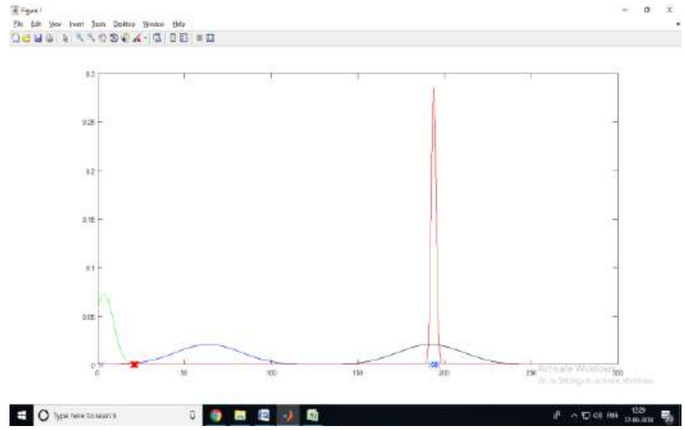


Fig. 9: Different Gaussian functions for modeling the distribution of pixels using 10 past frames

In the current frame, at the position (41, 184), the pixel value is 184.

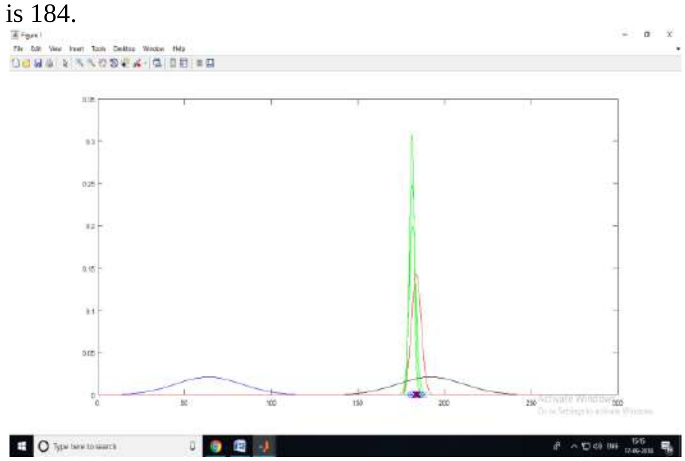


Fig. 10: Different Gaussian functions for modeling the distribution of pixels using 10 past frames

Fig: 9 and 10 show that the pixel is modeled using GMM. In Fig: 9, the red pixel is more occupied by the green Gaussian function which is foreground. In Fig: 10, the red pixel is more occupied by the red Gaussian function which is a background. The number of iterations taken for modeling the pixel are given in table 3.

TABLE 3: MODELING OF PIXEL USING A WINDOW OF FRAMES

| Number of Past Pixels Considered for Modeling | Modeling | Number of Iterations Taken |
|---|---|---|
| 10 | Foreground pixel with position (41, 43) | 2 |
| 10 | Background pixel with position (41, | 3 |

| | | |
|---|---|---|
| | 184) | |

## IX. CONCLUSIONS

Weighted Running Window Back ground model based Gaussian Mixture Model is developed. The algorithm is best suited for tracking the humans. By taking the aspect ratio of the pedestrian into consideration, the region of interest is divided into four regions. The regions are assigned to different parts of the human body. The WRW based GMM algorithm is applied on each and every pixel in the region. By considering only the finite number of pixels from the past frames in deciding whether the present pixel belongs to foreground or background, the algorithm takes less number of iterations to converge. The proposed algorithm is compared with the existing Gaussian Mixture Model algorithm.

## X. LIMITATIONS AND FUTURE SCOPE

The proposed algorithm is best suitable for pedestrian tracking. However, if the speed of the object is greater than 7kmph, it will not produce better results. The above algorithm can be extended further for identifying the suspicious people by incorporating face recognition algorithm.

## XI. ACKNOWLEDGMENTS

The author gratefully acknowledges University Grants Commission SERO for their financial support. This work is a part of Minor Research Project sanctioned in the year 2017 with file no. UGC/SERO/6950. The author gratefully acknowledge Prof. Golpala Krishna Mohan Pillutla and Prof. Madhava Latha Makkena for guiding and mentoring for successfully completion of the project.

## XII. REFERENCES

[1] Elgammal, A., Harwood, D., Davis, L. "Non-parametric model for background subtraction", European Conference of Computer Vision, pp. 751–767 (2000).

[2] Kaew TraKul Pong P, Bowden R (2001), "An improved adaptive background mixture model for realtime tracking with shadow detection", proceedings of 2nd European workshop on advanced video based surveillance systems, AVBS01. Sept 2001. Video based surveillance systems: computer vision and distributed processing, Kluwer Academic Publishers, pp 1–5.

[3] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," Proc. ICPR, 2004.

[4] J. McHugh, J. Konrad, V. Saligrama, and P.-M. Jodoin "Foreground-adaptive background subtraction, " IEEE Signal Process. Lett., vol. 16, no. 5, pp. 390–393, May 2009.

[5] Liao S, Zhao G, Kellokumpu V, Pietikäinen M & Li SZ. (2010) "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes", Proceedings CVPR: 1301–1306.

[6] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., "Performance Analysis of WRW Background model based GMM with other algorithms for tracking humans in surveillance videos" in Journal on Software Engineering, Volume 12, Issue 4, (April – June 2018) pp. 32 – 42.

[7] Gonzalez, R. C., Eddins, S. L., & Woods, R. E. (2004), Digital Image Publishing Using MATLAB. Prentice Hall.

[8] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., "Qualitative Performance Metrics for Object Tracking Algorithms" in Journal on Image Processing, Volume 4, Issue 4, (October - December 2017), pp. 1 – 15.

[9] Comaniciu, D., Ramesh, V., & Meer, P. (2003). Kernel based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5), 564-577.

[10] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., "Human Tracking Under Severe Occlusions" in Journal on Software Engineering, Volume 12, Issue 1, (July - September 2017), pp.29 – 37.

[11] Hager, G. D., Dewan, M., & Stewart, C. V. (2004, June). Multiple kernel tracking with SSD. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.

[12] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., "Color and Shape Based Automatic Detection of Pedestrians in Surveillance Videos" in Journal on Image Processing, Volume 5, Issue 1, (January – March 2018) pp. 1 – 11.

[13] Birchfield, S. T., & Rangarajan, S. (2005, June). Spatiograms versus histograms for region-based tracking. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 2, pp. 1158-1163). IEEE.

[14] Harihara Santosh Dadi, P. G. Krishna Mohan and Madhavi Latha M., "Human Tracking Using Weighted Running Window Background Model Based GMM" in Journal on Software Engineering, Volume 12, Issue 2, (October - December 2017), pp. 44 – 54.

[15] Lee, D. S. (2005). Effective Gaussian mixture learning for video background subtraction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5), 827832.

[16] Paalanen, P ., Kamarainen, J. K., Ilonen, J., & Kälviäinen, H. (2006). Feature representation and discrimination based on Gaussian mixture model probability densities-practices and algorithms. Pattern Recognition, 39(7), 1346-1358.

[17] Wang, Y., Zhang, J., Wu, L., & Zhou, Z. (2010). Mean shift tracking algorithm based on multi-feature space and grey model. Journal of Computational Information Systems, 6(11), 3731-3739.

[18] Benfold, B., & Reid, I. (2011, June). Stable multi-target tracking in real-time surveillance video. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (pp. 3457-3464). IEEE.

## XIII. BIOGRAPHY

Mr. D. Harihara Santosh obtained his B.Tech. and M.Tech Degrees from JNT University, Hyderabad in the year 2005 and 2010. Presently he is pursuing Ph.D, in Video Processing at JNTU, Hyderabad. He has 22 publications in both International and National Journals and presented 20 papers at various International and National Conferences. His areas of interests are Image and Video Processing.

## XIV. APPENDIX

Past pixel values at the position (41, 43) from 269$^{th}$ frame to 1$^{st}$ frame are given by

X=[ 2 192 194 194 195 192 195 196 196 195 190 194 193 192 192 198 190 202 198 196 200 203 201 198 195 198 200 204 195 194 207 210 30 77 192 194 194 200 199 198 195 195 201 194 196 192 198 191 199 197 199 193 195 203 193 199 193 192 194 204 205 199 198 200 200 198 193 199 203 199 203 199 200 203 205 200 197 199 201 200 196 200 199 200 199 195 196 193 193 191 198 195 165 185 196 106 89 105 112 155 188 177 181 180 81 176 188 179 184 0 163 176 166 168 184 188 184 185 191 190 192 192 191 190 191 193 193 195 199 198 198 201 200 201 198 198 199 200 199 200 197 200 200 201 200 197 200 201 197 193 191 189 190 188 180 10 178 191 193 194 197 199 195 193 198 198 200 200 201 199 196 203 199 201 200 198 198 199 198 201 201 197 198 198 200 197 197 194 192 195 188 193 189 190 190 187 191 184 187 179 174 167 161 166 175 67 110 158 62 42 172 192 186 185 190 189 192 192 193 190 188 191 193 191 194 190 192 190 192 192 189 190 191 191 194 192 192 191 191 190 190 195 193 195 191 196 193 189 188 193 33 1 0 193 191 190 191 192 191 193 202 193 198 194 190 193 193 193 202 ];

Past pixel values at the position (41,184) from 269th frame to 1st frame are given by

X=[ 183 186 185 187 184 183 183 180 188 181 174 182 183 185 179 182 177 167 176 78 122 176 173 177 173 170 154 32 21 20 17 62 130 181 172 173 179 181 187 181 180 184 185 183 185 175 182 182 184 185 185 178 181 186 178 182 187 180 180 177 186 181 180 180 181 188 181 184 182 181 180 44 174 173 184 183 186 187 186 186 185 188 179 184 180 166 27 166 168 173 180 183 186 185 184 186 183 187 184

185 184 190 182 189 187 186 189 183 183 184 192 192 190 188 186 181 185 185 178 174 24 12 113 43 182 181 182 190 185 188 183 186 187 186 183 173 171 18 1 78 154 98 178 180 187 180 183 183 191 188 188 187 182 179 180 183 185 186 185 184 186 188 192 192 183 181 182 181 189 184 187 186 184 180 186 186 185 187 183 187 189 183 184 179 182 181 186 184 178 181 177 185 176 181 186 177 183 176 169 174 173 172 75 36 35 50 34 30 22 30 180 187 179 188 176 178 185 184 183 188 177 183 180 179 173 174 173 166 175 24 14 14 142 34 49 38 71 90 173 174 180 185 180 182 179 178 182 184 179 189 183 183 182 177 181 182 182 179 182 180 175 181 179 180 5 0 176 175 174];

Past pixel values at the position (41, 43) from 269th frame to 215th frame are given by

X=[ 2 192 194 194 195 192 195 196 196 195 190 194 193 192 192 198 190 202 198 196 200 203 201 198 195 198 200 204 195 194 207 210 30 77 192 194 194 200 199 198 195 195 201 194 196 192 198 191 199 197 199 193 195 203 193 ]

Past pixel values at the position (41, 184) from 269th frame to 215th frame are given by

X= [183 186 185 187 184 183 183 180 188 181 174 182 183 185 179 182 177 167 176 78 122 176 173 177 173 170 154 32 21 20 17 62 130 181 172 173 179 181 187 181 180 184 185 183 185 175 182 182 184 185 185 178 181 186 178 ];

Past pixel values at the position (41, 43) from 269th frame to 260th frame decreasing weights are given by

X=[ 2 2 2 2 2 2 2 2 2 2 192 192 192 192 192 192 192 192 192 194 194 194 194 194 194 194 194 194 194 194 194 194 194 195 195 195 195 195 195 192 192 192 192 192 195 195 195 195 196 196 196 196 196 195 ]

Past pixel values at the position (41, 184) from 269th frame to 260th frame with decreasing weights are given by

X= [183 183 183 183 183 183 183 183 183 183 186 186 186 186 186 186 186 186 186 185 185 185 185 185 185 185 185 187 187 187 187 187 187 187 184 184 184 184 184 184 183 183 183 183 183 183 183 183 183 180 180 180 188 188 181];