Abstract

This thesis discusses the study of object tracking technique based upon background features/parameters. This concept is based upon the scientific principle of relative localization and the principle of natural compression with the increase in the distance from the location of camera. The second principle has been utilized for depth modelling of the foreground objects and the background information has been used not only to improve the depth modelling but also to learn to better localize. Method shows a definite improvement in the object localization process. Method also works in real time being having no computationally intensive algorithm.

Chapter 1

Introduction

Object tracking is a well-researched problem in computer vision and has many practical applications. The objective of target localization or object tracking is to faithfully locate a previously specified target object in subsequent video frames. The problem and its difficulty depend on several factors, such as the amount of prior knowledge about the target object and the number and type of parameters being tracked (e.g., position, velocity, acceleration, size, detailed contour). Different cases may occur, the object to be tracked can be any mobile object in a scene or it can be a specific object that is already known and the trajectory of which is to be followed. Here we are interested in the second case. It is possible to learn the considered object in a first time when representations of this object are available. Once the learning has been performed, object tracking consists in object occurrence contextual searching in successive frames of a video sequence. Object Tracking has remained a challenging task because an object can drastically change appearance when deforming, rotating out of plane, or when the illumination of the scenario changes. Most tracking algorithms are monocular and model based, i.e., an initial model of the object to be tracked is used to detect its position in video frames. The initial model could be in the form of a histogram, pre-defined shape, edge map or a high level model like interest points. The choice of the model is important because it should be both selective and robust. Tracking becomes so difficult when the tracking is to be done in dense place, which may be dense by movable or immovable objects. So, in that case or scenario

the tracking is helpful with the help of the fixed objects in the background, from which we can locate the region of interest in the scenario. Science explains us that to locate anyone's position in any scenario, you have to point out your ROI relative to some object in that scenario, for e.g. that person is at so & so position to the left of that tree, here the ROI is the person and the tree is reference object with the help of which ROI is located. So, in that manner the ROI are being located in real science, utilizing that only rule of science, we have incorporated in our system of tracking. We have divided our system in two categories: first, without background consideration & and second, with background consideration. Within these two categories, we have used three algorithms i.e. HMM, Fuzzy Logic & SVM with which we are going for the predicting new states of the ROI in the scenario and after the prediction from all the three sources for both the categories, the conflict is being resolved by the DSmT and the final decision is formed in the Dezert-Smarandache theory (DSmT) PCR6 rule, in terms of the better tracking technique. The proposed scheme is simple and provides an effective tracking.

1.1 Overview of Related Work

Many authors have proposed and incorporated number of techniques for object tracking, which may consist of the tracking single object and tracking multiple objects in different environments. Object tracking finds applications in activity analysis, video understanding, video segmentation etc. Tracking can be formulated as the correspondence of detected objects represented by points, kernel & silhouette across the frames. Point correspondence methods can be divided into two broad categories, namely, deterministic and statistical methods. The deterministic methods

use qualitative motion heuristics [19] to constrain the correspondence problem. On the other hand, probabilistic methods explicitly take the object measurement and take uncertainties into account to establish correspondence. [24] Solve the correspondence by a greedy approach based on the proximity and rigidity constraints. Their algorithm considers two consecutive frames and is initialized by the nearest neighbor criterion. 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. Templates and density-based appearance models have been widely used because of their relative simplicity and low computational cost. Image intensity is very sensitive to illumination changes, image gradients [25] can also be used as features. More efficient algorithms for template matching have been proposed [26]. [28] Use a weighted histogram computed from a circular region to represent the object. Instead of performing a brute force search for locating the object, they use the mean-shift procedure. Mean-shift tracking requires that a portion of the object is inside the circular region upon initialization [9]. [29] Propose joint modeling of the background and foreground regions for tracking. The background appearance is represented by a mixture of Gaussians. Appearance of all foreground objects is also modeled by mixture of Gaussians. The shapes of objects are modeled as cylinders. They assume the ground plane is known, thus the 3D object positions can be computed. Tracking is achieved by using particle filters where the state vector includes the 3D position, shape and the velocity of all objects in the scene. Shape matching can be performed similar to tracking based on template matching, where an object silhouette and its associated model is searched in the current frame. [30] Performed shape matching using an edge-based representation. Some of the popular object tracking algorithms include the SSD [8], Mean Shift Tracking [9], Optical Flow [10], Semantic Tracking

[11], Feature matching and Contour tracking [17, 18]. Foreground detection, robustness, illumination changes, occlusion, accuracy, presence of clutter and reducing the computational complexity have been only some of the main challenges of object tracking. [32] Propose a contour tracker where the contour is parameterized as an ellipse. Each contour node has an associated HMM and the states of each HMM are defined by the points lying on the lines normal to the contour control point. The observation likelihood and the state transition probabilities, the current contour state is estimated using the Viterbi algorithm [33]. After the contour is approximated, an ellipse is fit to enforce elliptical shape constraint. Several authors have even proposed their own Markovian models in order to deal with object tracking [1] [2], Real-time discrete state estimation is carried out either through Hidden Markov Models (HMM)[5], through Fuzzy Logic[14], and [34] used a Support Vector Machine (SVM) classifier for tracking. SVM is a general classification scheme that, given a set of positive and negative training examples, finds the best separating hyperplane between the two classes SVM [15]. This thesis takes advantage of natural process of variation of average speed V/s distance from camera to convert a 2D image into an approximate 3D model. Thesis also uses the background for getting advantage in object tracking. The Dezert-Smarandache theory (DSmT) can be considered as a generalization of the DST theory [3] with a number of PCR rules for fusion [4]. In this paper we have used the DSmT PCR6 rule for the fusion of sources and giving the final decision in terms of the better tracking technique.

1.2 Scope of the Thesis

This thesis is devoted to the scientific rule of nature of tracking or localizing ones in any scenario with the background reference points with add on as occlusion handling.

1.3 Outline of the Thesis

This thesis is formulated as,

In the second chapter, the methodology of the system is explained step by step and which also incorporates the image processing part comprising of the background subtraction & image segmentation, nearest neighbour data association part along with graph theoretic approach.

In the third chapter, the depth modelling is explained which explains the variation of speed with respect to the variation in distance, 2D parameter estimation with which we approximating the 3D mapping of the whole scenario and through which all the 3D parameters are estimated. As the 3D mapping is done for all the objects the occlusion is handled in a better way or we could say nearly the occlusion is prevented.

In the fourth chapter, we proposed the state estimation algorithms, which comprises of three parts, first estimation is done by a group of HMM in terms of the Graph Theoretic parameters and a final decision is formulated by a decision HMM. Second estimation is done with the help of Fuzzy Logic, the rules are being proposed with the help fuzzy inference system for object state prediction, and the third estimation with the help of SVM, the prediction is done in terms of the objects state in the future frames. All the three estimation is individually done for both the categories.

In the fifth chapter, after the prediction from all the three sources/algorithms for both the categories, the data fusion is done with the Dezert-Smarandache theory (DSmT) model, then combination rules are being incorporated and the conflict between different sources is being resolved by the DSmT and the final decision is formed

using the DSmT PCR6 rule for the fusion of sources and giving the final decision in terms of the better tracking technique. The proposed scheme is simple and provides an effective tracking.

In sixth chapter, experimental results are demonstrated with and without the consideration of background features.

In seventh chapter, the comparison with other authors proposed estimation algorithms is done with some performance metrics of the system, also includes the conclusion and the future scope of the project.

Last chapter depicts the references which are being referred for the motivation and implementation of our system.

Chapter 2
Methodology

The methodology is being organized in a hierarchy manner as object detection, object representation, object identification, 3D modeling, predicting algorithms and the final decision as inscribed in figure 2.1. Below are the proposed methods steps in hierarchy manner:

Step1: with the video dataset the frames structure is formed.

Step2: the object is detected with the help of frame differencing method, in which the previous frame is subtracted from the current frame for moving object detection with the help of appropriate threshold.

Step3: The detected objects in the frame are being represented with the help of region-props properties as centroids and boundingbox, the boundingbox enclosing the object and the centroid representing the center of gravity of the detected object. Each particular centroid or detected object is represented with a particular node in the scene.

Step4: As the frame proceeds from current frame to next frame than each object or node is recognized with the help of nearest neighbour classification, in which the minimum distance is being calculated between the nodes in current frame with respect to the next frame. The sequence of the visited nodes by the current node is the output of the algorithm. The nearest neighbour algorithm is easy to implement and executes quickly.

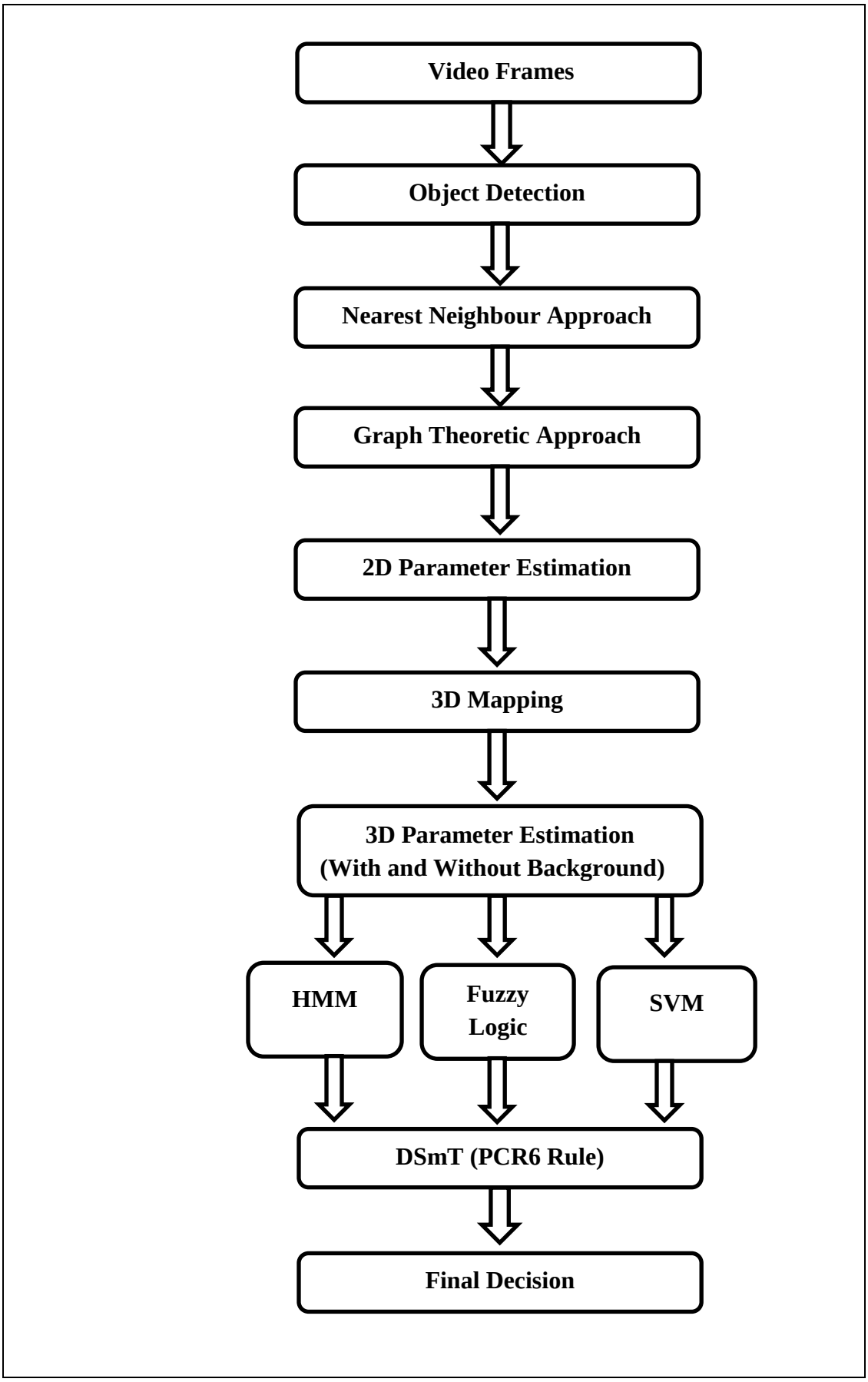

Figure 2.1: Block Diagram of the system

Step5: Graph theoretic approach: with the centroids of the objects it represents the x & y co-ordinates of the objects position in the scene or frame leading us to our first parameters in terms of position(x, y) of the object in the scene. As all objects are represented with centroids having different x & y co-ordinates according to their position, the relative distances are to be calculated between each centroid in the scene or frame with respect to the camera leading us to our second parameter in terms of distances between the objects of the scene or frame. A Tensor is prepared which comprises of these parameters for each frame, from where the parameter data can be retrieved for further processing. With the help of the frame-rate of video and the computed relative distances the third parameter is evaluated as speed, along with the direction of movement of each object the velocity vector is putted as another parameter in the tensor.

Step6: Then with the help of 2D data from the tensor, the 3D plot is drawn between x co-ordinate, y co-ordinate & the relative distances respectively along the 3 axis of the 3D plot, the plot also describes that the object paths do not occlude with each other as in case of 2D, so occlusion is removed. With the nodes in the 3D plot, each node representing each object comprising of three information in terms of x, y & distance between them. A reconstruction of 3D view of the position of the objects in the scene is formed with the help of monocular vision or 2D view through a single static camera. Now, for each frame a 3D plot is being constructed.

Step7: With all the 3D plots corresponding to all frames, again the three 3D parameters are computed i.e. 3D position, 3D Distance, 3D velocity & 3D acceleration.

Step8: Three HMM models are being constructed for all the three parameters as 3D position, 3D velocity & 3D acceleration. For all the three models the transition & the

observation probability is assigned through the training of these models from the previous sample frames. With the help of these three Hidden-Markov Models a collective prediction is being done in terms of the object position, velocity & acceleration for the future frames.

Step9: The inputs and outputs variables in terms of 3D parameters are assigned for the training of the system. With the help of Clustering & Fuzzy Inference System (FIS) relationship the rules are generated with respect to the triangular membership function. The future values are computed with the above rules in terms position of the object.

Step10: The clusters are formed for x coordinate and the y co-ordinates corresponding the objects position and training is done with the help of it. New position in terms of x & y co-ordinates is being classified within the trained clusters resulting us with new positions of object in the future frames.

Step11: Now, taking background into consideration, in these the background is not subtracted from the frames with respect to the objects in the frame. Few major or unique stationary points in the background are being referred to as the background reference points whose center of gravity are being represented with help of the centroids, giving us the position of that particular fixed background objects. As in real science, we used to judge any objects positions with help of some reference/fixed objects in the background, so, using that particular thing in our project also, we will also be predicting the new position of the moving object in the video. Repeating the steps from 3 to 10, now with background will give us more correct results.

Step12: With each method a probability is assigned according to the verification of the output results for both categories with background & without background

considerations. Using the Dezert-Smarandache Theory of conflicting Masses, this will give us the final result in terms of best or fused decision of the whole system.

2.1 Image Processing

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images is referred to as imaging.

2.1.1 Object detection

Performance of an automated visual surveillance system considerably depends on its ability to detect moving objects in the observed environment. A subsequent action, such as tracking, analysing the motion or identifying objects, requires an accurate extraction of the foreground objects, making moving object detection a crucial part of the system. In order to decide on whether some regions in a frame are foreground or not, there should be a model for the background intensities. This model should also be able to capture and store necessary background information. Any change, which is caused by a new object, should be detected by this model, whereas un-stationary background regions, such as branches and leafs of a tree or a flag waving in the wind, should be identified as a part of the background. In this thesis we propose method handle those problem related to un-stationary background such as branches and leafs

of a tree by reducing the resolution of the image. Our object detection method consists of two main steps. The first step is pre-processing step including gray scaling, smoothing, and reducing image resolution and so on. The second step is filtering to remove the image noise contained in the object. The filtering is performed by applying the morphology filter such as dilation and erosion. And finally connected component labeling is performed on the filtered image. The entire process of moving object detection is illustrated in figure 2.2.

2.1.1.1 Pre-processing

The first step on the moving object detection process is capturing the image information using a video camera. Image is capture by a video camera as 24 bit RGB (red, green, blue) image which each color is specified using 8-bit unsigned integers (0 through 255) that representing the intensities of each color. The size of the captured image is 320x240 pixels. This RGB image is used as input image for the next stage. In order to reduce the processing time, gray-scale image is used on entire process instead of color image. The gray-scale image only has one color channel that consists of 8 bit while RGB image has three color channels. The color conversion between gray-scale image and RGB image is defined by the following equation:

$$\boldsymbol{Y = 0.3 \times R + 0.59 \times G + 0.11 \times B} \quad (1)$$

Where, Y is gray-scale image, R is red, G is green and B is Blue of RGB image, respectively. Image smoothing is performed to reduce image noise from input image in order to achieve high accuracy for detecting the moving objects. The smoothing process is performed by using a median filter with m×m pixels.

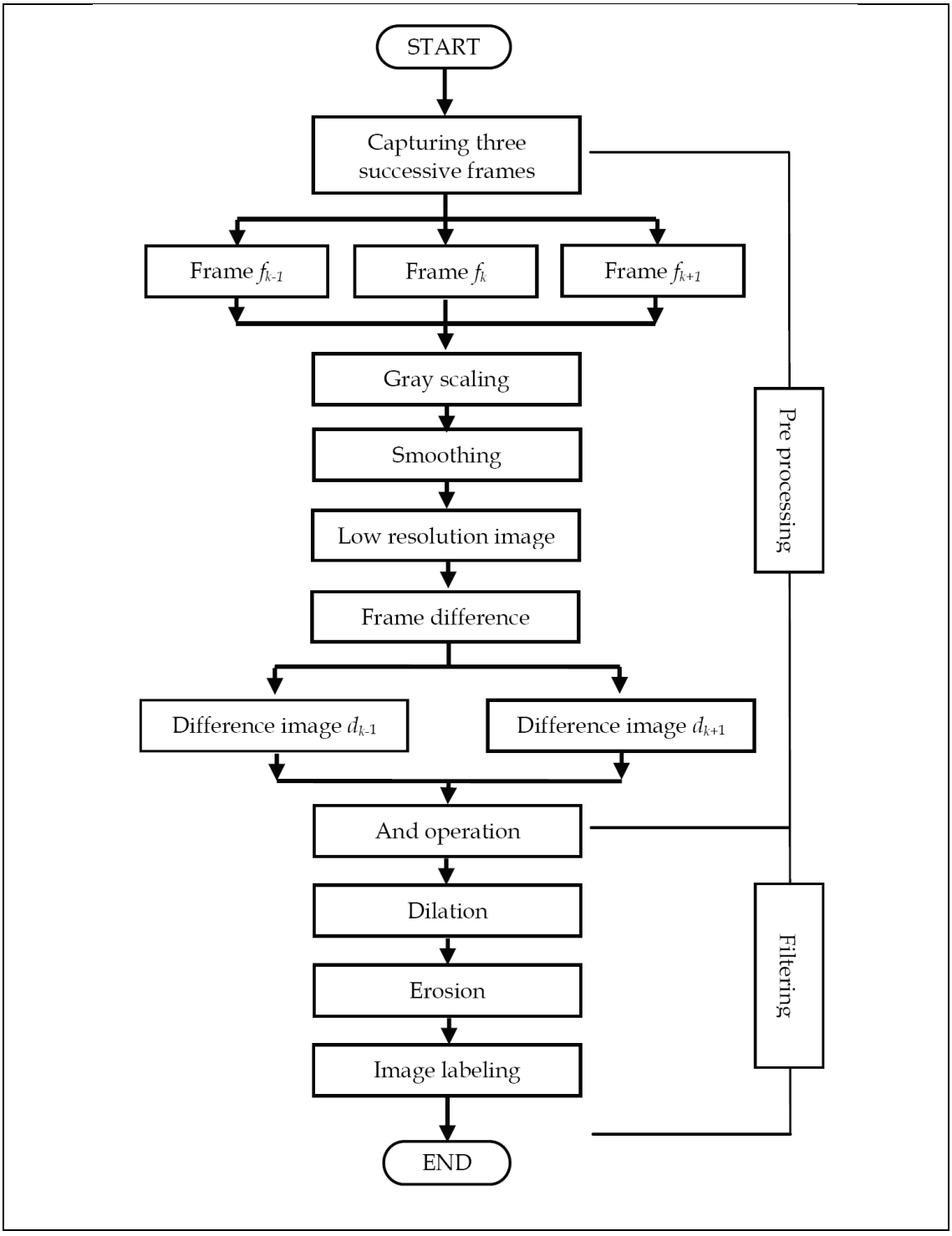

Figure 2.2: Flow of object detection

We consider un-stationary background such as branches and leaf of a tree as part of the background. The un-stationary background often considers as a fake motion other than the motion of the object interest and can cause the failure of detection of the

object. To handle this problem, we reduce the resolution of the image to be a low resolution image. A low resolution image is done by reducing spatial resolution of the image with keeping the image size [36] and [37]. In this article, the low resolution image is done by averaging pixels value of its neighbors, including itself. We use a video image with resolution 320x240 pixels. The original image size is 320x240 pixels. After applying the low resolution image, the numbers of pixels will be 160x120, 80x60, or 40x30 pixels, respectively, while the image size is still 320x240 pixels. The low resolution image can be used for reducing the scattering noise and the small fake motion in the background because of un-stationary background such as leaf of a tree. These noises that have small motion region will be disappeared in low resolution image.

To detect the moving object from the background based of image subtraction, generally there are three approaches can be performed: (i) background subtraction as discussed in [38], (ii) frame difference as discussed in [39], and (iii) combination of background subtraction and frame difference as discussed in [40]. Background subtraction is computing the difference between the current and the reference background image in a pixel-by-pixel. Frame difference is computing the difference image between the successive frames image. In this article, we applied frame difference method to detect the moving objects. In our case, frame difference method is performed on the three successive frames, which are between frame fk and fk-1 and between frame fk and fk+1. The output image as frame difference image is two difference images dk-1 and dk+1 as expressed in Eq. (2). Threshold is performed by threshold value T on the difference image dk-1 and dk+1 as defined in Eq. (3) to distinguish between the moving object and background.

$$\boldsymbol{d_{k-1} = |f_k - f_{k-1}|} \quad (2)$$

$$\boldsymbol{d_{k+1} = |f_k - f_{k+1}|}$$

$$\boldsymbol{d'_k(x,y) = \begin{cases} 1, & d'_k(x,y) > T \\ 0, & otherwise \end{cases}} \quad (3)$$

Where, *k*' =*k*- 1 and *k*+ 1 .

The process is followed by applying AND operator to d_{k-1} and d_{k+1} as expressed in Eq. (4). The output image of this operation is named as motion mask m_p.

$$\boldsymbol{m_p = d_{k-1} \cap d_{k+1}} \quad (4)$$

2.1.1.2 Filtering

In order to fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour, a morphological operation is applied to the image. As a result, small gaps between the isolated segments are erased and the regions are merged. To extract the bounding boxes of detected objects, connected component analysis was used. We find all contours in image and draw the rectangles around corresponding contours with minimum area. Since the image may contain regions which are composed of background noise pixels and these regions are smaller than actual motion regions, we discard the region with a smaller area than the predefined threshold. As a result, the processing produces perfect bounding boxes. Morphological operation is performed to fill small gaps inside the moving object and to reduce the noise remained in the moving objects [41]. The morphological operators implemented are dilation followed by erosion. In dilation, each background pixel that is touching an object pixel is changed into an object pixel. Dilation adds

pixels to the boundary of the object and closes isolated background pixel. Dilation can be expressed as:

$$f(x,y) = \begin{cases} \mathbf{1}, & \boldsymbol{if\ there\ is\ one\ or\ more\ pixels\ of\ the\ 8\ neighbors\ are\ 1} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \quad (5)$$

In erosion, each object pixel that is touching a background pixel is changed into a background pixel. Erosion removes isolated foreground pixels. Erosion can be expressed as:

$$f(x,y) = \begin{cases} \mathbf{0}, & \boldsymbol{if\ there\ is\ one\ or\ more\ pixels\ of\ the\ 8\ neighbors\ are\ 0} \\ \mathbf{1}, & \boldsymbol{otherwise} \end{cases} \quad (6)$$

Morphological operation eliminates background noise and fills small gaps inside an object. This property makes it well suited to our objective since we are interested in generating masks which preserve the object boundary.

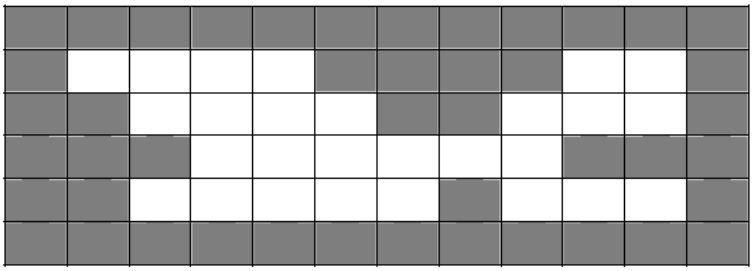

Figure 2.3: Binary image

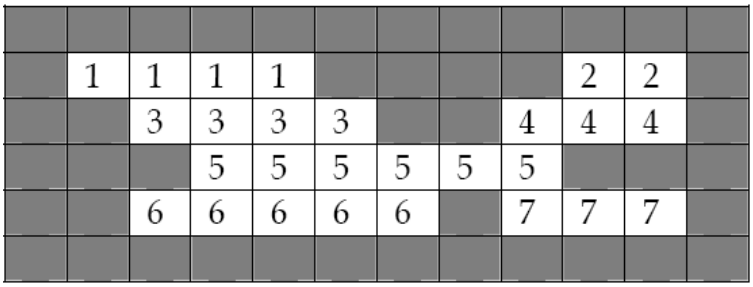

Figure 2.4: Image is labeled in the same row

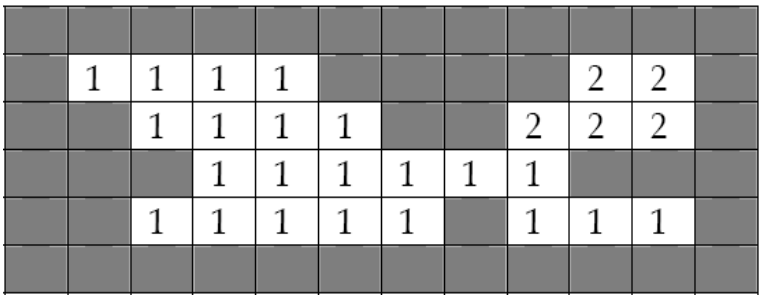

Figure 2.5: Labeled image is scanned from top-left to bottom-right

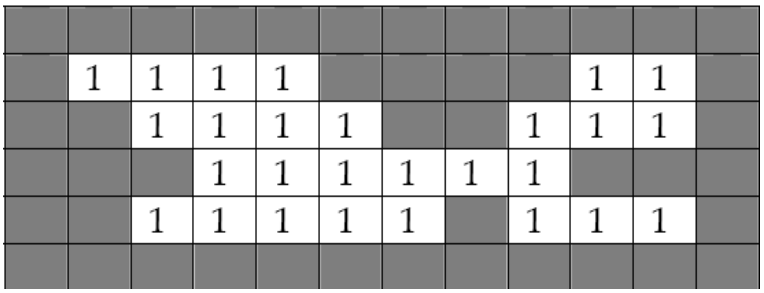

Figure 2.6: Final labeled image

Connected component labeling is performed to label each moving object emerging in the background. The connected component labeling [36] groups the pixels into components based on pixel connectivity (same intensity or gray level). In this article, connected component labeling is done by comparing the pixel with the pixel in four neighbors. If the pixel has at least one neighbor with the same label, this pixel is labeled as same as neighbor's label. The algorithm of connected component labeling algorithm is described as follows:

1. Firstly, image labeling is done on binary image as shown in figure 2.3 where object is shown as 1 (white) and background is shown as 0 (black).
2. The image is scanned from top-left to search the object pixel. The label is done by scanning the image from left to right and comparing label with the neighbor's label in the same line. If the neighbor has the same pixel value, the pixel is labeled as same as previous label as shown in figure 2.4.

3. Next, the labeled image is scanned from top-left to bottom-right by comparing with the four (or eight) neighbors pixel which have already been encountered in the scan (the neighbors (i) to the left of the pixel, (ii) above it, and (iii and iv) the two upper diagonal terms). If the pixel has at least one neighbor, then this pixel is labeled as same as neighbor's label as shown in figure 2.5.
4. On the last scanning, the image is scanned from bottom-right to top-left by comparing with the four neighbors pixel as step 3. The final labeled image is shown in figure 2.6.

The detected objects in the frame are being represented with the help of region-props properties as: centroids (x_{cent}, y_{cent}) and boundingbox (x_{max}, x_{min}, y_{max}, y_{min}), the boundingbox enclosing the object and the centroid representing the center of gravity of the detected object. Each particular centroid or detected object is represented with a particular node in the scene.

2.2 Nearest Neighbour Approach

In pattern recognition, the k-nearest neighbor algorithm (*k*-NN) is a non-parametric method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its k

nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The k-nearest neighbor algorithm is sensitive to the local structure of the data.

2.2.1 Algorithm

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for continuous variables is Euclidean distance.

In Cartesian coordinates, $x = (x_1, x_2,..., x_n)$ and $y = (y_1, y_2,..., y_n)$ are two points in Euclidean n-space, then the distance from x to y, or from y to x is given by :

$$\boldsymbol{d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}} \quad (7)$$

So, as the frame proceeds from current frame to next frame than each object or node is recognized with the help of nearest neighbour classification, in which the minimum distance is being calculated between the nodes in current frame with respect to the next frame. The sequence of the visited nodes by the current node is the output of the

algorithm. The *k*-nearest neighbour algorithm is easy to implement and executes quickly.

2.3 Graph Theoretic Approach

In mathematics and computer science, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices or nodes and lines called edges that connect them. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. Graphs are one of the prime objects of study in discrete mathematics. With Graph theoretic approach the centroids of the objects represents the x & y co-ordinates of the objects position in the scene or frame leading us to our first parameters in terms of position(x, y) of the object in the scene. As all objects are represented with centroids having different x & y co-ordinates according to their position, the relative distances are calculated between each centroid in the scene or frame with respect to the camera leading us the distances between the objects of the scene or frame. A Tensor is prepared which comprises of the position of all the detected objects, relative distances between the objects with respect to camera for each frame, from where these parameter data can be retrieved for further processing. With the help of the frame-rate of video and the computed relative distances another parameter is evaluated as speed, along with the direction of movement of each object the velocity vector is putted as the other parameter in the tensor. With the help of the graph theoretic approach different parameters have been computed for each frame in the video.

Chapter 3

Depth Modelling

Measurement of depth of the whole scenario in terms of the object placements referred to as the depth model. The figure 3.1 depicts the depth modelling of the scenario comprising of objects at different locations or different distances from camera. The model describes the variation in speed of an object with respect to the camera. It explains the non-linearity that the object which is near to a camera it seems to be faster in comparison to an object which is much far from the camera, the far one will appear as slower with respect to camera. In the real scenario the object which is far and another which is near to camera will move at same speed, but with respect to camera the speed varies with distance, that we have proved in the depth model.

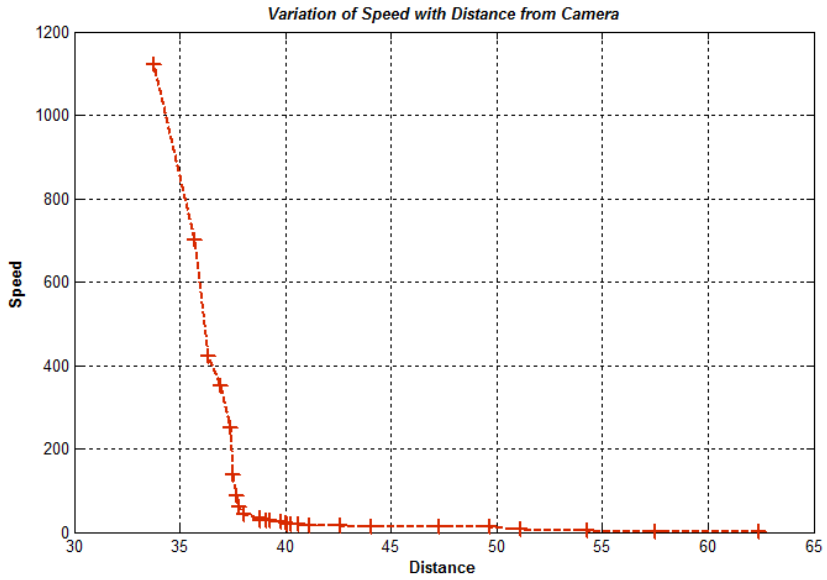

Figure 3.1: Depth Modeling

With this depth model we can have a rough estimate of the object position in that particular scenario, i.e. the slower object will occupy the far position and the fast

object will occupy the least position or near to camera. For proving the depth modelling, we have done an experiment in which we have an object which moves along the x direction in the field of view of camera at different distances from the camera and for each distance we have computed the objects speed with respect to the camera. The figure 3.1 is the representing the plot of the distances versus speed values, depicting non-linearity.

3.1 2D Parameter Estimation

2D Parameters are being referred to as the positions of the objects in each frame of video, relative distances between the objects with respect to camera, the speed and the velocity of individual objects. So the 2D parameters described above are being computed with the help of the graph theoretic approach for each frames objects. 2D plot is being drawn between x co-ordinate and the frame number, showing how the x co-ordinate trajectory of the different detected objects with the consecutive frames is being plotted as shown in figure 3.2, the figure describing the random motion of the different detected objects.

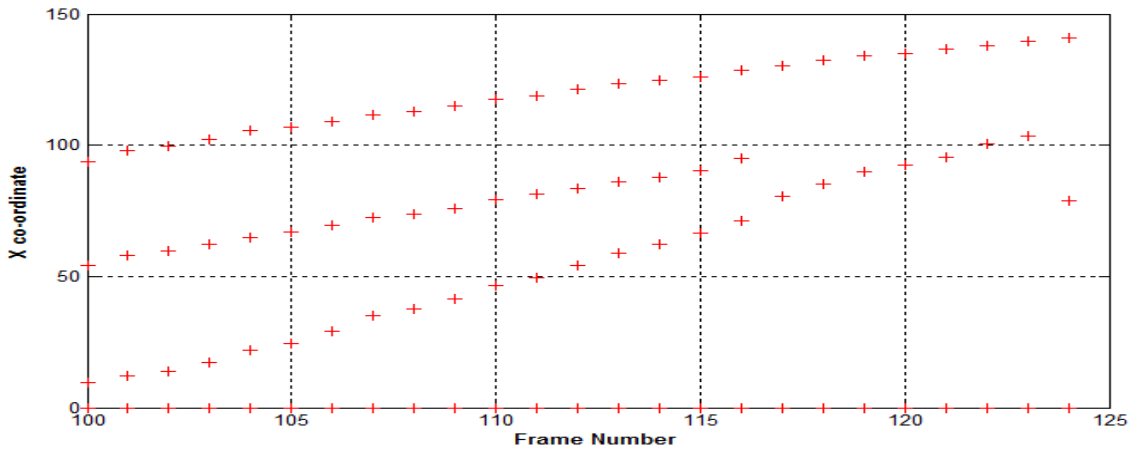

Figure 3.2: Trajectory of different objects through x co-ordinate.

Similarly, 2D plot between y co-ordinate and frame number, showing how the y co-ordinate of the different detected objects varies with the consecutive frames as shown in figure 3.3, figure also describes that representing the detected objects with the center of gravity in the consecutive frames will lead to minimum variation in y c-ordinate with respect to its initial values.

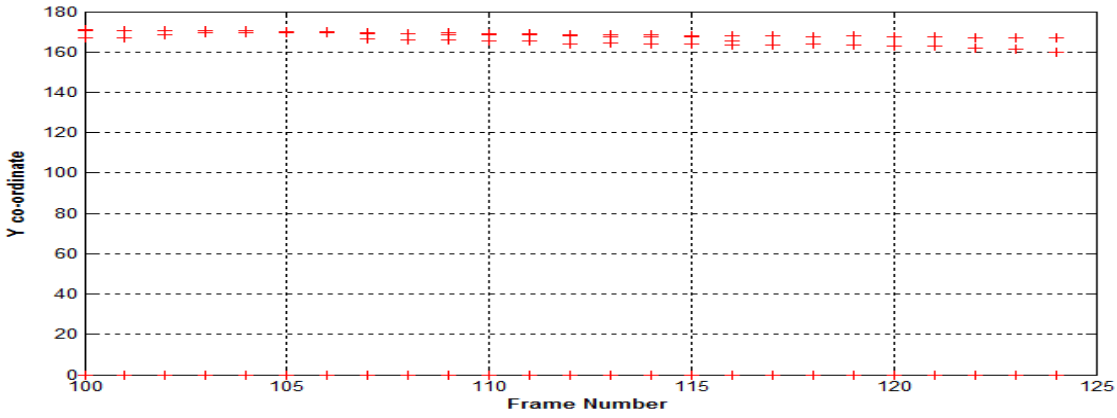

Figure 3.3: Trajectory of ROI through y co-ordinate.

Plotting a combination of x & y co-ordinate trajectories as 2D plot between x & y coordinates shown in figure 3.4, the figure describes real scenario of video comprising of different detected objects motion paths.

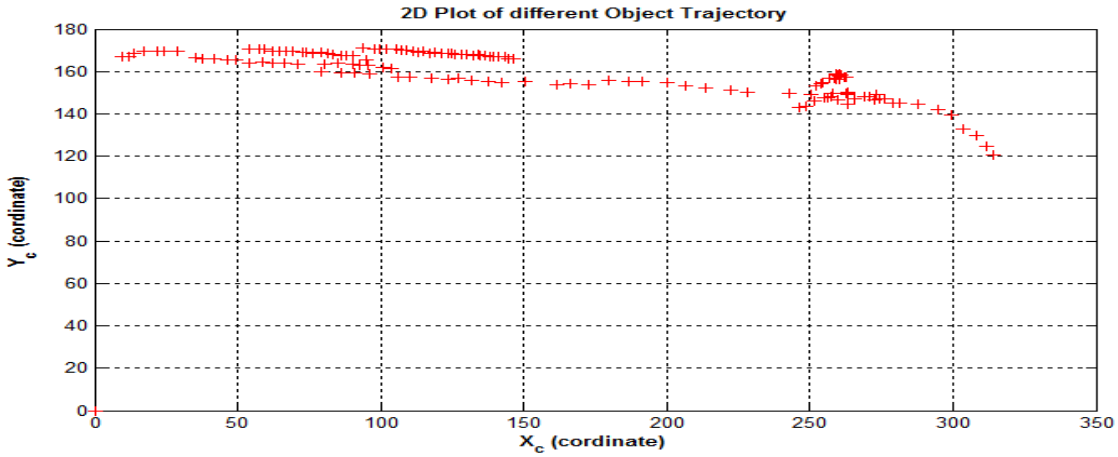

Figure 3.4: 2D Plot of the ROI Trajectory

The disadvantage of this plot is that the different objects paths are being overlapped by some other detected objects path in the same scenario i.e. occlusion occurs and only few objects path can be interpreted correctly. So, to avoid occlusion the 3D reconstruction has to be done of the real scenario with the help of estimation of 3D parameters.

3.2 3D Mapping

With the help of formulated tensor, the 3D plot is drawn between x co-ordinate, y co-ordinate & the relative distances respectively along the 3 axis of the 3D plot as shown in figure 3.5, the figure also describes that the object paths do not occlude with each other as in case of 2D, so occlusion is removed. With the nodes in the 3D plot, each node representing each object comprising of three information in terms of x, y & distance between them. A reconstruction of 3D view of the position of the objects in the scene is formed with the help of monocular vision or 2D view through a single static camera. Now, for each frame a 3D plot is being constructed.

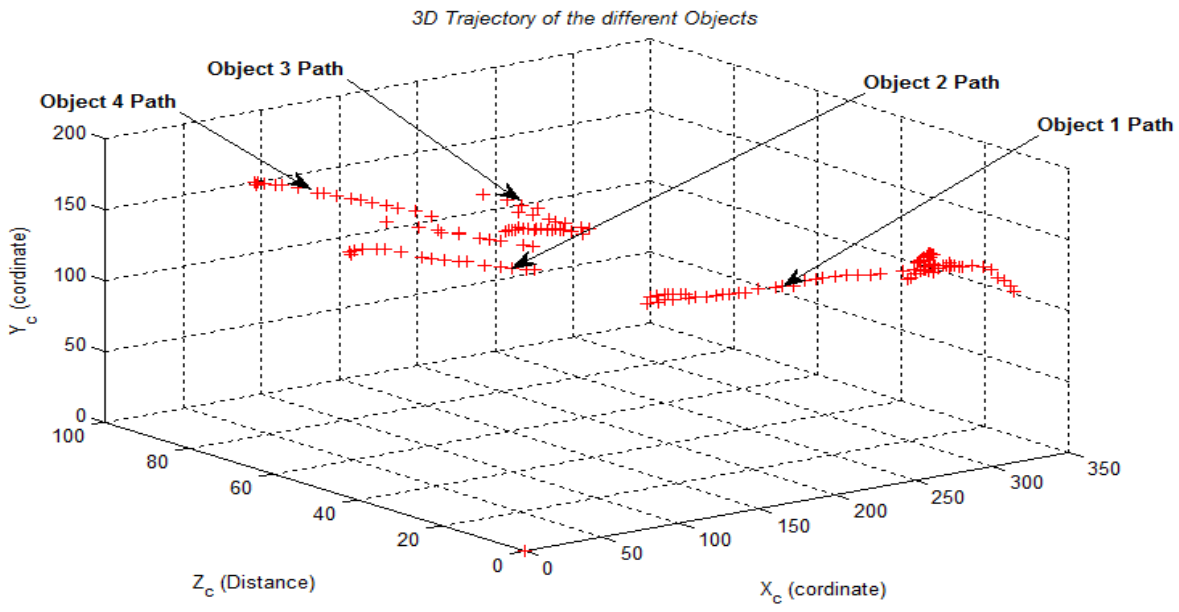

Figure 3.5: 3D mapping of the objects state.

The figure depicts that the frame or the sequence of frames consists of four objects having their own different paths with different positions and speeds with respect to each other, all the paths are clearly differentiable from one another as not in the case of 2D plot. So the 3D mapping is more advantageous over the 2D, as one dimension gets add-on to the 2D view taking it more towards emulating the real human vision.

3.3 Occlusion Handling

Occlusion refers to that, there is something you want to see, but can't due to some property of your sensor setup, or some event and Occlusion handling is refers to exactly how it manifests itself or how to deal with such problem. Occlusion occurs if an object you are tracking is hidden (occluded) by another object. In our system the contribution of the step i.e. conversion of 2D to 3D scenario leads to effectively handle the condition when the occlusion occurs. In the 3D mapping of the 2D view as shown in figure 3.4, the occlusion is removed, no object is being occluded by any other object, each and every object state is being separated by some distances as depicted in the same figure which also adds on to the correctness of our system in terms of better tracking or estimation, also when some object overlaps the another one. So, the occlusion handling is done in a better manner in our system with the help of 3D states of the objects from the monocular vision of the scenario.

3.4 3D Parameter Estimation

The 3D plots corresponding to all frames is being plotted, and with the help of the 3D plots, each plot representing each frame of video, again all the parameters are computed but now these are 3D parameters as associated with 3D maps i.e. 3D position, 3D Distance, 3D velocity & 3D acceleration. Another tensor is being created which will comprise the 3D parameters of each frame for further processing or estimation.

Chapter 4

Estimation Algorithms

4.1 Hidden Markov Model

A Hidden Markov Model (HMM) is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. In simpler Markov models, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

The formal definition of a HMM is as follows:

$$\lambda = (A, B, \pi) \tag{8}$$

S is our state alphabet set, and V is the observation alphabet set:

$$S = (s_1, s_2, \cdots, s_N) \tag{9}$$

$$V = (v_1, v_2, \cdots, v_M) \tag{10}$$

We define Q to be a fixed state sequence of length T, and corresponding observations O:

$$Q = q_1, q_2, \cdots, q_T \tag{11}$$

$$O = o_1, o_2, \cdots, o_T \tag{12}$$

A is a transition array, storing the probability of state j following state i. Note the state transition probabilities are independent of time:

$$\mathbf{A} = [a_{ij}],\ a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i) \tag{13}$$

B is the observation array, storing the probability of observation k being produced from the state j, independent of t:

$$\mathbf{B} = [b_i(k)],\ b_i(k) = P(x_t = v_k \mid q_t = s_i) \tag{14}$$

$\boldsymbol{\pi}$ is the initial probability array:

$$\boldsymbol{\pi} = [\pi_i],\ \pi_i = P(q_1 = s_i) \tag{15}$$

In our system for both the categories, separately we have modelled three HMM's for all the 3D parameters i.e. HMM1 for position, HMM2 for velocity & HMM3 for acceleration and then a final decision is made in terms of the ROI state estimation.

HMM1:

States: Forward, Backward, Unchanged, Front, Back

Observations: Increment, Decrement, Unvaried

HMM2:

States: Horizontal Velocity, Vertical Velocity

Observations: +x, -x, +y, -y

HMM3:

States: Positive Acc., Constant Acc., Negative Acc.

Observations: Fast, No-Variation, Slow

For all the three HMM's in each category the state transition matrix and the state emission matrix comprising of probabilities is defined for all the states and the observations in the model according to the possible output tokens. By taking the appropriate length of the frames, the HMM is trained and then the maximum likelihood estimate of the transition and emission probabilities is calculated after which the most probable state path is estimated. Probabilistic output is generated in terms of the states by the HMM.

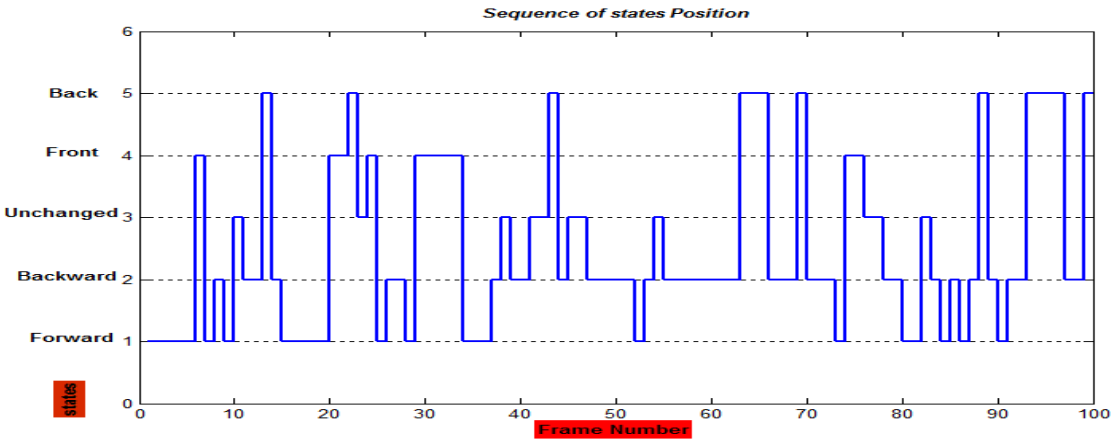

Figure 4.1: HMM for position without background consideration

The figure 4.1 & 4.2 are representing the probabilistic output of HMM1 (Position) in terms of sequence of states without background consideration and with background consideration respectively, the state varies between the five states as forward, backward, unchanged, front & back in terms of position of the object or ROI, where the forward state will lead object to move in positive x direction, the backward state will lead object to move in negative x direction, Unchanged state will lead to no motion or no change in object state, Front state will lead to object motion towards the camera & Back state will lead to movement of object away from the camera in opposite direction. Estimation of the states by HMM1 for the future frames will describe the trajectory of the object along with the direction. From the figure 4.1, i.e.

first category (without background consideration) we can interpret that most variation in the position of the object is in backward state, than in forward state, then in back state, than in unchanged state and the least is in front state.

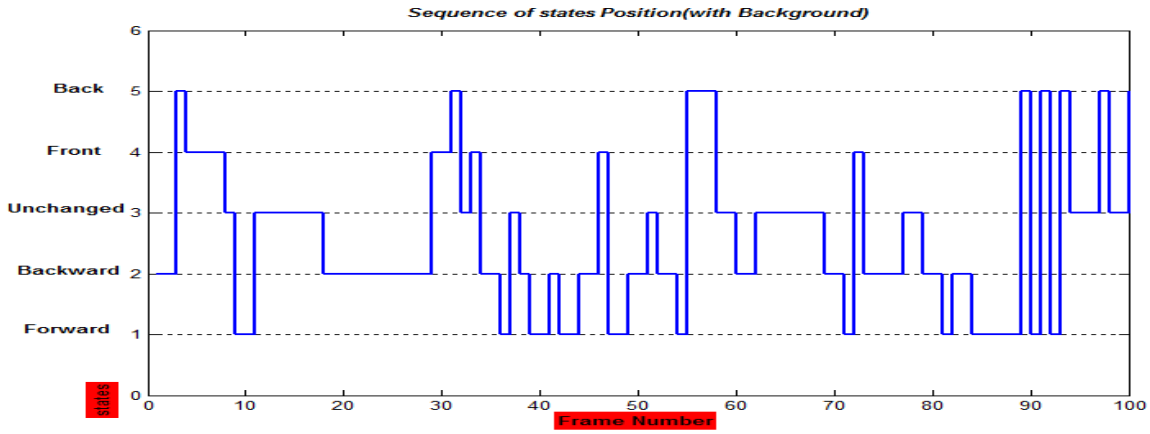

Figure 4.2: HMM for position with background consideration

Similarly from the figure 4.2, i.e. second category (with background consideration) we can state, interpret that most variation in the position of the object is in backward state, than in Unchanged state, followed by forward back state and the least is the front state.

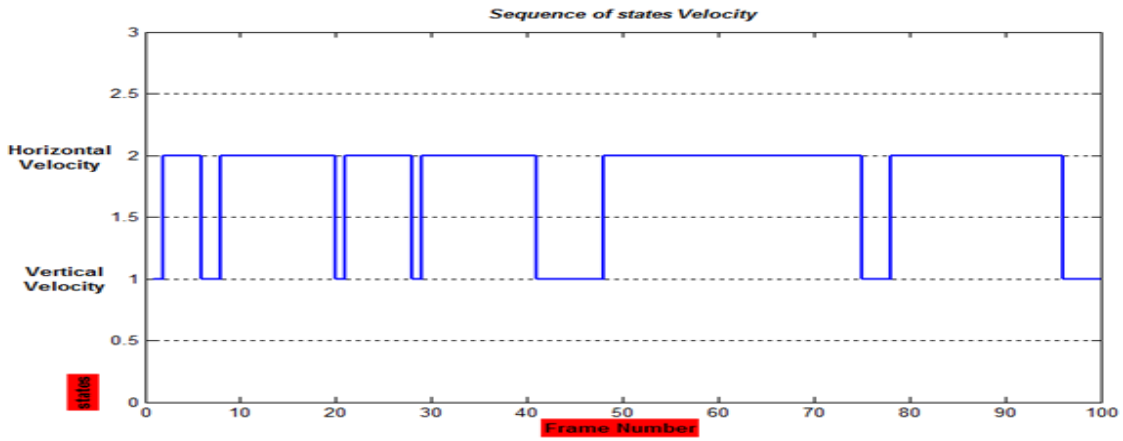

Figure 4.3: HMM for velocity without background consideration

The figure 4.3 is representing the probabilistic output of HMM2 (Velocity) in terms of sequence of states as horizontal velocity & vertical velocity in terms of the velocity vector of the object or ROI, where the horizontal velocity state will lead to motion of object horizontally i.e. either in +x or –x direction, deciding factor will be through HMM1, Similarly with the help of HMM1 the vertical velocity state will lead to motion in either +y or –y direction. More the same number of states in the consecutive frames will lead to increase in velocity in that particular direction. Interpretation from figure 4.3, i.e. for first category can be done as the object has more of horizontal movement compared to the vertical movement and for the second category i.e. figure 4.4, same interpretation can be done for velocity of the object.

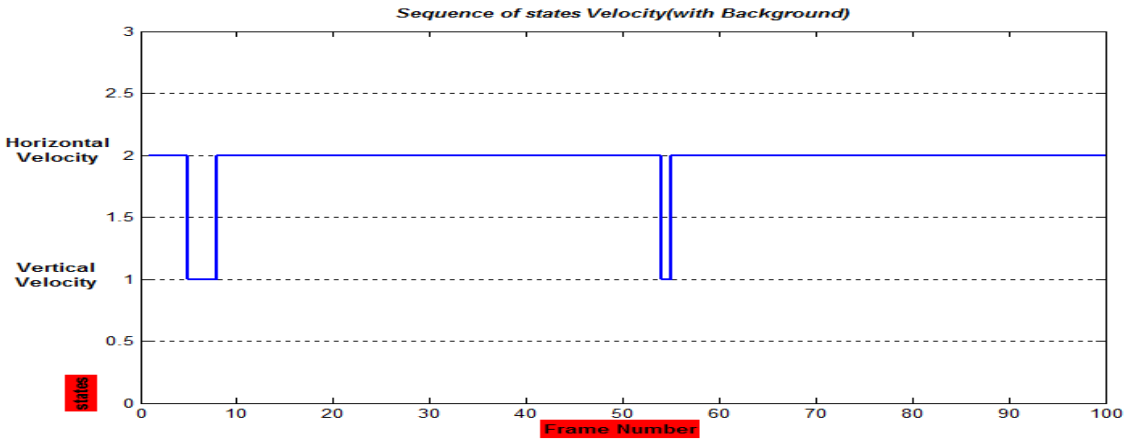

Figure 4.4: HMM for velocity with background consideration

The figure 4.5 & 4.6 are representing the probabilistic output of HMM3 (Acceleration) in terms of sequence of states without background consideration and with background consideration respectively, the state varies between the three states as positive acceleration, constant Acceleration & negative acceleration in terms of acceleration of the object or ROI. The three states where the positive acceleration is referred to as the increase in velocity, constant acceleration leads to constant velocity

& the negative acceleration leads to the decrease in the velocity of the object with respect to the sequence of frames in the video.

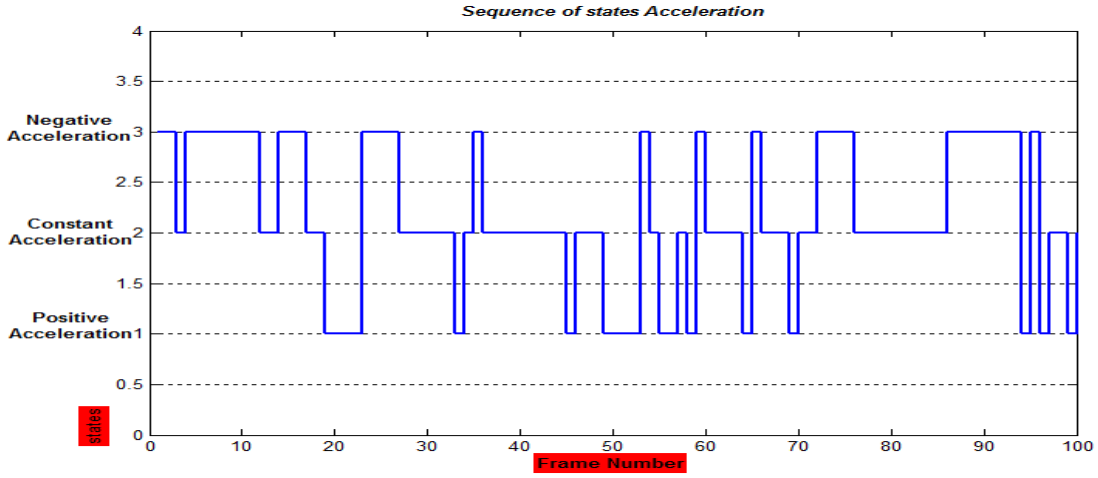

Figure 4.5: HMM for acceleration without background consideration

As interpreted from figure 4.5, there is mainly constant acceleration in the objects state followed by some positive acceleration in the first category. Whereas in the case of second category, objects state is evenly distributed between the positive & negative acceleration for the given frame sequence with ups & downs in velocity of object.

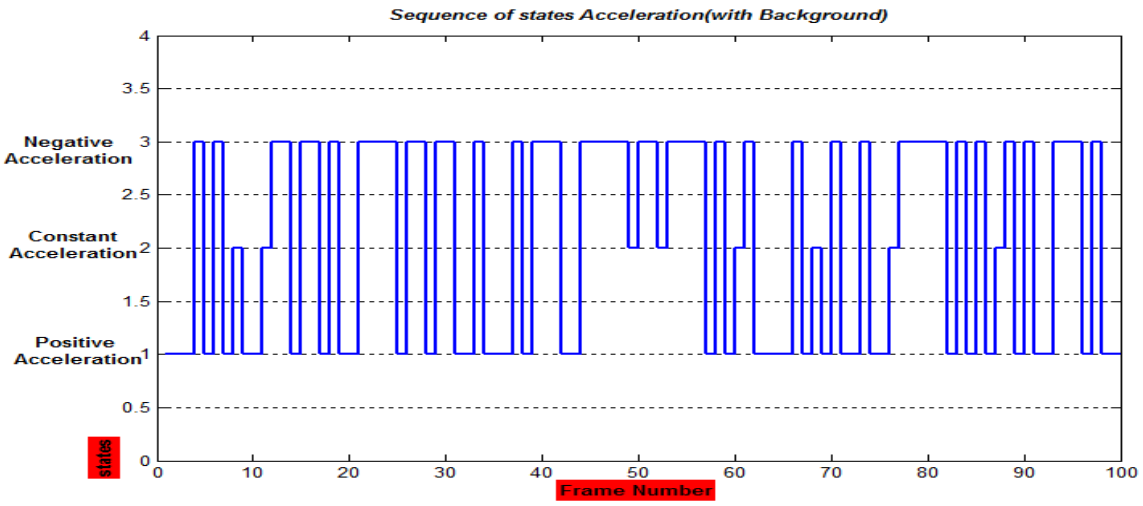

Figure 4.6: HMM for acceleration with background consideration

So, with the three HMM's taking the 3D parameters as their input tokens for the probabilistic framework, the state estimation is done for both the categories and the final state comprising of object position, velocity & acceleration is formalized.

4.2 Fuzzy Logic

From the three algorithms for predicting objects state, the first HMM is covered in the above section, in which a group of 3 HMM, one for each 3D estimated parameter is modelled and a cumulative decision is taken with respect to objects state, now taking the second algorithm of our system as the fuzzy logic, mainly comprising of the FIS.

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions.

Using fuzzy logic makes possible to change only two binary logic states "element belongs to the set" and "element does not belong to the set" to the gradual transition between these two states. Fuzzy inference system requires the definition of a knowledge base, e.g. implicative rules. These rules allow the use of linguistic variables [13] as IF-THEN rule. In our system as shown in figure 4.7, the observed data is taken in form of input & output variables, which constitute a set of observed values of the five input variables (position_x, position_y, position_z velocity, velocity & acceleration) and the corresponding output variable representing the

observed output as plotted in figure 4.8 & 4.9 respectively. Clustering of data based on the relationship between input and output variables is done.it will lead to us the natural grouping in data from large data sets.

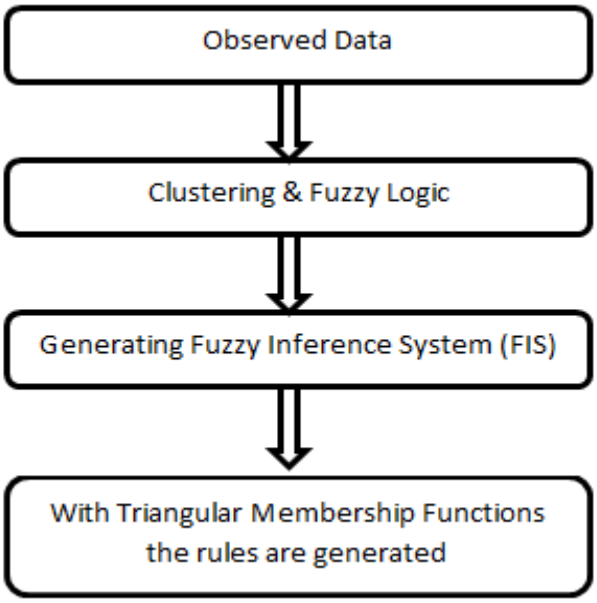

Figure 4.7: Fuzzy Logic System

Fuzzy logic will be employed to capture the broad categories identified during clustering into a Fuzzy Inference System (FIS). The fuzzyfication of each input variable must be done. Fuzzyfication means assigning of sharp numerical values to fuzzy sets with membership functions (Triangular). Fuzzy sets are based on linguistic expressions. After the fuzzy inference system completes its work, i.e. derives result, the fuzzy rules are generated and viewed for the prediction of the object's state.

Plotting input variables with respect to the different samples of frame as shown in figure 4.8, describes the variation in the observed state of the object in terms of its position, velocity & acceleration with their appropriate magnitudes.

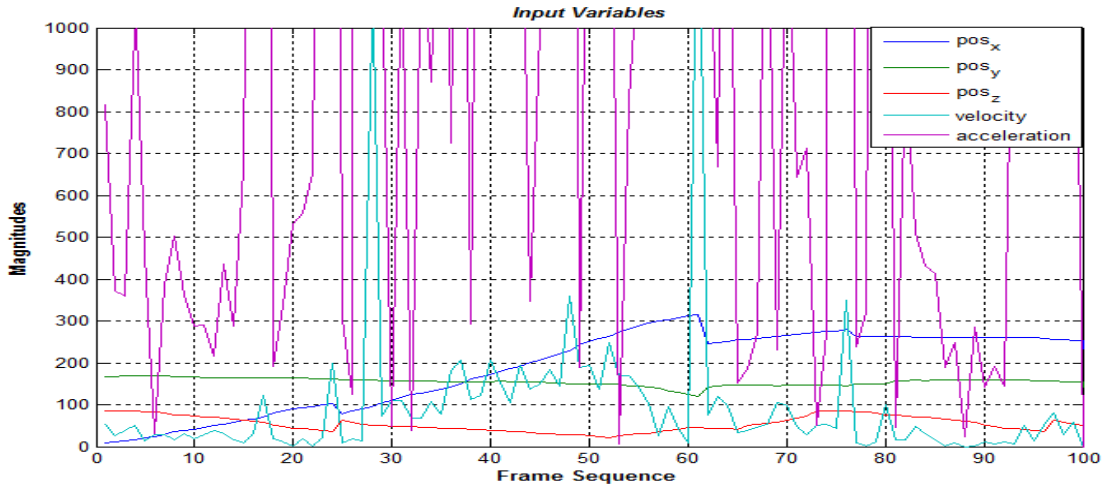

Figure 4.8: Plot of Input Variables

The acceleration values were high in magnitude compared to the position magnitude of object, therefore the acceleration index going out of bound in above plot of input variables. Similarly in the figure 4.9, the observed output in terms of the new position of the object is plotted with the different samples of frame with respect to the input variables.

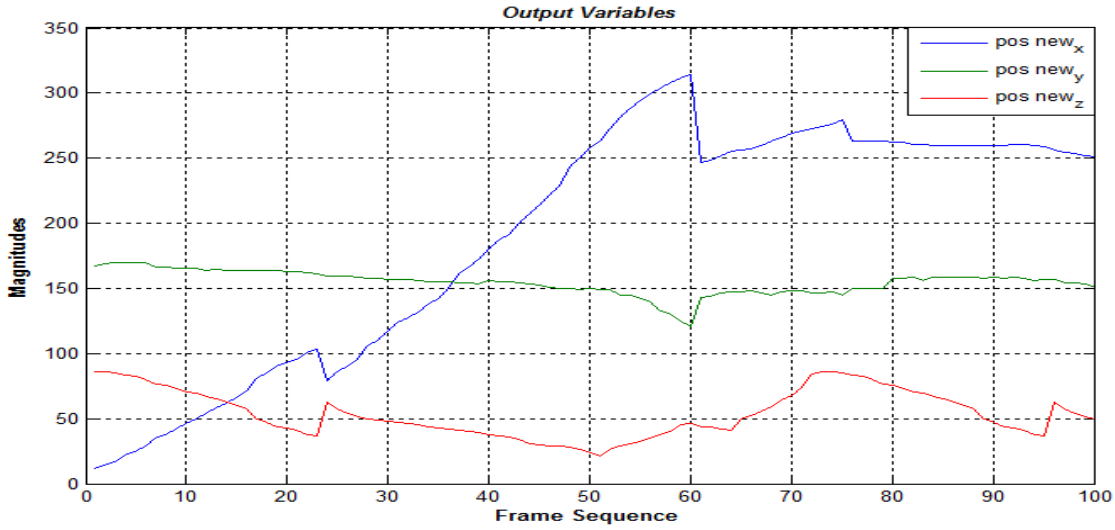

Figure 4.9: Plot of Output Variables

The FIS parameters used in our system can be interpreted from the table 4.1, which leads to different parameters as functions with their shape and number of input &

output Membership Functions (MF) along with the number of fuzzy rules generated. The five input & three output membership function used in our system is of triangular type with a total of 24 fuzzy rules for the prediction of position of the object in the future frames.

All parameters of FIS are summarized in the Table I.

Parameter	Value
Shape of position_x input MF	Triangular
Count of position_x input MF	4
Shape of position_y input MF	Triangular
Count of position_y input MF	4
Shape of position_z input MF	Triangular
Count of position_z input MF	4
Shape of velocity input MF	Triangular
Count of velocity input MF	3
Shape of acceleration input MF	Triangular
Count of acceleration input MF	3
Shape of newposition_x output MF	Triangular
Count of newposition_x output MF	4
Shape of newposition_y output MF	Triangular
Count of newposition_y output MF	4
Shape of newposition_z output MF	Triangular
Count of newposition_z output MF	4
Total count of fuzzy rules	24

Table 4.1: FIS Parameters

Fuzzy rules generated from the system in figure 4.7, can be described as follows in table 4.2 assuming the fuzzy sets include the following:

- Large Negative: LN
- Small Negative: SN

- Nil: N
- Small Positive: SP
- Large Positive: LP

Rules	x_t	y_t	z_t	**Velocity**	**Acceleration**	x_{t+1}	y_{t+1}	z_{t+1}
1	SN	LN	SN	SP	SP	N	SN	SN
2	N	SP	SN	SN	LN	SP	SP	N
3	LP	N	N	SP	SP	LP	SP	SP

Table 4.2: Few developed fuzzy rules of the system.

In table 4.2, the x, y, & z has fuzzy sets in terms of the position of object in the3D space where Nil representing the centre of the axis, LN as extreme origin, & LP as end of the axis & vice-versa and for velocity & acceleration the fuzzy sets represents the Nil as no motion, LN as large decrease in velocity & acceleration respectively & vice-versa. Taking rule 1, when the position in present frame is, near to centre on x & z axis and near to origin on y axis and small increase in velocity & acceleration then in the next frame the new position will be, on the centre of x axis and it will be near to centre on y & z axis.

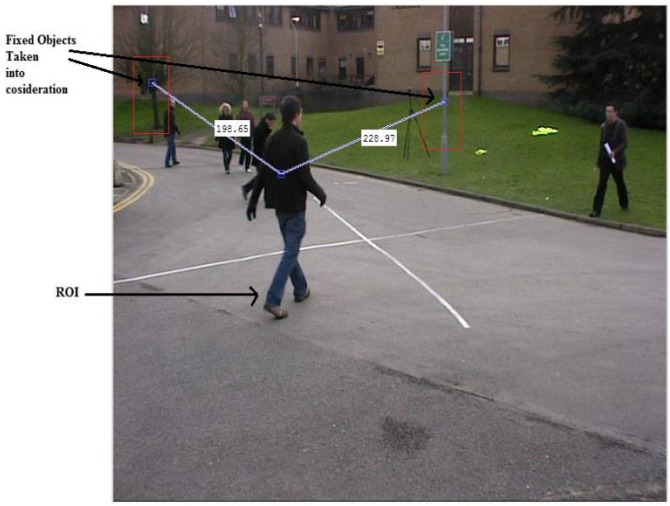

Figure 4.10: Dataset Image consisting of the ROI & the reference points of the background along with the calculated parameter.

In the second category of our system, when the background is taken into consideration, in which certain fixed objects in the background are taken as reference points to locate the moving object or ROI, as described in figure 4.10. The datasets figure 4.10 & 4.11 is taken from PETS, in which a tree and a pole is referred to as the reference objects or points from where the parameters such as, distances, velocity, etc. is calculated with respect to the reference points. With these parameters the observation is done and the remaining same process is performed as shown in the figure 4.7, now considering the background. The new position of the object or ROI is estimated with help of the fuzzy rules generated at the end of the process through the different FIS parameters.

Figure 4.11: Frame number 10 to 120 in multiple of 10 respectively from left to right, showing the variation in ROI position with respect to the reference points.

The predicted position is being plotted in the figure 4.12 in terms of parameters with respect to reference points. When considering the background in our system will lead

to estimate the parameters from the reference points which act as input FIS parameters and the output FIS parameters as the predicted new position of the objects state. With the input parameters a searching mechanism is developed for the current sample of frame for finding the position of object, referring to the previous sample distances & velocity of the object relative to the fixed object points in the scenario.

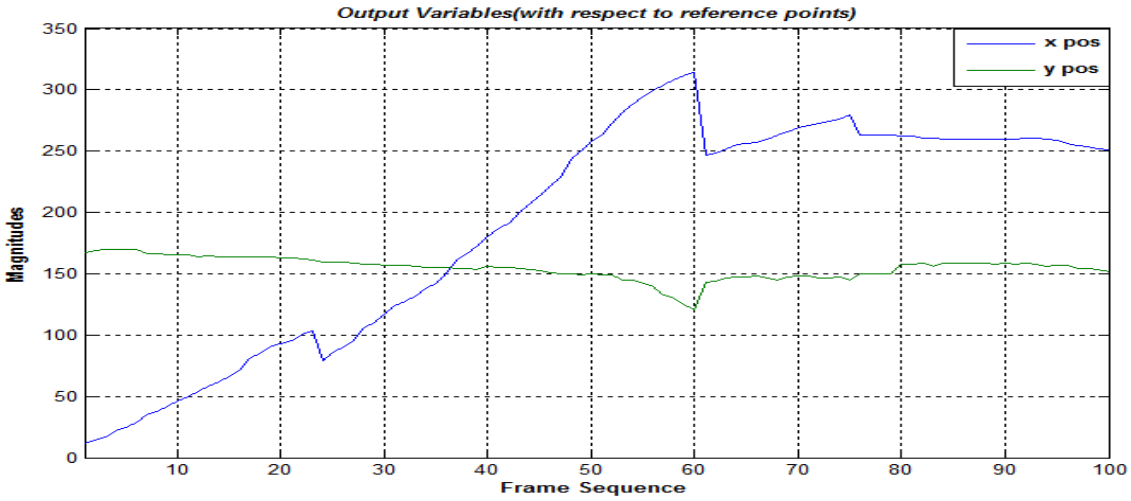

Figure 4.12: Plot of the predicted state of the ROI from the fuzzy rules with respect to reference points.

From the searching mechanism of the objects new position with the help of the distances and velocity of the previous frame relative to the fixed objects acting as reference points, the new distances for the current frame sample is evaluated from the observations made up to previous frame, after getting the distances from fixed reference points the object position is marked as the new position in the current frame, taking the velocity relative to fixed reference points also in consideration. As shown in figure 4.10, two reference points, first as a tree & second as a pole is referred to as the fixed points in the background. The figure also explains that with the motion of the object which is to be tracked, in any direction, it will lead to change in the distances. Suppose, the object is moving in the right direction than the distance from the first reference point will be more as compared to the distance from the second reference

point and similarly, if object moves towards the extreme left than the distance from first point will be less with respect to the second fixed point. The distance varies with the variation in the position of object which is being tracked. So, with the observation of the distances from the previous frame, the prediction can be done for the new position of object in current frame relative to the estimated distances for current frame.

4.3 Support Vector Machine

For the two categories of our system the prediction of the objects state is done, now moving on to the last algorithm of our system for the prediction of object state is SVM. In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyse data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks.

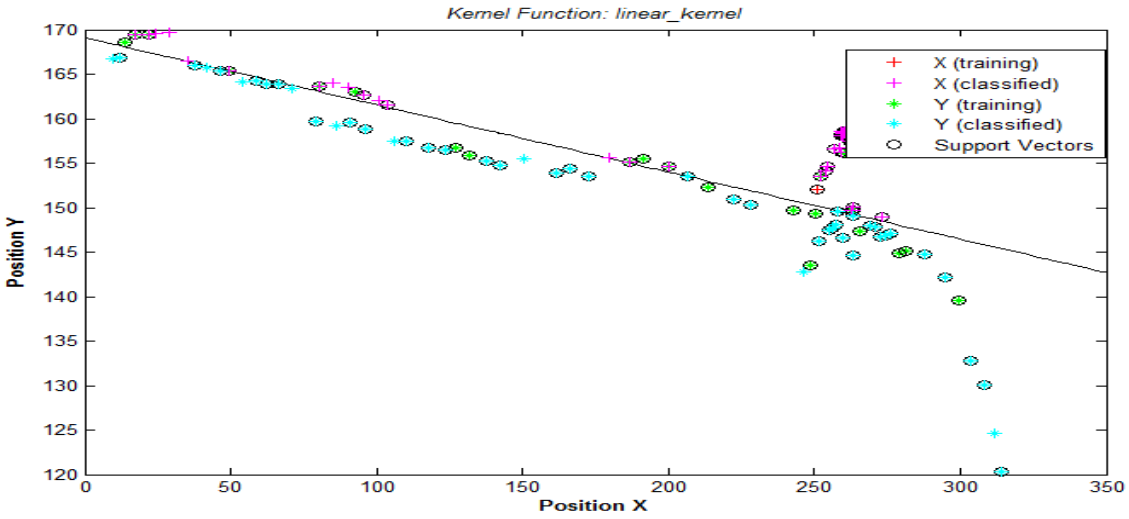

Figure 4.13: SVM Classification without considering the background reference points.

Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class called functional margin. The process for classifying the parameter data involves, the sample data from the dataset is loaded, groups are classified, training and testing sets are created, SVM classifier is trained using a linear kernel function, then grouped data is plotted, for classification of the test data, test set is also classified using the hard margin SVM classifier the hard classification is done, and then the classifier performance is noted.

In the first category, the classifier plots by clustering the parameter in terms of the position of monocular vision separated by a hard decision, and after the classifier training the predicted states of the ROI is classified as depicted in figure 4.13.

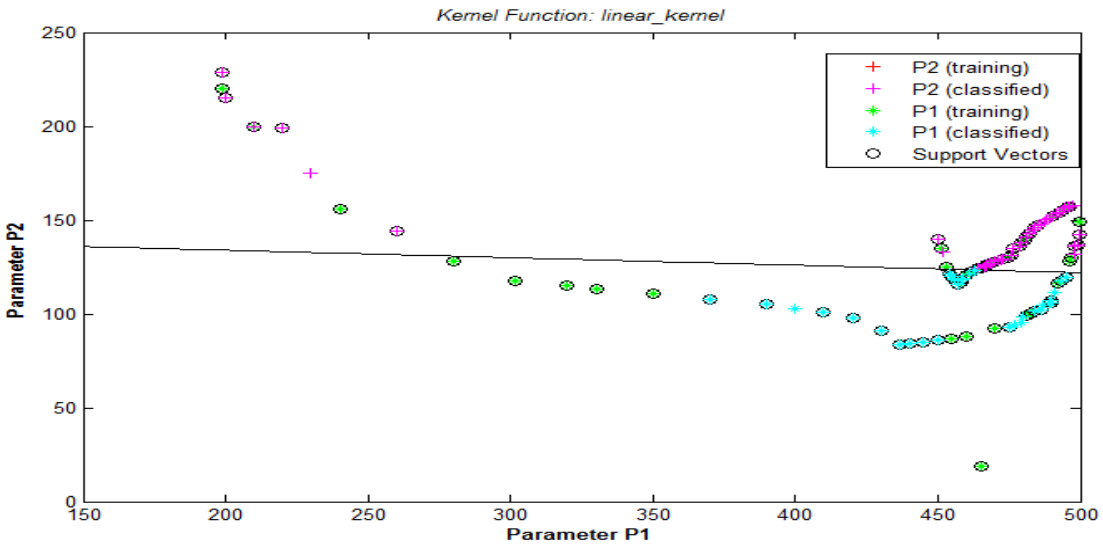

Figure 4.14: SVM Classification with the consideration of the background reference points.

In the second category, the classifier plots the clusters of parameters (distances p1 & p2) with respect to the reference points in the background and after the training of classifier; the classification is done for the predicted parameters with respect to the reference points as depicted in figure 4.14, two parameters separated by hyperplane. In the figure, where p1 denoting the distance from reference point first & p2 denoting the

distance from reference point second, with SVM both classes of distances are separated than new distances are classified for prediction of new position of object in the current frame.

Chapter 5
Data Fusion

To overcome the drawbacks of Dempster's fusion rule and in the meantime extend the domain of application of the belief functions, a new mathematical framework is proposed, called Dezert-Smarandache Theory (DSmT) with a new set of combination rules, among them the Proportional Conflict Redistribution no. 6 which proposes a sophisticated and efficient solution for information fusion. DSmT deals with uncertain, imprecise and high conflicting information for static and dynamic fusion as well [3, 4, and 6].

5.1 DSmT Model

Let the frame of discernment be $\Theta = \{\theta_1, \theta_2, ..., \theta n\}$ with exhaustive elements $\theta_{i.}$ This model is called as free model denoted as $M^f(\Theta)$, when no assumptions are made about the hypotheses i except for the exhaustiveness. The model does not incorporates most real-life problems since some combinations of hypotheses are time dependent or are not valid anymore when more knowledge is available, therefore a hybrid model, denoted M, can be constructed to deal with these integrity constraints. In DSmT and without additional assumption on Θ but the exhaustivity of its elements, the hyper-power set are defined, i.e. Dedekind lattice. Hyper-power set, D^{Θ}, is defined as the set of all composite propositions built from elements of Θ with intersection and union operators such that:

1. $\emptyset, \theta_1, \theta_2, \ldots, \theta n \in D^{\Theta}$
2. If A, B$\in$D$^{\Theta}$ then A, B$\in$D$^{\Theta}$ and A, B$\in$D$^{\Theta}$

No other elements belong to D$^{\Theta}$ except those obtained using rules 1 and 2.

From a frame Θ, we define a (general) basic belief assignment (bba) as a mapping m(.):D$^{\Theta}\rightarrow$[0,1] associated to a given source of evidence as:

$$\boldsymbol{m(\emptyset) = 0 \;\&\; \sum_{X \in D} m(X) = 1}$$

The quantity m(X) is called the generalised basic belief assignment (gbba) of X, also called the generalised mass of X.

5.2 Combination Rules in DSmT

The classic DSmT rule of combination holds when the model is free. When k independent sources give their belief masses according to $m_1(.)\ldots..m_k(.)$, the combination rule for $\forall$ X$\in$ D$^{\Theta}$ is given in equation 16.

Using the classic rule of combination in real-life fusion problems i.e. integrity constraints must be taken into account to impose assumptions about the model. For such cases, the hybrid rule of combination for k independent sources with belief assignments $m_1(.)\ldots..m_k(.)$ is defined for $\forall$ X$\in$ D$^{\Theta}$ in equation 17.

$$\boldsymbol{m_c^f(X) = \sum_{\substack{Y_1 \ldots .Y_k \in D^{\vartheta} \\ Y_1 \cap \ldots . \cap Y_k \in X}} \prod_{i=1}^{k} m_i(Y_i)} \quad ..(16)$$

$$\boldsymbol{m_c^{DSmH}(X) = \emptyset(X) \cdot [S_1(X) + S_2(X) + S_3(X)]} \quad ..(17)$$

$\emptyset$(X) is the characteristic non-emptiness function of a set X, i.e. $\emptyset$(X)=1 if X $\notin\emptyset$ and $\emptyset$(X) =0. $\emptyset_M$ is set of all elements of D^{Θ} which are empty through the constraints of the model M and classical empty set is $\emptyset$. S_1, S_2, S_3 are defined as,

$$S_1(X) = \sum_{\substack{Y_1 \ldots Y_k \in D^\vartheta \\ Y_1 \cap \ldots \cap Y_k \in X}} \prod_{i=1}^{k} m_i(Y_i)$$

$$S_2(X) = \sum_{\substack{Y_1 \ldots Y_k \in \emptyset \\ [U=X] \vee [(U \in \emptyset) \wedge (X = I_t)]}} \prod_{i=1}^{k} m_i(Y_i)$$

$$S_3(X) = \sum_{\substack{Y_1 \ldots Y_k \in D^\vartheta \\ Y_1 \cap \ldots \cap Y_k \in X \\ Y_1 \cup \ldots \cup Y_k \in \emptyset}} \prod_{i=1}^{k} m_i(Y_i)$$

With, $U = u(Y_1) \cup \ldots\ldots \cup U(Y_k)$, $u(Y)$ is the union of all θ_t that compose Y and I_t is the union of all elements in Θ, or total ignorance. There are many combination rules within DSmT, but we chose PCR6 as our combination rule to avoid transferring masses to relative ignorance and for a more intuitive and expected result when combination is done for more than two sources.

5.3 PCR 6

Proportional Conflict Redistribution number 6(PCR6) [12] is an alternative of PCR5 for the general case when the number of sources to combine become greater than two (i.e. s > 3). PCR6 gets better intuitive results as compared to PCR5 and it also does not follow back on the track of conjunctive rule as PCR5 does. The general idea behind the PCR6 rule is to transfer the conflicting masses to the non-empty elements that are involved in the conflict as opposed to transfer it to relative ignorance, which is the case of hybrid DSmT. PCR rules can be found in [7]. Equation 18, defines the rule for k independent information sources as follows.

$$m_c^{PCR6}(X) = m_c^f(X) + \sum_{i=1}^{k} F_i m_i(X)^2 \quad (18)$$

where,

$$F_i = \sum_{c_1 c_2} \left(\frac{\prod_{l=1}^{k-1} m_{\sigma_i(l)} \left(Y_{\sigma_i(l)} \right)}{m_i(X) + \sum_{l=1}^{k-1} m_{\sigma_i(l)} \left(Y_{\sigma_i(l)} \right)} \right)$$

$$c_1 = \bigcup_{j=1}^{k-1} Y_{\sigma_i(j)} \cap X \in \emptyset$$

$$c_2 = \left(Y_{\sigma_i(1)} \ldots\ldots Y_{\sigma_i(k-1)} \right) \in \left(D^{\vartheta} \right)^{k-1}$$

$$\sigma_i(l) \rightarrow \begin{cases} \sigma_i(l) = l & l < i \\ \sigma_i(l) = l + 1 & l \geq i \end{cases}$$

5.4 Problem Formulation

The Evaluation steps are represented in the figure 5.1, and are discussed as follows:

I. Defining the group of Experts:

The group of experts of the system are the three classifiers i.e. Hidden Markov Model as $m_1(.)$, Fuzzy Logic as $m_2(.)$ and Support Vector Machine as $m_3(.)$, their prediction results acts as an evidence for solving the problem which has been formulated. The three experts are defined as:

$$\mathbf{E = \{ m_1(.), m_2(.), m_3(.) \}}$$

II. Defining the frame of discernment Θ:

The frame of discernment is defined as:

$$\mathbf{\Theta = \{A, B\}}$$

Here, A means Tracking with Background Consideration and B means Tracking without Background Consideration. So we can get its hyper-power set as:

$$\mathbf{D^{\Theta} = \{\emptyset, A \cap B, A, B, A \cup B\}}$$

Then the generalized basic belief assignment/mass (gbba) is constructed as follows: **m(A)** is defined as the gbba for Tracking with Background Consideration, **m(B)** is defined as the gbba for Tracking without Background Consideration, **m(A∩B)** is defined as the gbba for the conflicting mass, **m(A∪B)** is defined as the gbba for unknown mass due to the restriction of knowledge and technology.

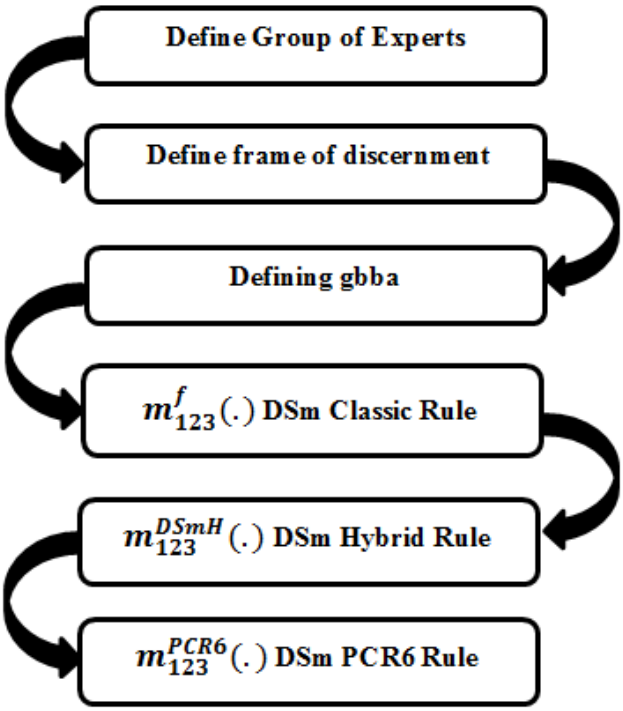

Figure 5.1: Evaluation Steps of DSmT

III. Defining the generalized basic belief assignment/mass (gbba):

Based on the probabilistic approach of how correct the classifier is for a particular frame, the belief assignment/mass the gbba of a set of mapping m(.) is constructed.

IV. Generating the classic DSm rule of Combination:

Considering the free model of DSmT, the classic DSm rule of combination as defined in equation 1 and is followed to find out the masses which are as depicted in Table 5.1.

V. Generating the Hybrid DSm rule of Combination:

Hybrid DSm (DSmH) rule of combination for three or more independent sources of information is defined in equation 17 and is followed for fusion and the masses are evaluated as depicted in table 5.1.

VI. Generating the DSm Proportional Conflict Redistribution no. 6 rule of Combination:

For number of evidences less than three, the DSm PCR5 rule is used for fusion or combination, but as our problem considers three sources of evidence as HMM, Fuzzy Logic, & SVM, we have used DSm PCR6 rule as expressed in equation 18, for combination and taking the final decision of our system. The decision is taken after the fusion incorporated by rule PCR6, than taking into consideration the highest belief assignment/mass of the system as the best Tracking Method with respect to the three classifiers.

	A	**B**	**A∪B**	**A∩B**
$\boldsymbol{m_1(.)}$	0.6	0.3	0.1	0
$\boldsymbol{m_2(.)}$	0.4	0.3	0.3	0
$\boldsymbol{m_3(.)}$	0.4	0.4	0.2	0
$\boldsymbol{m^f_{123}(.)}$	0.288	0.138	0.006	0.298
$\boldsymbol{m^{DSmH}_{123}(.)}$	0.288	0.138	0.304	0
$\boldsymbol{m^{PCR6}_{123}(.)}$	0.353	0.195	0.006	0

Table 5.1: Generalised basic belief assignments of three experts with the results of the three combination rules.

As in our system we calculated the gbba's for the two tracking approaches one with considering background & another without considering the background from all the three algorithms probabilistically which are as depicted in Table 5.1, first

the classic rule is computed followed by the hybrid rule after distributing the conflicting mass and the lastly, the DSm PCR6 rule is evaluated, which leads us to beliefs for both the categories of our system, with a higher belief to the category considering the background for tracking and we got the result as the Tracking is better when the Background is taken into consideration instead of Tracking without Background consideration.

Chapter 6

Results

With the help of the DSmT PCR6 it is proven that tracking is better when the background of the scenario is taken into consideration rather than background subtraction as it is done mainly in the conventional tracking methods.

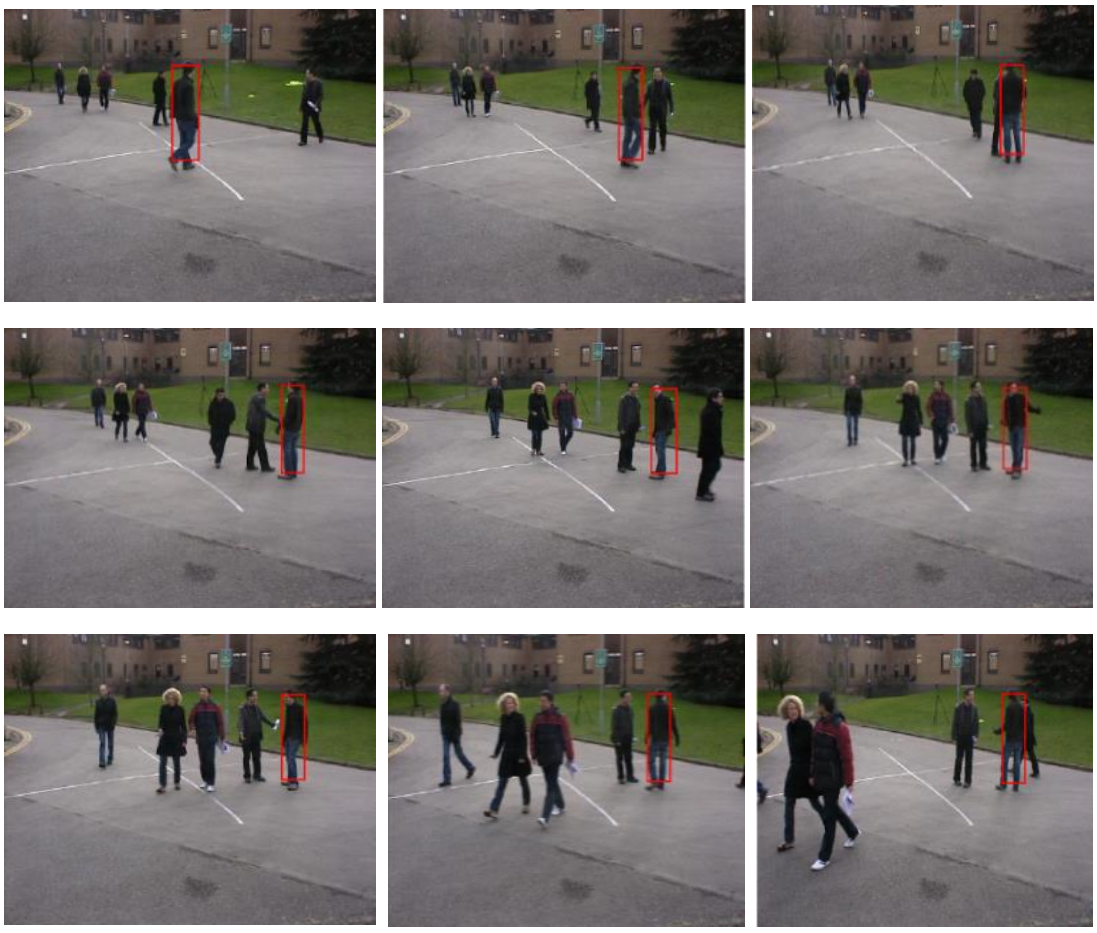

Figure 6.1: Frame number 20 to 100 in multiple of 10 from left to right respectively, showing the tracking of an object with the consideration of the background reference points.

Figure 6.2: Frame number 10 to 100 in multiple of 10 from left to right respectively, showing the tracking of an object without the consideration of the background reference points.

Figure 6.1, depicting the tracked object in various samples of frame along the time, with tracking relatively with the reference objects in the background, the reference objects can be fixed or movable; in our case it is fixed, where the tracking is better or we could say nearly perfect whereas in the other category i.e. without considering the background, it suffers in terms of imperfect tracking as depicted in figure 6.2, in which the frame number 40, 90 & 100 are showing the wrongly predicted state of the object means in every sample of ten frames nearly three will have the wrongly predicted states reducing the overall correctness of the system which are not considering the background of the scenario. So increasing the overall correctness of the system in terms of tracking will mainly depend on the consideration or rejection of

the background of the scenario, which is proven by our system in the DSmT section. In our system we are considering it i.e. utilizing the rule of science of locating any object in any scenario relatively with respect to some other object in the same scenario, which also has been appreciated by our proposed system.

Chapter 7
Conclusion

7.1 Comparison with other Estimating Techniques

To evaluate our method and compare with other tracking algorithms, we have used the public video database PETS [16]. Frame based evaluation is done for the comparison, it essentially measures object detection, detection refers to the location of an object in an image with respect to the ground truth locations. System generated detections are to be associated with the ground truth locations for performance evaluation. If the system detected object will largely overlap the corresponding ground truth detection, it will be annotated as TP (True Positive), if it fails than FP (False Negative) and in the presence of system detection, if it does not overlap ground truth detection than FP (False Positive). For the quantitative evaluation of the performance, several metrics such as, Object Detection Rate (ODR), Precision (P), Tracking Accuracy (TA) & FA/Frame, which were also adopted in [20], [21].

$$ODR = \frac{\#TP}{Total\ \#\ GT\ locations}$$

Object Detection Rate can also be annotated as Detection Probability.

$$P = \frac{\#TP}{Total\ \#\ detection - \#split + \#merged}$$

P reflecting the preciseness of the system, 100% precision reflects all detections are correct. For a frame if Ground Truth (GT) is one and the system detects it as two,

overlapping the same GT than there will be one TP & one split. Similarly if GT is two and the system detects as one than there will be two TP & one merge.

$$TA = \frac{\#Accurately\ Tracked\ Frames}{\#Tracked\ frames}$$

For each frame, we define tracking as accurate if the predicted point of position of object falls onto the same GT position. FA/Frame is defined as the false alarms per frame.

Comparison of our method with some proposed approaches for similar scenarios [22], [23], and [27] as shown in table 7.1. For better tracking the detection probability, the tracking accuracy and the precision should be high whereas the false alarm per frame should be as lower as possible**.**

Metrics / Methods	TA	Precision	FA/Frame
Huang et al. [23]	71.1%	68.5%	0.98
Leibe et al. [22]	79.1%	73.1%	0.38
Zhao et al. [27]	82.4%	79.7%	0.21
Ours[1]	88.3%	83.1%	0.12
Ours[2]	96.4%	89.8%	0.08

Table 7.1: Results on video based from PETS database
1: tracking without background consideration
2: tracking with the consideration of background points

The above results show that our method achieves the best performance with greater TA, greater Precision, and low FA/Frame. The testing video 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.

7.2 Conclusion

The proposed technique is qualitatively much better as compared to the previous techniques. Most of the problems in the tracking task are related to occlusion effect which has been extensively reduced to a simple classification task because of the new parameters related to depth which in general in 2-D image is difficult to obtain. The proposed method has a large future scope because background has been in general till now in papers has been marginalized to a mere noise which is generally removed from the scene. Here the background landmarks can be used in a number of manners to improve upon the foreground prediction.

References

[1] G. Rigoll, S. Eickeler, and S. M¨uller, "Person tracking in real-world scenarios using statisticals methods," in IEEE International Conference on Automatic Face and Gesture Recognition – FG'2000, Grenoble, France, March 2000, pp. 342–347.

[2] H. H. Bui, S. Venkatesh, and G.West, "Tracking and surveil- lance in wide-area spatial environments using the abstract hidden markov model," International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 1, pp. 177–195, February 2001.

[3] Dezert J., Smarandache F., A short overview on DSmT for Information Fusion, Proc. of 10th Int. Conf. on Fuzzy Theory and Techn., Salt Lake City, July 2005.

[4] Smarandache F., Dezert J. (Editors), Applications and Advances of DSmT for Information Fusion, Amer. Res. Press, Rehoboth, 2004, http://www.gallup.unm.edu/ smarandache/DSmTbookl .pdf.

[5] L. Rabiner and B. Juang, "An introduction to hidden Markov models," IEEE ASSp Magazine, vol. 3, no. 1, pp. 4–16, 1986.

[6] Dezert J., Smarandache F., DSmT: A new paradigm shiftfor information fusion, Proc. of Cogis'06 Conf., March 2006, Paris, France.

[7] F. Smarandache, J. Dezert, Proportional Conflict Redistribution Rules for Information Fusion, Proc. 8th Int. Conf. on Information Fusion, Philadelphia (PA), July 25- 29 2005.

[8] G.D. Hager, M. Dewan, C. Stewart, Multiple kernel tracking with ssd, in:
Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 791–797.

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

[10] J. Shin, S. Kim, S. Kang, S.-W. Lee, J. Paik, B. Abidi, M. Abidi, Optical flow-based real-time object tracking using non-prior training active feature model, Real- Time Imaging 11 (3) (2005) 204–218.

[11] B. Gunsel, A. Ferman, A. Tekalp, Temporal video segmentation using unsupervised clustering and semantic object tracking, Journal of Electronic Imaging 7 (3) (1998) 592–604.

[12] Martin A., Osswald C., A new generalization of the proportional conflict redistribution rule stable in terms of decision, see Chapter 2 in this volume.

[13] R. Jacobs, "Control model of human stance using fuzzy logic," Biological Cybernetics, vol. 77, pp. 63–70, 1997.

[14] Yiming Bai and Tieshan Li, " Robust Fuzzy Inference System for Prediction of Time Series with Outliers", in Proc. International Conference on Fuzzy Theory and Its Applications, 2012, pp.394-399.

[15] Weiyu Zhu, Song Wang, Ruei-Sung Lin and Stephen Levinson, "'Tracking of Object with SVM Regression", Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.

[16] J. Ferryman, IEEE Workshop Performance Evaluation of Tracking and Surveillance, 2009.

[17] J. Shi, C. Tomasi, Good features to track, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.

[18] A. Yilmaz, L. Xin, M. Shah, Contour-based object tracking with occlusion handling in video acquired using mobile cameras, Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1531–1536.

[19] VEENMAN, C., REINDERS, M., AND BACKER, E. 2001. Resolving motion correspondence for densely moving points. IEEE Trans. Patt. Analy. Mach. Intell. 23, 1, 54–72.

[20] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybrid-Boosted multi-target tracker for crowded scene," in Proc. IEEE Computer Vision Pattern Recognition Conf., Jun. 2009, pp. 2953–2960.

[21] X. Liu, L. Lin, S. Yan, H. Jin, and W. Jiang, "Adaptive object tracking by learning hybrid template on-line," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 11, pp. 1588–1599, Nov. 2011.

[22] 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.

[23] 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.

[24] SETHI, I. AND JAIN, R. 1987. Finding trajectories of feature points in a monocular image sequence. IEEE Trans. Patt. Analy. Mach. Intell. 9, 1, 56–73.

[25] BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 232–237.

[26] SCHWEITZER, H., BELL, J. W., AND WU, F. 2002. Very fast template matching. In European Conference on Computer Vision (ECCV). 358–372.

[27] 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.

[28] COMANICIU, D., RAMESH, V., ANDMEER, P. 2003. Kernel-based object tracking. IEEE Trans. Patt. Analy. Mach. Intell. 25, 564–575.

[29] ISARD, M. AND MACCORMICK, J. 2001. Bramble: A bayesian multiple-blob tracker. In IEEE International Conference on Computer Vision (ICCV). 34–41.

[30] HUTTENLOCHER, D., NOH, J., AND RUCKLIDGE, W. 1993. Tracking nonrigid objects in complex scenes. In IEEE International Conference on Computer Vision (ICCV). 93–101.

[31] M. Piccardi (October 2004). "Background subtraction techniques: a review". IEEE International Conference on Systems, Man and Cybernetics 4: 3099–3104.

[32] CHEN, Y., RUI, Y., AND HUANG, T. 2001. Jpdaf based hmm for real-time contour tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 543–550.

[33] VITERBI, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inform. Theory 13, 260–269.

[34] AVIDAN, S. 2001. Support vector tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 184–191.

[35] Yilmaz, A., Javed, O., and Shah, M. 2006. Object tracking: A survey. ACM Comput. Surv. 38, 4, Article 13 (Dec. 2006), 45 pages. DOI = 10.1145/1177352.1177355 http://doi.acm.org/10.1145/1177352.1177355

[36] Gonzalez, R. C. & Woods, R. E. (2001). Digital Image Processing, Addison-Wesley Longman Publishing Co., Inc.,Boston, MA.

[37] Sugandi, B.; Kim, H.S.; Tan, J.K. & Ishikawa, S. (2007). Tracking of moving object by using low resolution image, Proceeding of Int. Conf. on Innovative Computing, Information and Control (ICICIC07), Kumamoto, Japan, (4 pages).

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

[39] Lipton, A; Fujiyoshi, H. & Patil, R.(1998) .Moving target classification and tracking from real-time video, Proceeding of IEEE Workshop Applications of Computer Vision, pp. 8-14.

[40] 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.

[41] Stringa, E. (2000). Morphological change detection algorithms for surveillance applications, Proceeding of British Machine Vision Conf., pp. 402–412.