CHAPTER 1

INTRODUCTION

Introduction

Overview

Outline of the Thesis

1.1. INTRODUCTION

Object tracking is an important component of many vision systems. In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In order to assist human operators with identification of important events in videos, an intelligent visual surveillance system can be used. Such a system requires fast and robust methods for moving object detection, and then analysis after tracking. Moving object detection is the basic step for many video analysis tasks. The performance of this step is particularly significant because subsequent processing of the video data is greatly dependent on this step. Moving object detection aims at extracting moving objects that are of interest in video sequences. The problems with dynamic environmental conditions make moving object detection very challenging. Some examples of these problems are shadows, sudden or gradual illumination changes, rippling water, and repetitive motion from scene clutter such as waving tree leaves. Commonly used techniques for moving object detection are background subtraction, temporal frame differencing, and optical flow. The next step in the video analysis is object tracking. This problem can be formulated as a hidden state estimation problem given available observations. Another way to look at object tracking is the creation of temporal correspondence among detected object from frame to frame. It is used not only for visual surveillance, but also for augmented reality, traffic control, medical imaging, gesture recognition, and video editing. In the area of moving object detection a technique robust to background dynamics using background subtraction with adaptive pixel-wise background model update is described. Once the object is detected the next step involves the tracking of the detected object. Tracking is usually performed in the context of higher-level applications that require the location and/or shape of the object in every frame. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. There are a

variety of approaches, depending on the type of object, the degrees of freedom of the object and the camera, and the target application. In our work, this has been done by calculating the detection probabilities of the moving object for the next frame. To perform tracking in video sequences, an algorithm analyses sequential video frames and outputs the movement of target between the frames. Thus we perform tracking of an object in a video sequence, that is, continuously identifying its location.

1.2. OVERVIEW

Here we discuss about various tracking methods. The main tracking categories [10] are followed by a detailed section on each category.

- Point Tracking:

Objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. This approach requires an external mechanism to detect the objects in every frame. Point correspondence is a complicated problem-specially in the presence of occlusions, misdetections, entries, and exits of objects. Point Tracking can be defined as the correspondence of detected objects represented by points across the frames. Point Tracking is a difficult problem particularly in the existence of occlusions, false detections of object. Recognition of points can be done simply by thresholding, at of identification of these points. Point Tracking is capable of dealing with tracking very small objects only. Overall, point correspondence methods can be divided into two broad categories, namely, deterministic and statistical methods. The deterministic methods use qualitative motion heuristics to constrain the correspondence problem. On the other hand, probabilistic methods explicitly take the object measurement and take uncertainties into account to establish correspondence.
Some approaches based on Point Tracking are described below:

a. Kalman Filter:

Kalman filter are based on Optimal Recursive Data Processing Algorithm. Here Gaussian state distribution is assumed. Kalman filtering [13] is composed of two stages, prediction and correction. Prediction of the next state using the current set of observations and update the current set of predicted measurements:

$$X^t = DX^{t-1} + W, \quad (1)$$

$$C^t = DC^{t-1}D^T + Q^t \quad (2)$$

Where, X^t and C^t are the state and covariance predictions at time t. D is the state transition matrix which defines the relation between the state variables at time t and time t-1. Q is the covariance of noise W.

The second step is gradually update the predicted values and gives a much better approximation of the next state. It uses the current observation Z^t to update the object state:

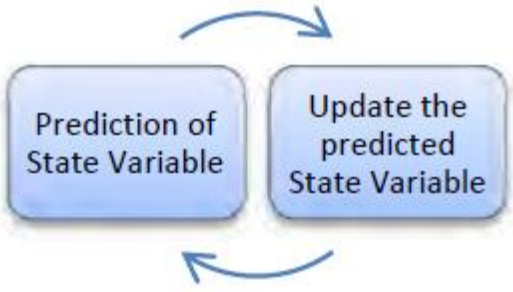
$$K^t = C^t M^t [MC^t M^T +]^{-1}, \quad (3)$$

$$X^t = X^t + K^t [Z^t - MX^t], \quad (4)$$

$$C^t = C^t - K^t MC^t \quad (5)$$

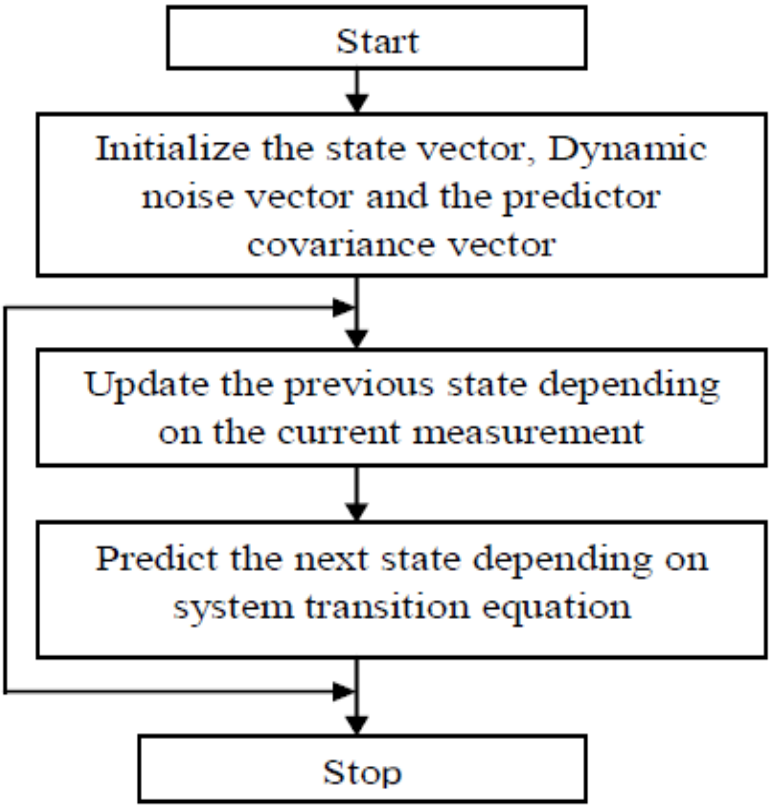
Where, M is the measurement matrix and K is the Kalman gain. The Kalman filter has been extensively used by the vision community for tracking.

It can be shown as:



Figure(1.1): Basic Steps in Kalman Filter

Kalman filter tries to find a balance between predicted values and noisy measurements. The value of the weights is decided by modelling the state equations. Kalman filter track the system in discrete interval of time. The flowchart for the algorithm is drawn below:

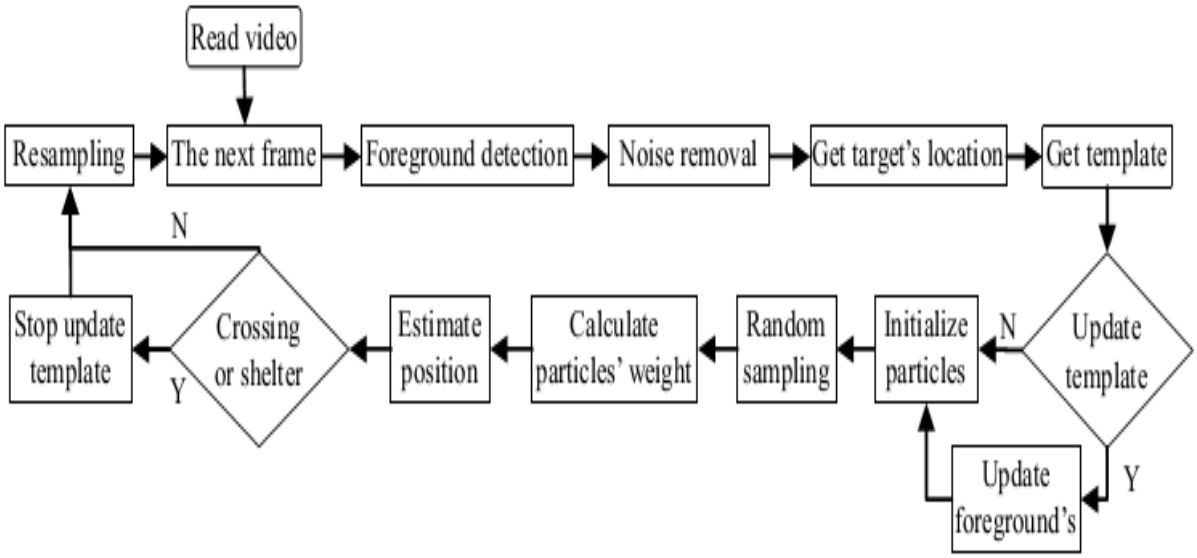


Figure(1.2): Algorithm for Kalman Filter

Kalman filtering [16] approach is capable in dealing with noise. It is applicable only for single object, multiple objects. Kalman Filter always gives optimal solution. It is used in vision community tracking.

a. Particle Filter:

Particle filter is used to track non-linear, non- Gaussian moving objects. Particle filter is used to detect moving objects in difficult scenes. The algorithm [15] uses codebook background model for detection of objects, then color histogram of every objects is obtained and particle sampling range is limited by the combination of foreground detection information, which results particle filter reflect the objects more exactly and timely. An algorithm of codebook background model is used for object detection and Particle filter algorithm is used for object tracking.



Figure(1.3): Algorithm for Particle Filter

Simulation is performed on the video sequences of size 320×240 pixels, frame rate 25fps. This algorithm helps in solving the problem of particle degradation which was arising in the case of traditional particle filter. Here author has mainly emphasized on information loss in the detection and tracking. This method makes the particle filter reflect the objects more accurately and timely. Here average processing time per frame

is calculated on the video sequences of 320×240 pixels, frame rate is 25fps and the result was 94ms.

b. Multiple Hypothesis Tracking (MHT):

The MHT algorithm is based on motion correspondence of several frames together. Better results are obtained if correspondence is established observing several frames rather than using only two frames. The MHT algorithm [34] upholds several suggestions for each object at each time. The final track of object is the most likely set of correspondences over time period of its observation. MHT is an iterative algorithm. Iteration begins with a set of existing track hypotheses. Each hypotheses is a crew of disconnect tracks. For each hypothesis, a prediction of object's motion in the succeeding frame is made. The predictions are then compared by calculating a distance measure.
MHT focuses on FOV (field of view) of object entering and object leaving. It can handle occlusion. It can track multiple objects.

- Kernel Tracking:

Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames. This motion is usually in the form of a parametric transformation such as translation, rotation, and affine. Kernel tracking is typically performed by computing the motion of the object, which is represented by a primitive object region, from one frame to the next. The object motion is generally in the form of parametric motion (translation, conformal, affine, etc.) or the dense flow field computed in subsequent frames. These algorithms differ in terms of the appearance representation used, the number of objects tracked, and the method used to estimate the object motion. We divide these tracking methods

into two subcategories based on the appearance representation used, namely, templates and density-based appearance models, and multi view appearance models.

Kernel tracking is usually performed by computing the moving object, which is represented by a potential object region, from one frame to another. The object motion is usually in the form of parametric motion such as translation, conformal, affine, etc. These algorithms appear different in terms of the representation used, the number of objects tracked, and the method used for approximating the object motion. There are several techniques based on representation of object, object features, appearance and shape of the object. Few of the tracking technique based on

Kernel tracking approach:

a. Dual-Tree Complex Wavelet Transform Technique:

Real Wavelet Transform suffers from shift variance and poor directionality. Object tracking method based on complex wavelet transform is used. Real Filter is used to obtain shift invariance. Two steps are followed; Segmentation and Tracking. While Segmentation Process is optical flow computation for finding moving object is used. Segmentation algorithm proceeds as follows: Take first and tenth frame of video sequence. Then, Convert them to grey level image; further determine optical flow using Horn Schunck method between these two images. Thus we find the magnitude square value of the optical flow $|V|^2$. Afterwards, find the mean value of the $|V|^2$ for the first image and compare its value with magnitude square value of the optical flow at each pixel location in the image. If $|V|^2$ at any pixel is greater than or equal to the mean value. Then keep its pixel value 1, otherwise assign it 0. On the other hand, in the Tracking Process, in different video frames centroid of the moving object is calculated.

Dual –Tree CxWT is an efficient way of implementing an analytic wavelet transform. Object is tracked in next frames by computing the energy of dual-tree complex wavelet coefficients [33] corresponding to the object area and matching this energy to that of in the neighborhood area. It has properties like directionality selectivity, Shift

invariance and perfect reconstruction. Experiment is performed on video sequence (240×320) and minimum energy difference & corresponding boundary values are calculated. Then centroid corresponding to this boundary value is calculated.

b. Histogram-based

In this Histogram-based target representation is improved by spatially masking (spatially smoothness achieved in similarity function) with an isotropic kernel. The traditional mean shift process is limited by the fixed kernel bandwidth. It was overcome by CAMshift. It coped well with camera motion, partial occlusions, clutter and target scale variations. But sophisticated motion filter required , if occlusions are present.

- Silhouette Tracking:

Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution. Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames. Objects may have complex shapes, for example, hands, head, and shoulders that cannot be well described by simple geometric shapes. Silhouette based methods provide an accurate shape description for these objects. The goal of a silhouette-based object tracker is to find the object region in each frame by means of an object model generated using the previous frames. This model can be in the form of a color histogram, object edges or the object contour. We divide silhouette trackers into two categories, namely, shape matching and contour tracking. Shape

matching approaches search for the object silhouette in the current frame. Contour tracking approaches, on the other hand, evolve an initial contour to its new position in the current frame by either using the state space models or direct minimization of some energy functional.

Objects having composite shapes for example, hands, head, and shoulders, are cannot be well defined by geometric shapes. Silhoue0tte based approach will give perfect description of shape of those objects. The aim of the silhouette based tracking is to find the object region by means of an object model. This model verifies the object region in each frame. Model can be represented in the form of color histogram, object edges or contour.

We classify silhouette tracking into two categories, namely, shape matching and contour tracking.

a. Contour Tracking:

Contour tracking methods develop an original contour in the foregoing frame to its new position in the present frame, overlapping of object between the current and next frame. Contour tracking is in the form of state space models. State of the object is named by the parameters of shape and the motion of the contour. The state is updated for each time according to the maximum of probability. Author has used two types of object representation one is implicitly modeled and the other one is explicitly modeled. Performance of the technique based on contour evolution by direct minimization has been analyzed. Here region statistics is calculated using grid points. Occlusion is fully handled.

b. Shape Matching:

This approach checks for object model in the existing frame. Shape matching performance is similar to template based tracking in kernel approach. Another approach to Shape matching is to find matching silhouettes in two successive frames.

Detection based on Silhouette is carried out by background subtraction. Models object are in the form of density functions, silhouette boundary, object edges. Edge based templates has been used by the author. Here temporal spatial velocity in 3D image per frame is calculated. It can track only single object. Occlusion handling is performed using Hough Technique.

1.3. OUTLINE OF THE THESIS

Object tracking is the process of repeated estimation of the state of an object in the next frame, given states in previous frames. Object identification is one of the initial, but paramount steps in object tracking. It is basically, determination of video statistics, object classification, determination of inconsistencies, and then finally human identification. Every object has its unique features in a video scene. These unique features help us to determine whether the object is same in the next frame of a video as we need to track or not. In our project we have first extracted those features using communication theory, or radar theory to be precise. This enriches us with the crucial estimated features of the objects. After this parametric estimation, we have calculated the optimum detection probabilities for the target object using Neyman Pearson Theory. Considering the decision probabilities, the miss probability, the false probability, the detection probability and the minimized cost function, determined using Neyman Pearson Theory, we identify the target object. In the next section we would see our proposed work.

CHAPTER 2

THE PROPOSED METHOD

Introduction to Our Technique

Object Detection

Object Tracking

Methodology

2.1. INTRODUCTION

This research involves a robust method for the identification of an object to be the same in the next frame as in the previous frame with the help of Neyman Pearson decision theory applied on the estimated values of the object features (e.g. position, displacement, velocity etc). We take it as a fact that every human, has its unique features, which are not correlated with any other human being. There may be an identical size, or perhaps they are moving with an identical velocity, but all in all, the possibility of all the features of each person being identical, simultaneously is negligible. Thus, we extract a number of features of each person in the video to make that person unique for the computer system. Subsequently, the feature values are estimated for the next frames. These estimated values for each feature of the person are nothing but the maximum likelihood estimates of the feature. Later, using the already determined estimated values of the features the detection probabilities of each person are determined for the next frame. It also gives us the risk involved in considering the concerned object in the current frame to be detected in the next frame. This statistical modeling is state of art in this field and can work on multiple objects in the video frames simultaneously. This is a very straightforward approach and the results from the Neyman Pearson Criterion recognize the object in an easy way. As it forms a simple hypothesis testing problem with only two conclusions, either the target object is present in next frame, or not.

2.2. OBJECT DETECTION

Moving object detection is the basic step for further analysis of video. Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. It handles segmentation of moving objects from stationary background objects. This focuses on higher level processing. It also decreases computation time. Due to environmental conditions like illumination changes, shadow object segmentation becomes difficult and significant problem. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights regions that changes dynamically in consecutive frames. Given the object regions in the image, it is then the tracker's task to perform object correspondence from one frame to the next to generate the tracks.

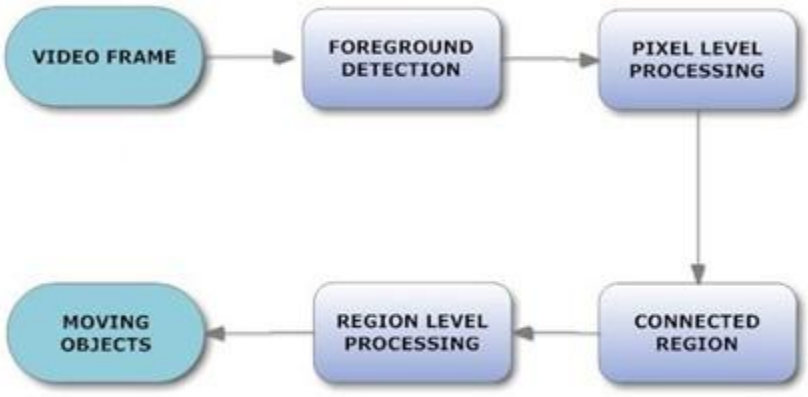


Figure (2.1): Object Detection flow chart

2.2.1 Video Frames:

The first step in the Object Detection is to fetch the video frames from which the moving objects have to be detected and then eventually tracked. Then the modeling of background is done. If the first frame is not free from foreground objects then, the background become complex to extract. Nonetheless, the initial frames help in modeling of the background frame. For this we first do the background scene initialization. There are various techniques used to model the background scene. The background scene related parts of the system is isolated and its coupling with other modules is kept minimum to let the whole detection system to work flexibly with any one of the background models. Next step in the detection method is detecting the foreground pixels by using the background model and the current image from video.

2.2.2 Foreground Detection:

The next step deals with distinguishing the foreground objects from stationary background. To achieve this, we can use a combination of various techniques along with low-level image post-processing methods to create a foreground pixel map at every frame. We then group the connected regions in the foreground map to extract individual object features such as bounding box, area, perimeter etc. The main purpose of foreground detection is to distinguishing foreground objects from the stationary background. Almost, each of the video surveillance systems uses the first step is detecting foreground objects. This creates a focus of attention for higher processing levels such as tracking, classification and behavior understanding and reduces computation time considerably since only pixels belonging to foreground objects need to be dealt with.

This pixel-level detection process is dependent on the background model in use and it is used to update the background model to adapt to dynamic scene changes. Also, due to camera noise or environmental effects the detected foreground pixel map contains noise.

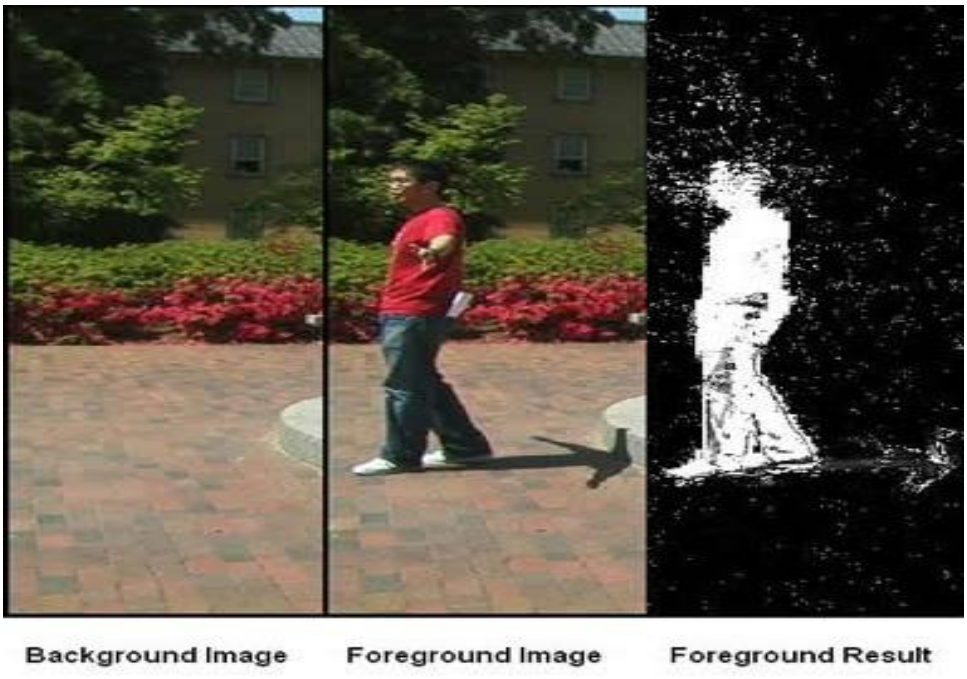


Figure (2.2): Foreground Detection

Foreground detection is simply the background subtraction [2]. Background subtraction is the process of separating out foreground objects from the background in a sequence of video frames. Many methods exist for background subtraction, each with different strengths and weaknesses in terms of performance and computational requirements.

Since background subtraction is being implemented on a wide range and thus within a wide range of computational budgets there are methods of varying complexity:

1. Low-complexity, using the frame difference method,
2. Medium complexity, using the approximate median method, and
3. High-complexity, using the Mixture of Gaussians method.

Frame Difference:

Frame difference is arguably the simplest form of background subtraction. The current frame is simply subtracted from the previous frame, and if the difference in pixel values for a given pixel is greater than a threshold T_s, the pixel is considered part of

the foreground [18]. However, a major flaw of this method is that for objects with uniformly distributed intensity values, the interior pixels are interpreted as part of the background. Another problem is that objects must be continuously moving. If an object stays still for more than a frame period, it becomes part of the background. This method does have two major advantages. One obvious advantage is the modest computational load. Another is that the background model is highly adaptive. Since the background is based solely on the previous frame, it can adapt to changes in the background faster than any other method. Moreover, the frame difference method subtracts out extraneous background noise, much better than the more complex approximate median and mixture of Gaussians methods. A challenge with this method is determining the threshold value. The threshold is typically found empirically, which can be tricky.

Approximate median:

In median filtering, the previous N frames of video are buffered, and the background is calculated as the median of buffered frames. Then, the background is subtracted from the current frame and thresholded to determine the foreground pixels. Median filtering has been shown to be very robust and to have performance comparable to higher complexity methods. However, storing and processing many frames of video requires an often prohibitively large amount of memory. This can be alleviated somewhat by storing and processing frames at a rate lower than the frame rate, thereby lowering storage and computation requirements at the expense of a slower adapting background.

The approximate median method [21] works as such: if a pixel in the current frame has a value larger than the corresponding background pixel, the background pixel is incremented by 1. Likewise, if the current pixel is less than the background pixel, the background is decremented by one. In this way, the background eventually converges to an estimate where half the input pixels are greater than the background, and half are less than the background, approximately the median the approximate median method does a much better job at separating the entire object from the background. This is

because the more slowly adapting background incorporates a longer history of the visual scene, achieving about the same result as if we had buffered and processed N frames. This method is a very good compromise. It offers performance near what you can achieve with higher-complexity methods (according to my research and the academic literature), and it costs not much more in computation and storage than frame differencing.

Mixture of Gaussians:

Among the high-complexity methods, two methods dominate the literature, namely, Kalman filtering and Mixture of Gaussians (MoG) [21]. Both have their advantages, but Kalman filtering gets slammed for leaving object trails that can't be eliminated. Also, MoG is more robust, as it can handle multi-modal distributions. For instance, a leaf waving against a blue sky has two modes, leaf and sky. MoG can filter out both. Kalman filters effectively track a single Gaussian, and are therefore unimodal, they can filter out only leaf or sky, but usually not both.

In MoG, the background isn't a frame of values. Rather, the background model is parametric. Each pixel location is represented by a number of Gaussian functions that sum together to form a probability distribution function F. To determine if a pixel is part of the background, we compare it to the Gaussian components tracking it. If the pixel value is within a scaling factor of a background component's standard deviation σ, it is considered part of the background. Otherwise, it's foreground. MoG is very good at separating out objects and suppressing background noise such as waving trees. However, there are several points where the method breaks down, allowing most of the background to seep into the foreground. These points correspond to relatively rapid changes in illumination. If we go back to the approximate median output we can see similar hiccups, although less pronounced. This is because the background model isn't adapting quickly enough. This is not to say the MoG is less robust, necessarily. But the problem MoG had with illumination changes in the test video does point to one of its main challenges; parameter optimization. The MoG method has five parameters which must be tweaked (the background component weight threshold T_s,

the standard deviation scaling factor D, the learning rate ρ, the total number of Gaussian components, and the maximum number of components M in the background model)

Hence the critical analysis would be that, the simplest method that is frame differencing is arguably the most robust. While it has major flaws, and is probably not suitable for most applications, frame differencing does the best job of subtracting out extraneous background noise such as waving trees. The second most robust method, approximate median, gives us significantly increased accuracy for not much more computation. It had a little trouble with quickly changing light levels, but handled them better than mixture of Gaussians. And Mixture of Gaussians, the most complex of the methods, gives us good performance, but presents a tricky parameter optimization problem.

2.2.3 Pixel-level post-processing operations:

They are performed to remove noise in the foreground pixels [25]. Once we get the filtered foreground pixels, in the next step, connected regions are found by using a connected component labeling algorithm and objects' bounding rectangles are calculated. The labeled regions may contain near but disjoint regions due to defects in foreground segmentation process. Hence, some relatively small regions caused by environmental noise are eliminated in the region-level post-processing step. In the final step of the detection process, a number of object features like area, bounding box, perimeter of the regions corresponding to objects are extracted from current image by using the foreground pixel map. In Pixel Level Post-Processing, the output of foreground detection contains noise. Generally, it affects by various noise factors. To overcome this dilemma of noise, it requires further pixel level processing. There are various factors that cause the noise in foreground detection such as:

Camera Noise:

Camera noise presents due to camera's image acquisition components. This is the noise caused by the camera's image acquisition components. This noise is produce

because of the intensity of a pixel that corresponds to an edge between two different colored objects in the scene may be set to one of the object's color in one frame and to other's color in the next frame.

Background Colored Object Noise:

The color of the object may have the same color as the reference background. Then it is difficult to detect foreground pixels with the help of reference background.

Reflectance Noise:

Reflectance noise is caused by light source. When a light source moves from one position to another, some parts in the background scene reflect light.

2.2.3.1 Gaussian Filter Smoothing:

The noise as explained above in the frames must be removed, otherwise, the connected regions will get distorted, and the object detection would not be accurate. Low pass filters are used for blurring and for noise reduction. Blurring is used in pre-processing tasks, such as removal of small details from an image prior to large object extraction, and bridging of small gapes in lines or curves.

Figure (2.3): A MATLAB example Gaussian filtered image

Gaussian filter [11] is used extensively in image processing for smoothening of the images, and also it can be computed using a simple mask. Hence, Gaussian smoothing is used as a sub operation and this can be performed using standard convolution method. The mask through which convolution of image is to be done is typically smaller than the actual image. Consequently, operation on pixels at a time is done when the mask is swept over the image. The sensitivity of the detector for noise depends upon the size of the Gaussian window, larger the Gaussian mask, lower is the sensitivity of detector towards noise. While with the increase in size of the Gaussian mask, the localization error also increases. An example of a 5*5 Gaussian filter is given below:

1/273	1	4	7	4	1
	4	16	26	16	4
	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Table (2.1): A 5*5 Gaussian Filter example

Gaussian low pass filter is use for pixel level post processing. A Gaussian filters smoothes an image by calculating weighted averages in a filter co-efficient. Gaussian filter modifies the input signal by convolution with a Gaussian function.

2.2.3.2 Morphological Filtering Techniques:

The background can have some external noise which has an intensity of '1' in a binary image and the foreground moving object can have internal noise which has an intensity of '0' in a binary image. We need to remove these errors as they create problem in proper object detection. A morphological filtering approach has been applied using sequence of dilation and erosion to obtain a smooth, closed, and complete contour of a gesture. Morphology deals with the shape or structure of an

object. Morphological techniques probe an image with a small shape or template called a structuring element. Morphology [11][23] is a tool for extracting image components that are useful for representation and description of region shape, boundary, skeleton, convex hull etc. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Morphological operation returns an image in which the pixel has a non-zero value only if the test is successful at that location in the input image. In the morphological dilation and erosion, the structuring element is moved over the actual image and the morphological computations are performed.

- Dilation:

The dilation of an image by a structuring element has the following three effects on the image: Filling of gaps, Removal of noise, Expansion of the object boundary

- Erosion:

The erosion of an image by a structuring element has the following three effects on the image: Removal of external noise, Boundary pixels get eliminated

- Opening:

The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B.

$$A \circ B = (A \otimes B) \oplus B \qquad (6)$$

In the opening operation the external noise is eliminated. The opening of A by B is simply the erosion of A by B followed by dilation of the result by B.

- Closing:

The closing of set A by structuring element B is

$$A \bullet B = (A \oplus B) \otimes B \qquad (7)$$

In the opening operation the internal noise is eliminated. Closing also tends to smooth section of contours but, it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour.

We can use low pass filter and morphological operations, erosion and dilation, to the foreground pixel map to remove noise that is caused by the items listed above. Our aim in applying these operations is removing noisy foreground pixels that do not correspond to actual foreground regions and to remove the noisy background pixels near and inside object regions that are actually foreground pixels.

2.2.4 Detecting Connected Regions:

After detecting foreground regions and applying post-processing operations to remove noisy regions, the filtered foreground pixels are grouped into connected regions. After finding individual regions that correspond to objects, the bounding boxes of these regions are calculated.

2.2.5 Region Level Post-Processing: As pixel-level noise removed, still some artificial small regions remain just because of the bad segmentation. To remove this type of regions, regions that have smaller sizes than a pre-defined threshold are deleted from the foreground pixel map. Once segmenting regions [25] we can extract features of the corresponding objects from the current image. These features are size, center-of-mass or just centroid and Bounded Area of the connected component. These features are used for object tracking and classification for the further processing in event detection.

Thus, Object detection [28] can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to the objects. This process is referred to as the background subtraction.

At the start of the system reference background is initialized with first few frames of video frame and that are updated to adapt dynamic changes in the scene. At each new

frame foreground pixels are detected by subtracting intensity values from background and filtering absolute value of differences with dynamic threshold per pixel. The threshold and reference background are updated using foreground pixel information. It attempts to detect moving regions by subtracting the current image pixel-by-pixel from a reference background image [27] that is created by averaging images over time in an initialized period. The pixels where the difference is above a threshold are classified as foreground. After creating foreground pixel map, some morphological post processing operations such as erosion, dilation and closing are performed to reduce the effects of noise and enhance the detected regions. Pixel is marked as foreground if the inequality is satisfied [4],

$$| It (x , y) - Bt (x , y) | > T \qquad (8)$$

Where, T is a pre-defined threshold,

It is the current frame,

and Bt is background image.

2.3. OBJECT TRACKING

Object tracking is a technique or method used to track the number and direction of objects traversing a certain passage or entrance per unit time. To achieve the goal of intelligent motion perception, much effort has been spent on visual object tracking, which is one of the most important and challenging research topics in computer vision. The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. Object tracker may also provide the complete region in the image that is occupied by the object at every time instant. The tasks of detecting the object and establishing correspondence between the object instances across frames can either be performed separately or jointly. The resolution of the measurement is entirely dependent on the sophistication of the technology employed. There are three key steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of tracks to recognize their behavior.

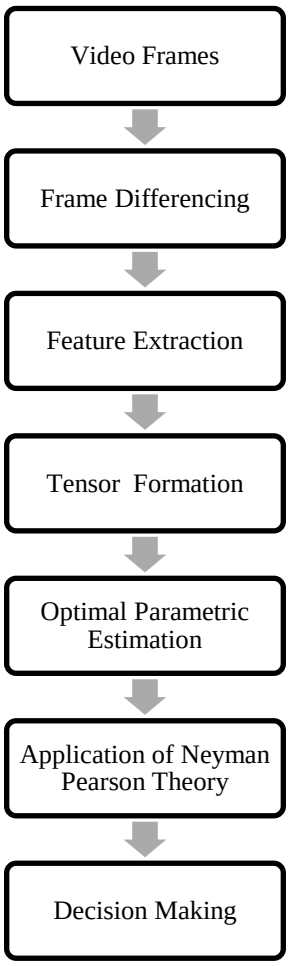
The object tracking is pertinent in the tasks of:

1. Motion-based recognition, that is, human identification based on gait, automatic object detection, etc.
2. Automated surveillance that is, monitoring a scene to detect suspicious activities or unlikely events
3. Video indexing, that is, automatic annotation and retrieval of the videos in multimedia databases
4. Human-computer interaction, that is, gesture recognition, eye gaze tracking for data input to computers, etc.
5. Traffic monitoring, that is, real-time gathering of traffic statistics to direct traffic flow, Vehicle navigation that is, video-based path planning and obstacle avoidance capabilities.

2.4. METHODOLOGY

The complete methodology has been shown in the form a flowchart, starting from the drawing out of the sequence of frames from a video scene, then background subtraction for object detection, followed by the extraction of features and tensor formation which is basically using of graph theory. Next the Neyman Pearson Criterion is applied on the parametric estimated feature values for the detection and identification of the target object.

Following steps have been followed:



Figure(2.4): Flowchart of the complete approach

2.4.1 Video Frames

Initially, the frames from the video sequence are fetched, here, we have used the view 7 of the PETS 2009 dataset of video frames. Then, for the application of the proposed approach, foreground objects are separated out from the background in a sequence of video frames. There are several methods for background subtraction, and they all have their strengths and weaknesses in terms of performance and computational requirements.

2.4.2 Frame Differencing

Frame Differencing is one of the simplest methods for background subtraction. In this, the current frame is subtracted from the previous frame, and if, for a given pixel the difference in pixel values is greater than a threshold, the pixel is considered to be a part of the foreground. Let for current frame F at time t and at pixel position x, the intensity is given by $F_t(x)$. Then for the previous frame, the intensity at the same pixel position is given by $F_{t-1}(x)$. Then, for a defined threshold T, the moving foreground object can be prominently taken out if,

$$| F_t(x) - F_{t-1}(x)| > T \qquad (9)$$

This method is highly adaptive because background depends only on the previous frame and most importantly it has the least complexity. Furthermore, it is also capable of subtracting background noise much better than the others. Now we have, sequence of frames with only foreground objects.

Naturally the foreground objects [7] will have some internal as well as external noise still left, which must be removed. So, image morphological operations, erosion and dilation is applied on the sequence of frames, to get noise free and uniform foreground

objects. This is a necessary step, so as to have the same number of connected regions as the number of objects in the video.

Figure(2.5): Objects represented by centroids in the frames

Afterwards, all the foreground objects are represented by their respective centre of gravity, which is the same as the centroid as shown in figure (2.5). Now as we have all the foreground objects we need to track, we will determine the features of each of the tracking entity.

2.4.3 Feature Extraction

The next step involves the extracting of features of each foreground object per frame from the multi view appearance using Graph Theoretic approach. When the person moves there exists three tracking operators, namely, Translation, Scaling and Rotation. Translation is concerned with the lateral movement. Scaling involves the range of the object. And the rotation is about the direction or changing angle of the person with movement. We have found out only two out of three operators in our case. We have left out the rotation operator which has more complexity involved. The other features we have calculated are height, width, area, coordinates displacement and velocity. The coordinates of each foreground objects are straight away given by the centroids, which form the basis of calculations of the other parameters. The displacement and velocity for the moving objects are then calculated by the simple speed-time-distance relation. The height, width and area are determined using the bounding box enclosing the moving foreground objects. The accuracy of this calculation is increased by determination of size within the bounding box. Thus we are with all the feature values calculated using graph theory.

2.4.4 Tensor Formation

A tensor term has been given to the cascaded matrix formed by bringing together of all the extracted features for each person. This is simply a way of useful representation of the acquired data.

Figure(2.6): Formation of Markov Chain from the Tensors of each frame

Subsequently, a markov chain of the tensors is made for each foreground object, which simply connects the tensors of previous frames with respect to the features. Thus we have a compilation of the extracted features for each foreground object with successive frames.

2.4.5 Optimal Parametric Estimation

Now the parametric estimation is done using radar theory to find out the estimation of the features for the next frame. In most situations, however, the true distributions are unknown and must be estimated from data. In the non-parametric density estimation [17], we assume no knowledge about the density. But for the parametric estimation [19], we assume a particular form for the density (e.g. Gaussian), so only the parameters need to be estimated. The Maximum Likelihood [12] solution for the parametric estimation seeks the solution that best explains the dataset. In the detection problem in radar systems, a signal is transmitted towards a target object from the radar which gets reflected back and this reflected signal is received by the same radar, which then estimates the mathematical parameters of the object like velocity, shape, range. This idea has been used here for the optimal parametric estimation of the features of the objects in the frames. Our eyes do the same job for us as the radar does. The binocular vision of human eye is a natural process, the light rays gets reflected from the objects, which are then received by the human eye, and this is how we are able to see and then the brain does its function to recognize the objects. This process has been utilized in our methodology, to calculate the features of the objects in the frames, and the recognition or the tracking part is done using the Neyman Pearson Decision Theory [12]. The received signals are always distorted due to the non ideal characteristics of the channel and the random noise added to the signal at the receiver input. So it becomes necessary to measure the various parameters of the received signals. Thus the optimal parametric estimation uses the observed values to approximate the unknown feature values. Consequently, the best estimate must be

found, as both the transmitted parameters and received signals may vary randomly. Therefore, based on the theory of parameter estimation, the optimum estimation of size, velocity and other concerned features for the next frame has been determined.
Firstly, we have estimated the size of the concerned object to determine its detection probability later. Considering a simple radar problem, we transmit a signal and then we receive the reflected signal with some noise, delay and different related energy.

First let us get familiar with the Maximum Likelihood Estimation Theory which we have used for the Optimal Parametric Estimation. It searches for the parameter values that have produced the observed distribution most likely. There by, we simply construct a model by its pdf through the given dataset, where θ form the parameter of that pdf. Suppose there is a random variable Y with a known pdf, in our case we have taken it to be Gaussian, and unknown parameter θ, then independent samples y_1, y_2, y_3 .... y_n of the random variable Y forms a probability distribution function $P(y_i|\theta)$, then the joint distribution function is given by,

$$P(y_i|\theta) = P(y_1|\theta)\ldots.P(y_n|\theta) \qquad (10)$$

Thus it assumes a particular model with unknown parameters [19] and so the probability of observing a given event is conditional on a specific set of parameters. Now the likelihood function [8] is nothing but the density function of θ.

$$L(\theta|y_1,y_2\ldots y_n) = \prod_{i=1}^{n} P(yi|\theta) \qquad (11)$$

Now the observed results bring us the most likely parameter θ_e, which is the parameter value that maximizes the likelihood function. It is given by,

$$\theta_e = \underset{\theta}{argmax}\, L(\theta|Y) \qquad (12)$$

Thus θ_e maximizes the likelihood function, which implies that,

$$L(\theta_e) \geq L(\theta), \forall\ \theta \qquad (13)$$

Therefore the best estimate could be determined. This maximum likelihood method [7] [20] of optimal parametric estimation is simple, useful and efficient. This in turns gives the Likelihood Ratio Test [9], which essentially forms basis for the decision rule in Neyman Pearson Test. For instance, if there are two parametric values θ_0 and θ_1, and the Likelihood functions for them are given by $L(\theta_0|y)$ and $L(\theta_1|y)$ respectively. Now if, $L(\theta_1|y) > L(\theta_0|y)$ then parameter θ_1 would be chosen as the maximum likelihood estimated, rather than θ_0 because the estimated value having proximity to θ_1 has the higher probability.

1. *Estimation of Size*

Using the explained theory we would like to find the best estimates of the various features of the objects. For the estimation of size of all the foreground objects in the frame, here, we have taken into account that size is a function of received energy, because for a frame sequence which is a sequence of binary images, the received energy is proportional to the size of the concerned object. Thus we calculate the cross correlation between the consecutive frames, which in turns give us the energy per frame.

$$R_{xy}(\mathrm{T}) = \int_{-\infty}^{\infty} x\ (t - \mathrm{T})\ y(\mathrm{T})^*\ d\mathrm{T} \qquad (14)$$

$$E = \int_{-\infty}^{\infty} x(\mathrm{T})\ x(\mathrm{T})^*\ d\mathrm{T} \qquad (15)$$

Now in terms of frames, the cross correlation between successive frames is given by, $R_{xy}(T)$. It is taken, that the time difference between successive frames is 1/25 seconds, that means, the $(t - T)$ factor has already been taken into consideration. Thus it reduces simply into equation of energy *E*. Hence, if we are able to find the cross correlation between successive frames, then energy per frame can also be calculated. This has been used to estimate the size of the objects, as it has been noted that size is proportional to the reflected energy. The estimation has been done using the Maximum Likelihood Method, in which the received observations are the independent samples of a random variable, $y(t-T) + n(t)$, where $y(t)$ is the transmitted signal, which has been received after a time delay of T, and $n(t)$ is the random noise added due to distortions. Now, we use the dataset *Y*, the random variable, from the tensor to form Gaussian distribution $P(y/\theta)$, where θ here is the size parameter. So, we have a gaussian distribution of the size from the tensor, which is P (Random Dataset of Size | Size Parameter). Subsequently, for maximum likelihood estimation, the objective is to determine the most likely values of the parameter, given an observed sample value. Hence, h_{ml} , that is, the most likely estimation of the size, gives us the best estimate.

2. *Estimation of Displacement*

We can apply similar parametric estimation approach for all the features. We have estimated the displacement of the concerned object per frame. We make use of the same radar theory, a signal is transmitted, and then in all the successive frames the reflected signal is received from different forward or backward positions for a moving object. This gives us the displacement of the object per frame. Suppose at time t, the position of target object is at *A*, and at time $(t+ T)$, the new position is *B.* Then the difference in these two positions gives the displacement of the target object for the current frame. Now, similarly as in the case of estimation of size, the gaussian distribution of the displacement from the tensor, which is P (Random dataset of Displacement | Displacement Parameter) is known. Consequently, that estimate of the

tensor parameter of displacement is adopted as the best estimate, which maximizes the joint probability density function $P(y/\theta)$, where y here is the random variable for the n independent samples of displacement in successive frames and θ is the displacement parameter. Hence, we also fetch the most likely estimated displacement d_{ml}, of the concerned person with the help of the probability distribution function and the displacements per frame.

3. Estimation of Velocity

Likewise, the estimation of velocity is done. Here also likelihood function $L(y/\theta)$ is maximized for the optimal solution, where y is the random variable and θ is the velocity parameter. Using the Normal probability distribution function of velocity from the tensor which is P (Random Dataset of Velocity | Velocity Parameter), we obtain the parameter v_{ml}, that maximizes the likelihood function and hence is the best estimate.

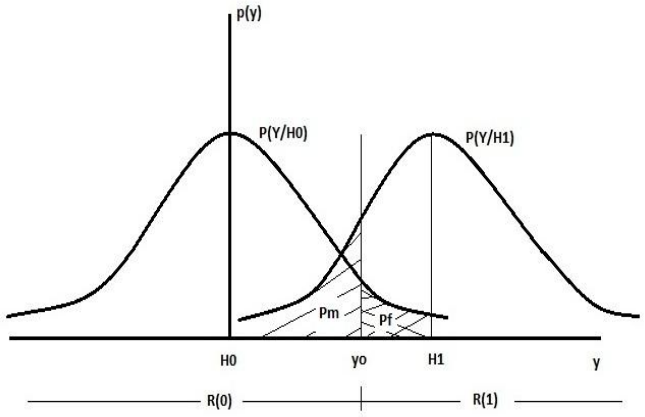
Once all the estimates of the features for all the objects per successive frames are calculated. Then the next step is the determination of the detection probabilities of those foreground objects, based on their estimated data.

2.4.6 Application of Neyman Pearson Theory

Now, assume that we have to classify an object based on the estimated features. Because even if the estimated features are known, the observation at the receiver end is still a random process, and thus overall problem of object detection is eventually a statistical decision problem. We will naturally choose the class that is most probable given the estimated values. The two situations of 'object present' and 'object not present' have simply been taken as two classes H_0 and H_1. These classes are such that,

H_0 : Null Hypothesis – Target Object is not present

H_1 : Target Object is present in the next frame



Figure(2.7): Decision Regions with Conditional Probabilities, False alarm Probability, Probability of Miss and Threshold value of the parameter

The probabilities $p(H_1)$ and $p(H_0)$ are the a priori probabilities, which follows $p(H_1) + p(H_0) = 1$, next the, $p(y/H_0)$ and $p(y/H_1)$ are the conditional probabilities, for the two Hypothesis H_0 and H_1, where y is the random variable parameter for the concerned object. The decision of H_0 or H_1, has to be based on some probabilistic decisive factor. The simplest criterion would be to choose the most likely hypothesis and hence the relative decision rule is given by,

Hypothesis H1, if $p(H1/y) > p(H0/y)$

and

Hypothesis H0, *otherwise*

where, p(H1/y) and p(H0/y) are the a posteriori probabilities, which can be determined using the Bayes Theorem,

$$p\left(H_1/y\right) = \frac{P(H_1)\,p(y/H_1)}{p(y)} \qquad (16)$$

and, similarly,

$$p\left(H_0/y\right) = \frac{P(H_0)\,p(y/H_0)}{p(y)} \qquad (17)$$

Practically, decision making is inherently biased, where different risk values are involved with different decisions. For an instance, in radar applications, the risk of missing the target is certainly higher than the risk of a false alarm. This is where we apply Neyman Pearson Theory [12][14] to know the detection probabilities, and thus classifying the objects. In the case of radar applications not only the priori probabilities but also the cost matrix is not normally known. Also, the Neyman Pearson criterion, used in detection theory, also leads to a Likelihood Ratio test, it basically fixes one class error probabilities, and seeks to minimize the other. So the Neyman Pearson criterion is very attractive since it does not require the knowledge of priors and cost function. In such cases we use a pre assigned value of the false alarm probability (Pf), in our case we have fixed it at Pf = 0.3, which is basically the false detection probability of the object that is affordable. Since the priori probabilities are not known, hence, mathematical formula for Probability of miss (Pm) and false alarm Probability (Pf) are:

$$Pf = \int_{y_0}^{\infty} p(y/H_{01})\, dy \qquad (18)$$

$$Pm = \int_{-\infty}^{y_0} p(y/H_1)\, dy \qquad (19)$$

Where, $p(y/H_0)$ and $p(y/H_1)$ are the conditional probabilities, and y_0 is the threshold value of the parameter calculated using the given Pf, it basically divides the range of y into the region R(0) and R(1).

Now, our problem remains to minimize the risk of miss. It eventually corresponds to maximizing the detection probability (Pd) of the target for a given false alarm probability, which is given by,

$$Pd = (1 - Pm) \qquad (20)$$

This we have solved using the method of Lagrangian multiplier (μ) by minimizing the equation,

$$C = Pm + \mu\, Pf \qquad (21)$$

This is basically the optimization equation, where C is the average cost for risk, which must be minimized. The expanded equation for the average risk, thus becomes,

$$C = \int_{y_0}^{\infty} p(y/H_0)dy + \mu \int_{-\infty}^{y_0} p(y/H_1)dy \quad (22)$$

This, in fact, can be put in the terms of likelihood ratio test. Thus, the condition for minimum average risk C is

$$\lambda_1 = \frac{p(^{y}/_{H_1})}{p(^{y}/_{H_0})} > \mu = \lambda_t \quad (23)$$

Hence, if the likelihood ratio $\frac{p(^{y}/_{H_1})}{p(^{y}/_{H_0})}$ is greater than the lagrangian multiplier, then the average risk associated with the decision is minimized, or in other words, it minimizes the risk of miss and subsequently maximizes the detection probability.
Thus the decision rule is

$$\lambda_1 > \lambda_t \quad \text{for hypothesis } H_1 \quad (24)$$

$$\lambda_1(y) \leq \lambda_t \quad \text{for hypothesis } H_0 \quad (25)$$

These form the basis for object identification, and the detection probabilities give the probability of finding that concerned object in the next frame. The most important benefit of applying Neyman Pearson Criterion is that, it does not only gives the result, but the concerned probabilities supporting the result as well.

Thus we are now with the estimated values of the features of multiple objects in all the frames, with their corresponding detection probabilities. This is then compared with the feature values got from the graph theoretic approach to know the error percentage and accuracy.

CHAPTER 3

RESULTS

Tracked Frames

Figures

Tables

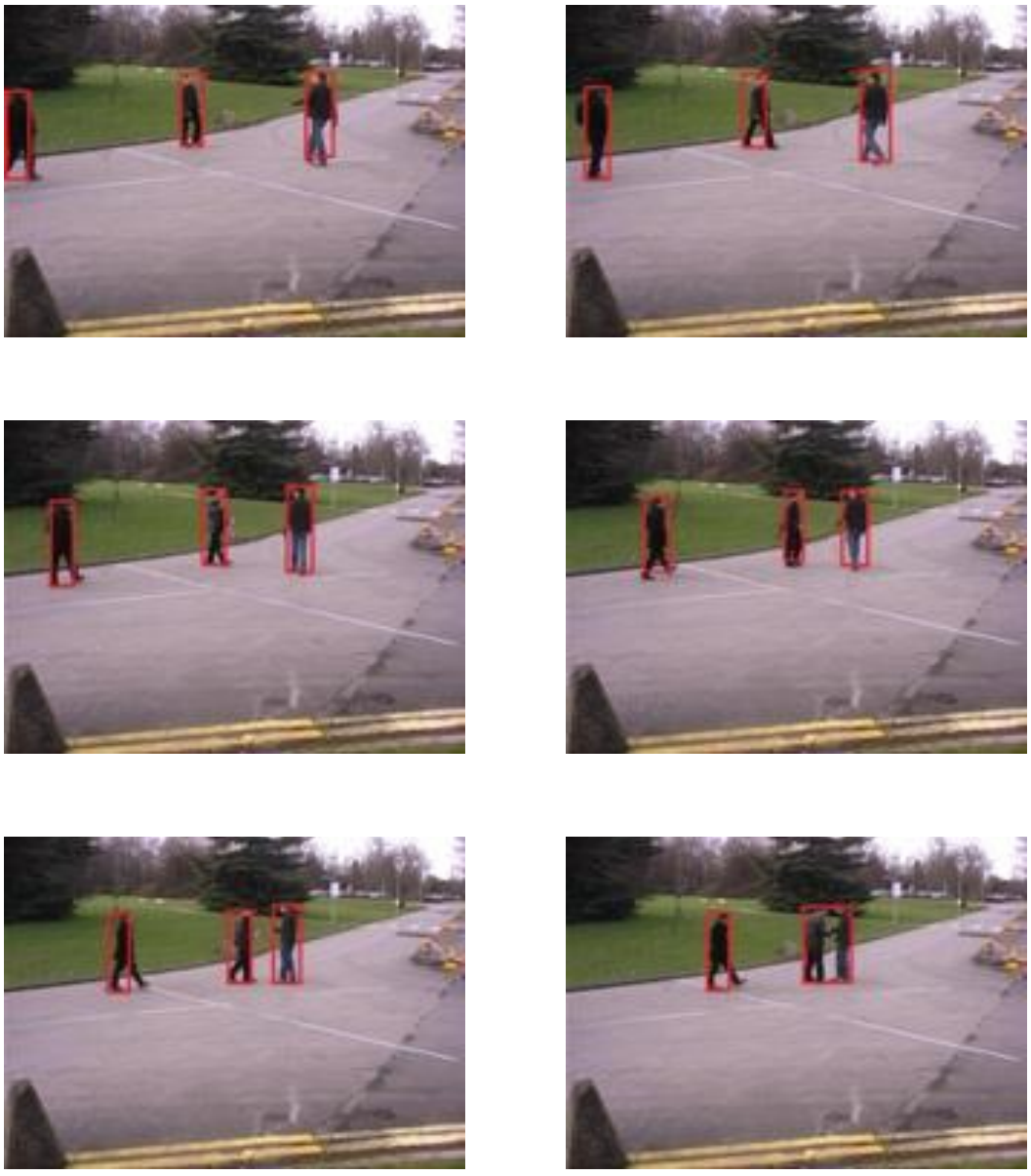
3.1 Results

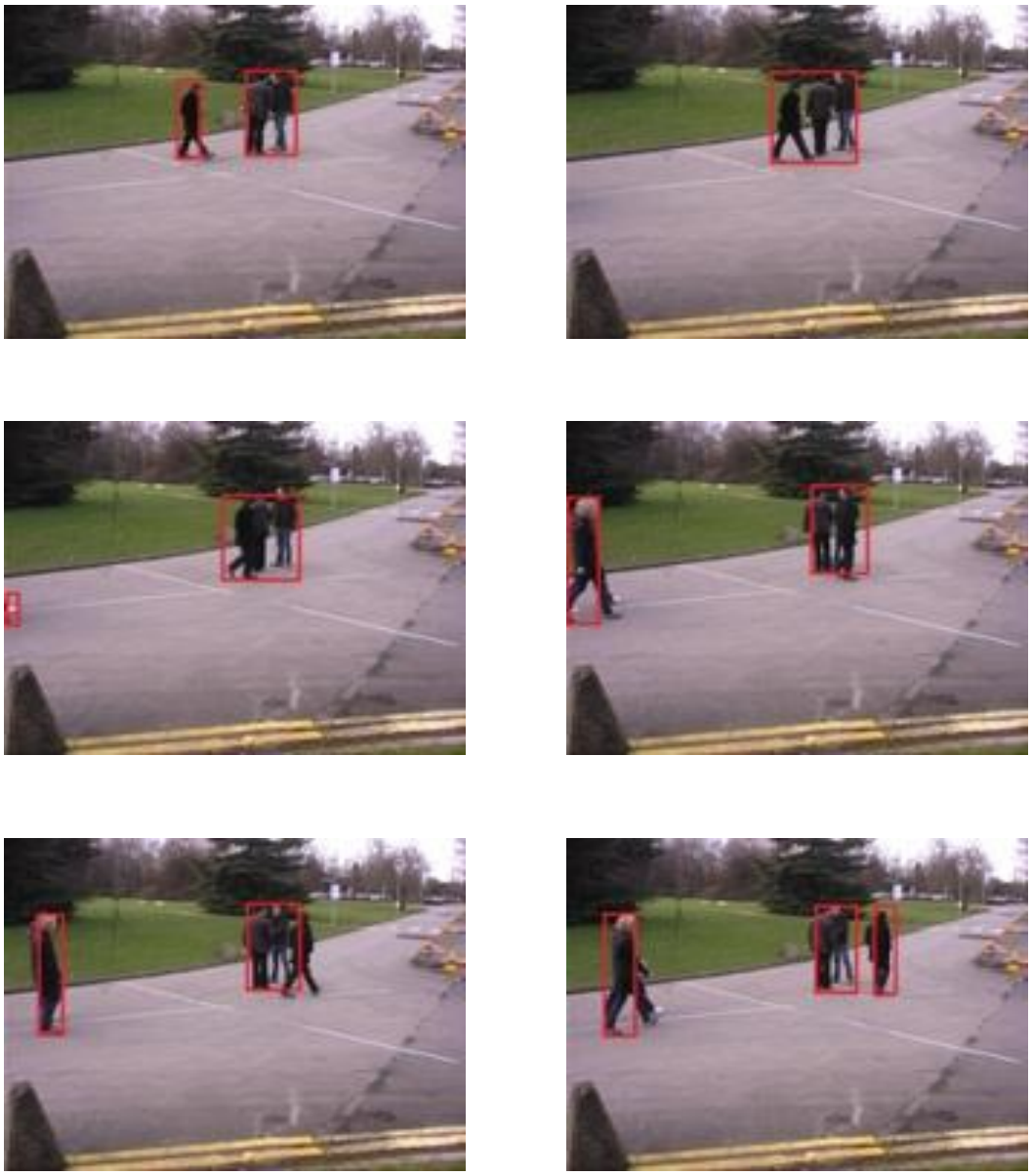
The comparison of the parametric estimation values with the values of the features from graph theoretic approach has been made, with detection probabilities for the next frames, for each estimation.

Figure(3.1): Sample Frame

The video frame below is the sample frame for the object aliasing to simplify the representations. The person at the extreme left is *"Object 1"*, while the one at the middle is *"Object 2"*, and we have named the person at the right as *"Object 3"*.

3.2 Tracked Frames





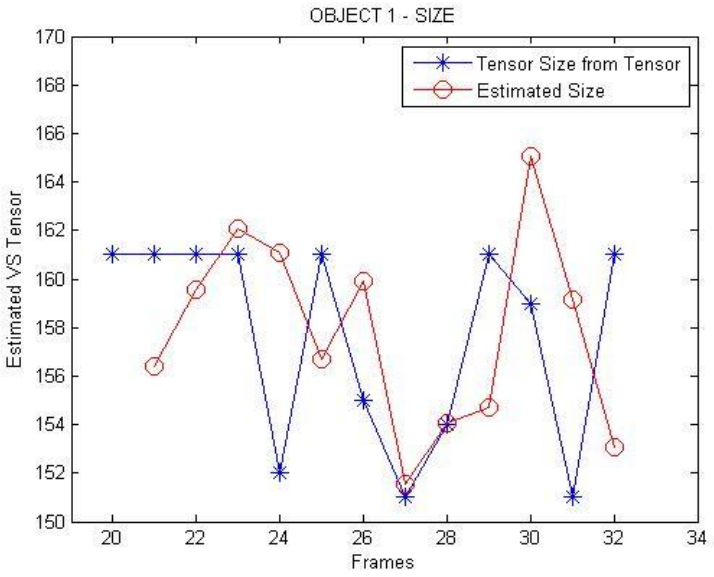
Figure(3.2): Tracked Frames

3.3 Figures

For each feature, that is, size, displacement and velocity of the *"Object 1"*, *"Object 2"* and *"Object 3"*, the estimated value and the tensor value are plotted for each successive frame. These plots shows the estimated feature values for the successive next frames, and how it is differing from the values fetched from Tensor. Thus for each frame, error in estimation can be calculated.

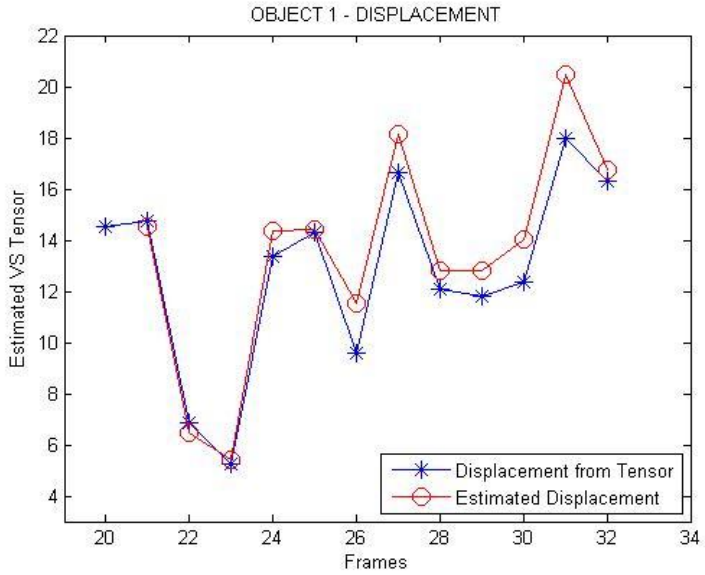
FOR OBJECT 1

A. Size



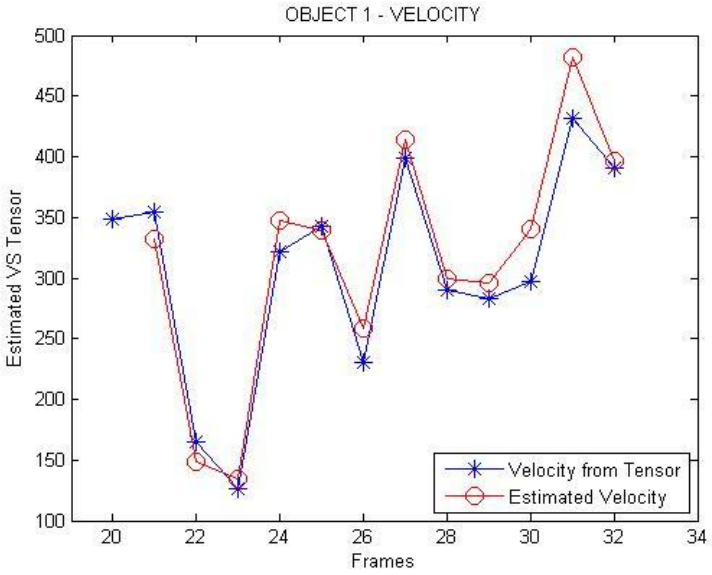
Figure(3.3): Plot to show the variation of Estimated Size with Actual Size with each successive frames for Object 1

B. Displacement



Figure(3.4): Plot to show the variation of Estimated Displacement with Actual Displacement with each successive frames for Object 1

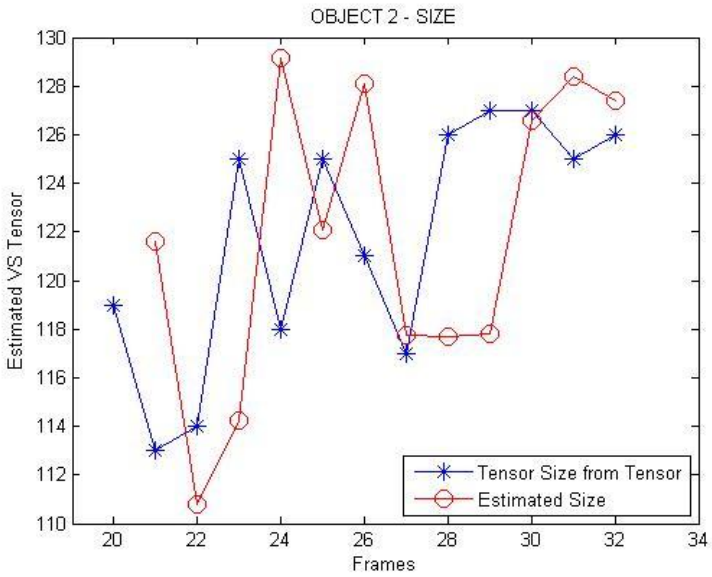
C. Velocity



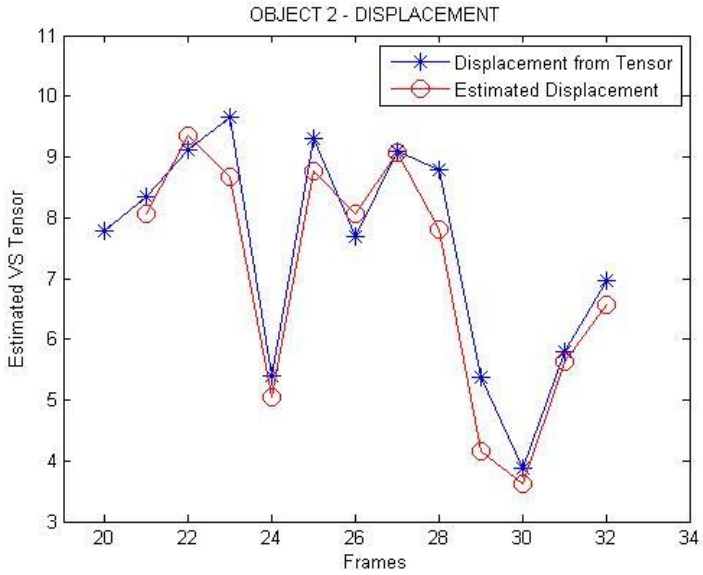
Figure(3.5): Plot to show the variation of Estimated Velocity with Actual Velocity with each successive frames for Object 1

FOR OBJECT 2

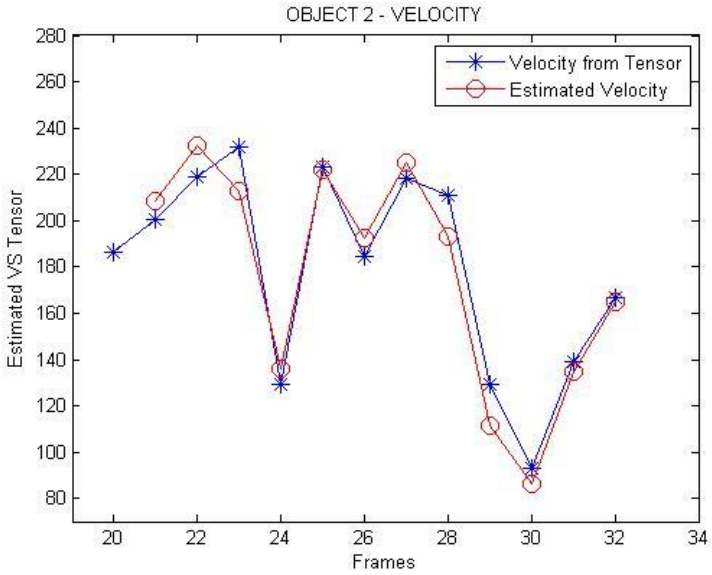
A. Size



Figure(3.6): Plot to show the variation of Estimated Size with Actual Size with each successive frames for Object 2



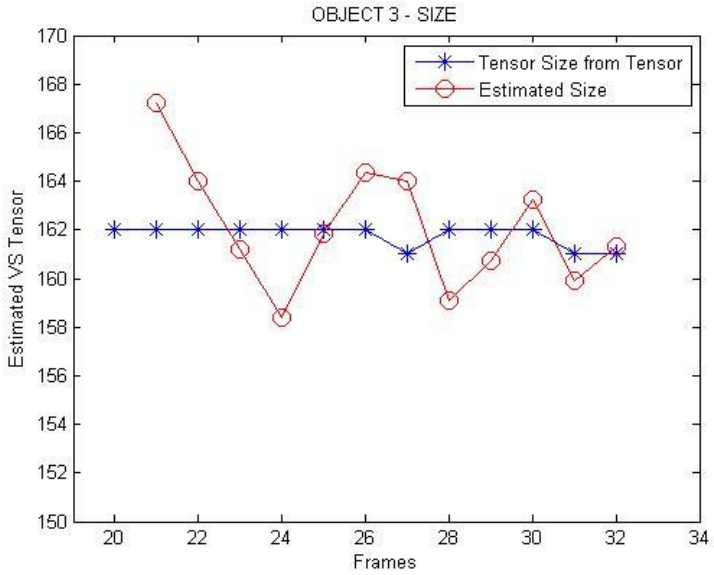
Figure(3.7): Plot to show the variation of Estimated Displacement with Actual Displacement with each successive frames for Object 2



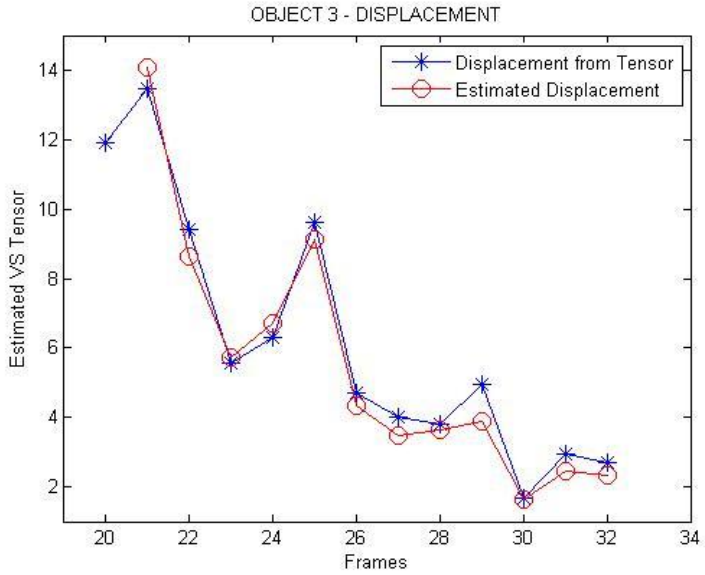
Figure(3.8): Plot to show the variation of Estimated Velocity with Actual Velocity with each successive frames for Object 2

FOR OBJECT 3

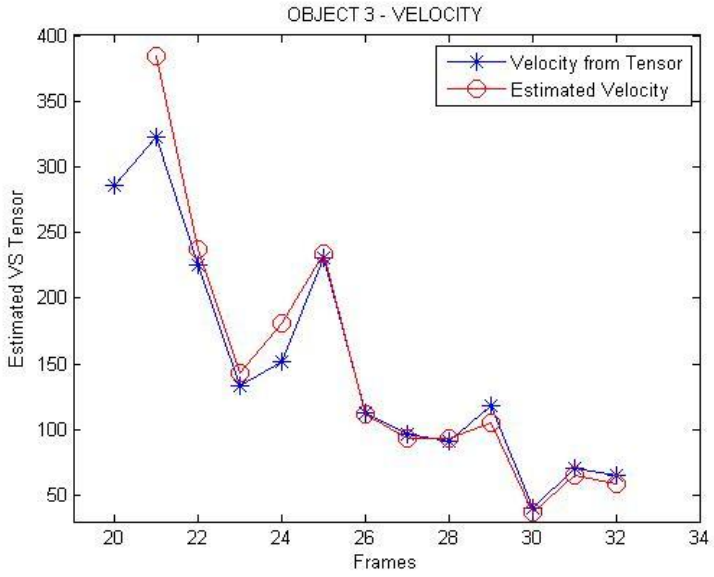
A. Size



Figure(3.9): Plot to show the variation of Estimated Size with Actual Size with each successive frames for Object 3



Figure(3.10): Plot to show the variation of Estimated Displacement with Actual Displacement with each successive frames for Object 3



Figure(3.11): Plot to show the variation of Estimated Velocity with Actual Velocity with each successive frames for Object 3

From the plots it is clear that the change in feature values of size, displacement and velocity follows a unique pattern or shape for successive frames for each object. The estimated values follows the pattern but with slight difference. However, this is enough to recognize the objects. But, the recognition is based on some detection probabilities, which tells whether the object could be detected in the next frame. We have till now the estimated data for each frame. Now Neyman Pearson criterion is applied to calculate the detection probabilities of the objects in successive frames.

3.4 Tables

The detection probabilities and the average risk connected with each decision have been shown in the tabular form for successive frames for all the three objects. The detection probabilities go up whenever the estimated value of the feature confirms with the tensor values. While the risk associated with the detection lowers for a higher detection probability, and increases for a lower detection probability. These results effectively show that the object is efficiently identified.

TABLE 3.1. Results of application of Neyman Pearson Criterion on Object 1

Frames	**Displacement**		**Velocity**		**Size**	
	Detection Probability	***Average Risk***	***Detection Probability***	***Average Risk***	***Detection Probability***	***Average Risk***
21	0.68838	0.80624	0.70606	0.78857	0.70957	0.78504
22	0.85929	0.42517	0.86898	0.41579	0.70452	0.7901
23	0.88756	0.37678	0.88226	0.39029	0.69946	0.79516
24	0.68838	0.79994	0.69259	0.80203	0.70199	0.7896
25	0.68838	0.80183	0.69871	0.78544	0.70957	0.76882

26	0.7741	0.62867	0.77194	0.61102	0.70452	0.78354
27	0.63096	0.86366	0.63127	0.86335	0.71969	0.74372
28	0.74558	0.623	0.73415	0.64075	0.71463	0.75616
29	0.74558	0.623	0.73781	0.63344	0.71463	0.75795
30	0.71701	0.67775	0.69871	0.71473	0.6944	0.80022
31	0.57334	0.92128	0.57095	0.92367	0.70452	0.77277
32	0.65969	0.75282	0.64846	0.76515	0.71716	0.74285

TABLE 3.2. Results of application of Neyman Pearson Criterion on Object 2

Frames	**Displacement**		**Velocity**		**Size**	
	Detection Probability	***Average Risk***	***Detection Probability***	***Average Risk***	***Detection Probability***	***Average Risk***
21	0.6724	0.59728	0.66766	0.61276	0.70907	0.67113
22	0.61715	0.67023	0.63027	0.66423	0.73527	0.62085
23	0.6724	0.60558	0.66107	0.62174	0.72544	0.6381
24	0.78238	045031	0.78368	0.4577	0.68942	0.70851
25	0.6724	0.60668	0.64568	0.64272	0.70907	0.67213
26	0.6724	0.59728	0.694	0.57733	0.6927	0.70268
27	0.61715	0.66612	0.64128	0.64885	0.71889	0.65244
28	0.6724	0.59404	0.69181	0.57976	0.71889	0.65231
29	0.83709	0.38585	0.82505	0.40409	0.71889	0.65268
30	0.83709	0.37998	0.86413	0.35357	0.69597	0.70704
31	0.78238	0.47224	0.78586	0.46995	0.6927	0.7283
32	0.72748	0.63961	0.73781	0.63198	0.69597	0.7921

TABLE 3.3. Results of application of Neyman Pearson Criterion on Object 3

Frames	**Displacement**		**Velocity**		**Size**	
	Detection Probability	***Average Risk***	***Detection Probability***	***Average Risk***	***Detection Probability***	***Average Risk***
21	0.29037	1.2042	0.24533	1.2493	0.69758	0.79703
22	0.57711	0.7587	0.53795	0.79924	0.70497	0.78017
23	0.71921	0.55455	0.72283	0.54149	0.70989	0.76716
24	0.64828	06451	0.65071	0.64043	0.71481	0.75438
25	0.5057	0.8427	0.54334	0.79151	0.70743	0.7715
26	0.78987	0.45795	0.784	0.45906	0.70497	0.78123
27	0.78987	0.4431	0.81847	0.41371	0.70497	0.78017
28	0.78987	0.44608	0.82112	0.41083	0.71235	0.75873
29	0.78987	0.45012	0.797705	0.44184	0.70989	0.76581
30	0.93029	0.27429	0.92927	0.27135	0.70497	0.77804
31	0. 86023	0.35668	0.87398	0.34151	0.71235	0.76108
32	0. 86023	0.3546	0.88716	0.32501	0.70989	0.7676

CHAPTER 4

CONCLUSION

Quantitative Measures

Comparison

Conclusion

4.1 Quantitative Measure

For the determination of the accuracy of results and comaparative appraisal with other tracking algorithms frame based evaluation has been done. Basically detection is measured which is the location of the target object with respect to the ground truth locations. Thus the detections which we have generated with our system becomes the material for the comparative analysis with the ground truth locations. There are three terminologies involved with this, they are: True Positive (TP), False Negative (FN) and False Positive (FP). TP refers to the situation when our detected object largely overlaps with the ground truth location. While, FN refers to the situation it fails entirely. And, FP here refers to the situation when in the presence of detection, it does not overlap the ground truth detection.

We also have some quantitative measures for the purpose of assessment of results, such as, Precision (P), Tracking Accuracy (TA), and False Alarms per Frame (FA/Frame).

$$Precision = \frac{True\ Positive}{Total\ Detection}$$

$$Tracking\ Accuracy = \frac{Accurately\ Tracked\ Frames}{Tracked\ Frames}$$

$$FA/_{Frames} = \frac{False\ Alarms}{Total\ number\ of\ Frames}$$

The tracking is said to be accurate if the predicted point of position of object falls onto the same Ground Truth position for each frame. While, FA/Frame is defined simply as the false alarms per frame.

In the following table, a comparison of our methodology with the previously worked upon approaches has been made on the basis of P, TA and FA/Frame. For better tracking the detection probability, the tracking accuracy and the precision should be high whereas the false alarm per frame should be as low as possible.

4.2 Comparison

TABLE IV. Quantitative Comparison

Methods	Quantitative Measure		
	TA	P	FA/Frame
Huang et al.[30]	71.1 %	68.5 %	0.98
Leibe et al.[31]	79.1 %	73.1 %	0.38
Zhao et al.[32]	82.4 %	79.7 %	0.21
Ours	93.2 %	82.4 %	0.3

The above results show that our method achieves the best performance with greater TA, greater Precision, and low FA/Frame. The testing frames are with frame rate of 15 fps and frame size of 352 × 288 pixels has been taken. The experimental results were measured on Intel Core 2 Duo, 2.80 GHz machine. Average processing capability of our system is 3-5 frames per second.

4.3 Conclusion

We have worked upon a state of the art tracking system that is able to detect multiple objects, and track them simultaneously, taking note of concerned estimated data and detection probabilities. Our method efficiently finds out the next state, that is, the estimated state for the next frame using optimal parametric estimation and to make it robust and accurate, we have applied Neyman Pearson criterion which suggests the efficacy of the estimated state through probabilistic tracking assignment. This adds more capability in our tracking algorithm, as we have seen while comparing with some of the previous techniques.

REFERENCES

[1] E. Saykol, U. Gudukbay, and O. Ulusoy, “A histogram-based approach for object-based query-by-shape-and-color in multimedia databases,” Technical Report BUCE-0201, Bilkent University, 2002.

[2] A. M. McIvor, “Background subtraction techniques,” In Proc. of Image and Vision Computing, Auckland, New Zealand, 2000.

[3] A. J. Lipton, H. Fujiyoshi, and R.S. Patil, “Moving target classification and tracking from real-time video,” In Proc. of Workshop Applications of Computer Vision, pages 129–136, 1998.

[4] J. Heikkila and O. Silven, “A real-time system for monitoring of cyclists and pedestrians,” In Proc. of Second IEEE Workshop on Visual Surveillance, pages 74–81, Fort Collins, Colorado, June 1999.

[5] F. Heijden, “Image Based Measurement Systems: Object Recognition and Parameter Estimation,” Wiley, January 1996.

[6] A. Amer, “Voting-based simultaneous tracking of multiple video objects,” In Proc. SPIE Int. Symposium on Electronic Imaging, pages 500–511, Santa Clara, USA, January 2003.

[7] M. Han, A. Sethi, Y. Gong, “A detection based multiple object tracking method,” University of Illinois at Urbana Champaign, Champaign, IL, USA.

[8] L. I. Perlovsky, R. W. Deming, “Maximum Likelihood Joint Tracking and Association in a Strong Clutter” IEEE.

[9] M. Isard, J. MacCormick, “BraMBLe: A Bayesian Multiple-Blob Tracker,” Compaq Systems Research Center Palo Alto, CA 94301, USA.

[10] A. Yilmaz, O. Javed, M. Shah, “Object tracking: A survey,” ACM Comput. Surv. 38, 4, Article 13 (Dec. 2006).

[11] Gonzalez, “Digital Image Processing,” Addison Wesley Longman Publishing Co., Inc., Boston, MA.

[12] J. Das, S. K. Mullick, P. K. Chatterjee, “Principles of Digital Communication,” New Age International (P) Limited Publishers, Reprint 2002.

[13] J. Zhong, S. Sclaroff, “Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter,” Ninth IEEE International Conference on Computer Vision (ICCV 2003).

[14] R. M. Royall, “Statistical Evidence: A likelihood paradigm,” London Chapman and Hall.

[15] C. Hue, P. Perez, “Tracking Multiple Objects with Particle Filtering,” IEEE Transactions On Aerospace And Electronic Systems Vol. 38, No. 3 July 2002.

[16] X. Li, K. Wang, Y. Li, W. Wang, “A Multiple Object Tracking Method Using Kalman Filter,” International Conference on Information and Automation June 20 - 23, Harbin, China, IEEE 2010.

[17] Y. Wang, Z. Li, Y. Wang, F. Chen, “A Bayesian Non-parametric Viewpoint to Visual Tracking,” IEEE 2013

[18] Reza Hoseinnezhad, Ba-Ngu Vo, and Ba-Tuong Vo, “Visual Tracking in Background Subtracted Image Sequences via Multi-Bernoulli Filtering,” IEEE Transactions On Signal Processing, Vol. 61, No. 2, January 15, 2013.

[19] W. Limprasert, A. Wallace, G. Michaelson, “Real-Time People Tracking in a Camera Network,” IEEE Journal On Emerging And Selected Topics In Circuits And Systems, March 18, 2013.

[20] Hidetomo Sakaino, “Video based Tracking, Learning, And Recognition Method For Multiple Moving Objects,” IEEE Transactions On Circuits And Systems For Video Technology, Vol. X, No. Y, Z 2013.

[21] S. Wei, L. K. Wang, J. H. Lan, “Moving Object tracking Based on Background Subtraction Combined Temporal Difference,” International Conference on Emerging Trends in Computer and Image Processing (ICETCIP'2011) Bangkok Dec., 2011.

[22] LIU, Y.; Haizho, A. & Xu Guangyou, “Moving object detection and tracking based on background subtraction, Proceeding of Society of Photo-Optical Instrument Engineers” (SPIE), Vol. 4554, pp. 62-66,2001.

[23] Stringa, E., (2000), “Morphological Change Detection Algorithms for Surveillance Applications,” IEEE International Conference on Tools with Artificial Intelligence (ICTAI.00).

[24] Desa, S. M. & Salih, Q. A. (2004), "Image subtraction for real time moving object extraction," Proceeding of Int. Conf. on Computer Graphics, Imaging and Visualization (CGIV'04), pp. 41–45.

[25] C.D.Badgujar, D. P. Sapkal, "A Survey on Object Detect, Track and Identify Using Video Surveillance," IOSR Journal of Engineering (IOSRJEN) Volume 2, Issue 10 (October 2012), PP 71-76.

[26] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, "Recent advances and trends in visual tracking: A review," Neurocomputing, Elsevier, 2011.

[27] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, A.Hengel, "A Survey of Appearance Models in Visual Object Tracking," ACM Transactions on Intelligent Systems and Technology, 2013.

[28] K. A. Joshi, D. G. Thakore, "A Survey on Moving Object Detection and Tracking in Video Surveillance System," International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012.

[29] S. K. Patel, A. Mishra, "Moving Object Tracking Techniques: A Critical Review," Indian Journal of Computer Science and Engineering (IJCSE) Vol. 4 No.2 Apr-May 2013.

[30] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in Proc. Eur. Conf. Comp. Vision, 2008, pp. 788–801.

[31] B. Leibe, K. Schindler, and L. V. Gool, "Coupled detection and trajectory estimation for multi-object tracking," in Proc. Int. Conf. Computer Vision, 2007, pp. 1–8.

[32] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," IEEE Trans. Pattern Anal. Mach. Intelligence, vol. 30, no. 7, pp. 1198–1211, Jul. 2008.

[33] M. Khare, T. Patnaik, A. Khare, "Dual Tree Complex Wavelet Transform based video object tracking," Communications in Comupter and Information Science Volume 101, pp 281-286, Springer Link, 2010

[34] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," IEEE A&E systems magazine vol. 19, no. 1 January 2004.