Object Tracking in Audio-Visual Scene

Subhanjan Ray

Roll No-710EC4139

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela

2015

Object Tracking in Audio Visual Scene

A thesis in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Technology

In

Electronics & Communication Engineering

&

Master of Technology

In

Communication & Signal Processing

by

Subhanjan Ray

Roll No-710EC4139

Under the Supervision of

Prof. L. P. Roy

Department of electronics and Communication

National Institute of Technology Rourkela

Rourkela

2015

DECLARATION OF ORIGINALITY

I hereby declare that this thesis was composed entirely by me and all information in this thesis has been obtained and presented in accordance with academic rules and ethical conduct. The work reported herein was conducted by me in the Department of Electronics and Communication Engineering at National Institute of Technology, Rourkela. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Subhanjan Ray

Roll No- 710EC4139

Department of ECE

NIT Rourkela

CERTIFICATE

This is to certify that the work in this thesis with the title"Object Tracking in Audio Visual Scene" by Mr. Subhanjan Ray has been carried out under my supervision in partial fulfillment of the requirements for the degree of Bachelors of Technology in Electronics and Communication and Masters of Technology in Communication and Signal Processing during session 2010-2015 in the Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela, and this work has not been submitted elsewhere for a degree.

Dr. L.P Roy
Assistant Professor
Department of Electronics and Communication
NIT Rourkela

Acknowledgement

I would like to thank my advisor, Professor L. P. Roy, for his excellent guidance and support throughout the thesis work. He has motivated me to be creative and taught the ethics of research and the skills to present ideas and writing the thesis.

I would also like to express gratitude to our HOD Prof. Kamala Kanta Mohapatra, for providing the support and facilities for the completion of this project. I would like to thank Faculty Members, Electronics and Communication department for their constant encouragement.

I am also grateful to all friends and colleagues for their support and for being actively involved in my step by step progress. The discussions and conversations with these people was always very productive and motivating.

Finally, I would like to thanks my parents who have always been very supportive and understanding throughout my life.

Subhanjan Ray

Roll No- 710EC4139

Department of ECE

NIT Rourkela

Abstract

The thesis aims at tracking of an object using visual and audio information simultaneously. Presently such kind of systems have limited use in traffic monitoring, security surveillance systems, spying and espionage etc. After substantial literature review it was found that Audio-Visual fusion is one of the techniques yet to be explored for object tracking due to the difficulty involved in synchronizing both the modalities viz audio and video. The advantage of such a system is that it is more robust in challenging environments. The aim is to overcome the shortcomings that are found in standalone audio and video techniques. In environments where low lighting conditions are found, the video techniques fail to track the object since the installed cameras don't receive the proper images but the audio tracking assists in such cases. Similarly, there might be situations in which there is a lot of background noise, room reverberations or any combination of the above factors. In such a situation, audio tracking fails but video tracking works robustly. In this thesis, a colour based tracking approach using particle filter has been implemented by entering the gray scale indices of the red, green and blue planes of the object's image into the system as the input. Then an audio source localization method based on Simplex Optimization is implemented. This approach is based on capturing the sound signals from a scene by a set of microphones and then locating the source location by using time delay estimation techniques and simplex optimization. Then an attempt to fuse the information obtained from both modes has been studied.

Key Words:- Particle Filters, Time Delay Estimate, Simplex Optimization

List Of Figures

Contents

Chapter 1

Introduction

The ongoing research on object tracking in audio visual scenario has attracted many scholars & researchers. Detecting the objects in the video and tracking its motion to identify its characteristics has been emerging as a demanding research area in the domain of signal processing and computer vision. The tracking of object based on visual information mostly consists of two parts viz object detection and tracking.[8] The object detection is mostly done by background subtraction [9] and tracking uses algorithms such as mean shift tracker[10],[13], kalman filter[11],[12] and particle filters[3]. Similarly, tracking of an object based on audio information consists of Time Delay Estimation [14],[15] and Source Localization using optimization techniques[16]. In a brief introduction of this thesis, the feature descriptors that are employed in tracking to provide a description of the objects being tracked, object detection techniques and tracking methods are discussed. The merits and demerits of each approach has been analyzed as well.

1.1 Motivation

Tracking objects in video sequences of surveillance camera is nowadays a demanding application. There are many existing methods of object tracking but all have some drawbacks. Some of the extensively researched models till date are contour-based models, region-based models and feature point-based models. Tracking via sound based models also faces problems of multipath and background noise.To overcome the drawbacks that all these methods possess, method combining the audio and video approaches is presented. The motivation is to make the tracker work in different environments such that the audio tracker works where the video tracker fails and vice versa. Situations like bad lighting or occlusions can make video based trackers unfruitful but in those situations audio trackers work robustly. Similarly acoustic reverberations and background noise are some of the problems faced in audio trackers which the video tracking compensates. Thus, an integrated approach combining both information is expected to help in obtaining a better result.

1.2 Objective

The objective of the thesis is to implement robust audio-visual tracking systems. In this thesis, a colour based object tracking system has been implemented. The system implemented is capable of tracking a single target of a specific colour in a video sequence. It is robust in performance with occlusions. However, more work needs to be done in case of similar coloured backgrounds. A acoustic source tracking technique based on time delay estimation and simplex optimization has been implemented via simulation as well. The effect of variation in parameters on tracking performance has been studied as well. The performance of different time delay estimation techniques have been compared too. An attempt has been made to use both audio as well as video techniques simultaneously but could not be completed within the limited scope of time.

1.3 Object Tracking :- An Overview.

A. Contour-based object tracking model

The contour based object tracking or Active contour model uses the outline of an object in a frame to track it [17], where the objects are tracked by assuming their outlines as boundary contours. These boundaries are then continuously updated in each subsequent frame to obtain a trajectory of the object. The active contour model being a discrete approach takes the advantage of the point distribution model to limit the shape. One of the primary disadvantages of this approach is that it makes tracking an object difficult since its performance is highly sensitive to the initial boundary contour that is tracked.

B. Region-based object tracking model

The region based object tracking is based on tracking the color distribution of the target [18,19]. It represents the object based on the color. Although it is computationally efficient, it's performance degrades when there are several similarly coloured objects in the same frame. Complications arise when there are similarly coloured occlusions as well. Since the model does not utilize the shape information

of the object, it's performance is sensitive to the colour distribution of the background.

C. Feature point based tracking algorithm

The Feature point based model uses feature points to describe the target [20, 21]. The algorithm is based on 3 steps. The first step consists of feature extraction by extracting elements to track it over the sequence of frames. The second step clusters the extracted elements into higher level features. The third step is feature correspondence in which features extracted from subsequent frames are matched and compared. The difficulty involved in feature point based tracking is feature correspondence step because a feature point in one frame of a video sequence might have similar points distributed over its other frames as well .This leads to ambiguity in tracking the object.

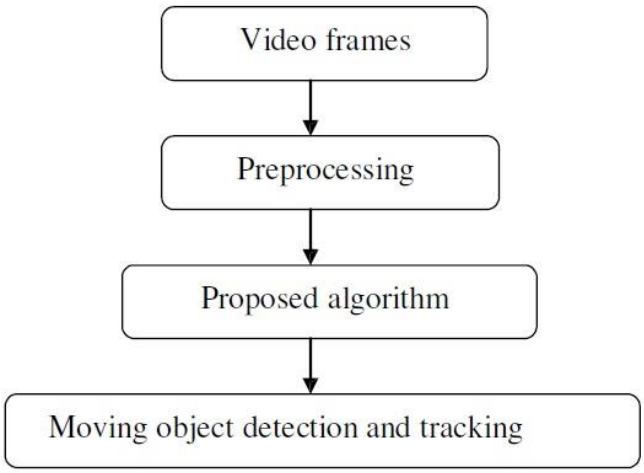

Figure 1 Typical object tracking system in video.

1.3.1 Feature Descriptors

In video object tracking, selection of the right features plays important role. To distinguish the target from other objects, it's visual features must be uniquely described.

A.Color features: Color feature descriptors are used to accentuate the distinguishing factor in case of intensity based descriptors [22]. To describe the color

information of an object RGB color space is usually used .Since a no single colour space can accurately describe the colour features other colour spaces like HSV colour spaces are used as well.

B. Gradient features: Gradient features are important in edge detection of the object in which the contour of the target is already known. Application like human detection in video sequences heavily rely on gradient features.

C. Edges features: The boundary detection of an object mostly uses the edge feature since there is a sharp change in the gray scale intensity of the image just at the object's boundary. To identify such sharp variations, edge detection techniques are employed. A distinctive advantage of the edge features over color features is that they are less sensitive to illumination changes. Most edge detection methods use Canny Edge Detection since it is the optimal algorithm.

D. Optical flow: The motion of individual pixels in a specific region can be determined by a dense field of displacement vectors called optical flow. Assuming the brightness of a pixel to be constant in successive frames it attempts at tracking the same brightness in nearby pixels. It is commonly used in object and video segmentation based tracking algorithms.

E. Multiple features fusion: The multi-feature fusion fuses information obtained from different features and modalities and is an active field of research presently.[25]

F. Biological features: Biological Features are descriptors of physical attributes of humans or animals being tracked. Attention Regions (ARs) and Biologically Inspired Model (BIM) feature the recently used biological features.

1.3.2 Object Detection

The object detection algorithms can be classified into 3 categories mentioned as follows:-

1. Segmentation Based

Segmentation based algorithms segment the image into different sections to isolate the regions of interest or the target. Algorithms used in effective partitioning of images is crucial to the detection process.

A.Graph cut: A graph cut method considers the image as a graph and segments the object's image by graph partitioning .A graph is partitioned into many disjoint subgraphs, by pruning the weighted edges of the graph where weights are assigned on the basis of color, intensity of brightness and texture. Yi and Moon [26] considered graph cut image segmentation as pixel labeling problems. The foreground label was set as 1 and background label was assigned value 0. The minimum graph cut process was then used to minimize the energy function between the two sub graphs and pixel labelling was done.

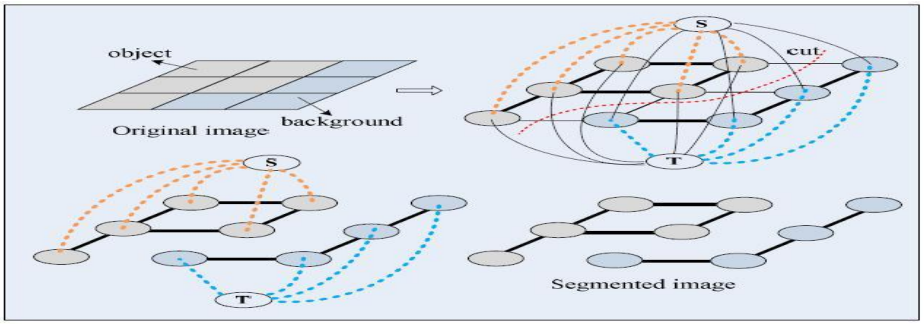

Figure 2 Illustration of graph cut for image segmentation [26]

B. Mean-shift clustering: Mean shift clustering mitigates the image segmentation problem by finding clusters of the image in its frame. Comaniciu and Meer [27] used the Mean-shift clustering to find clusters in the joint spatial and color space, [*l*, *u*, *v*, *x*, *y*], where [*l* , *u*, *v*] denotes the color and [*x*, *y*] is the spatial location. For an input image, a large number of clusters are randomly assigned from the available data. With each subsequent step the clusters keep moving closer to the data's mean lying

inside a multidimensional ellipsoid which has the centers of all the clusters as its center. The shift of these cluster centers defines a vector called the *Mean-shift vector*.

2. Background modeling Based Object Detection

1. Gaussian Mixture Model: The Gaussian Mixture Model tracks the motion of an object by matching it with the known object distribution in the initial frames of a video. It is suitable for dynamic background modeling since it can handle complex distribution of each pixel. It's disadvantages are that it has a slow convergence rate at initial stages of detection and sometimes detects false motions in complex backgrounds.

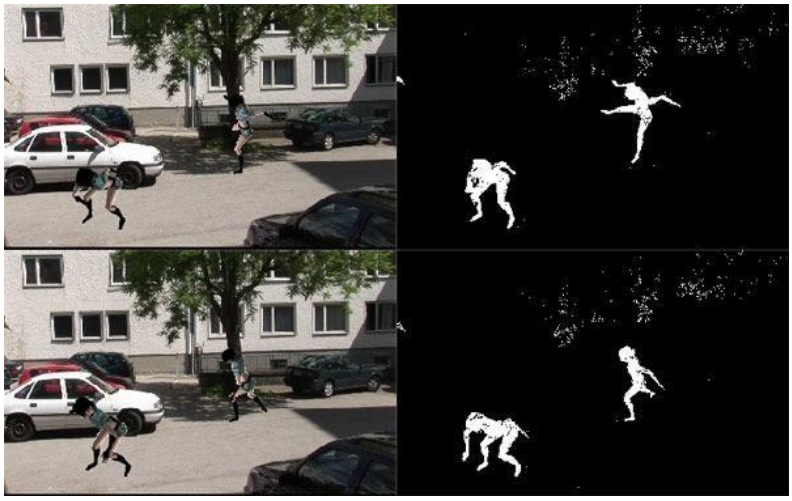

Figure 3 Segmentation using GMM based background modeling. Foreground pixels are marked in white [28]

B. **Eigen-space Decomposition of Background**: Eigen space decomposition projects the current image to eigen space and calculates the difference between the reconstructed and actual image to isolate the foreground. Widely used as a background modeling technique for object tracking it is less sensitive to illumination as well.

C. **Hidden Markov Model**: In recent days, Hidden Markov Model is widely used for background subtraction. It classifies the intensity variations in a pixel into different values according to the environment it is employed in. For an example a camera based

surveillance system can either have a background state, foreground state or the shadow state while tracking cars on a highway .Hidden Markov Models (HMM) used by Rittscher et al. [30] classified small blocks of an image into the above three states. Stenger et al. [31] use HMMs for detecting two events viz light on/off.

1.3.3 Object Tracking

The function of the object tracker is identification of the target in the frame of a video and estimating the trajectory of the object in the subsequent frames over a period of time. The object detection task and object correspondence establishment task between the instances of the object across frames can be done separately or jointly. In the first scenario, the object location is determined in a single frame by object detection techniques and correspondence is established across time by object tracker. In the second scenario, information obtained from each frame helps in determining the object region and estimation of correspondence is done jointly by repetitive updating of object region and its location.

1.2.3.1: Statistical Methods of Tracking

A. **Kalman Filters**: The Kalman Filter, being a single object state estimation filter is used as an estimator to predict and correct the state vector of the system. For a linearly modeled system, Kalman Filter can most accurately describe the discrete time process by the following equations.

1) Process equation

$$x_{k+1} = Ax_k + w_k$$

Where x_k is the system state vector, w_k is additive white Gaussian noise, *A* is the process transition matrix and subscript k denotes state number.

2) Measurement equation

$$z_k = Hx_k + v_k$$

Where z_k is measurement vector, v_k is the Gaussian measurement noise vector and *H* is the measurement matrix.

The most important steps of Kalman filter are prediction (time update) step and correction (measurement update) step. The new state of the variables are predicted by a state model.

Prediction Step: The process equation and measurement equation describes a linear model. As x_k is not measured directly, therefore the information provided by measured z_k is used to update the unknown state x_{k+1}. A priori estimate of state $x_{k+1}^-$and error covariance estimate P_{k+1}^- is obtained for the next time step.

$$x_{k+1}^- = A\hat{x}_k$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k$$

Where Q_k is the noise covariance matrix.

Correction Step: The current observation is incorporated into a priori estimate from the time update in the measurement update equations to obtain an improved posteriori estimate. In the time and measurement update equations, $\hat{x}_k$ is an estimate of the system state vector x_k and K_k is the kalman gain and P_k is the covariance matrix of the state estimation error.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V^T)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-)$$

$$P_k = (1 - K_k H_k)P_k^-$$

B. **Particle Filters**: The disadvantage offered by Kalman filter is that it provides optimal solution only when the modeled system is linear and the state variables follow a normal distribution. Therefore when the system is nonlinear and state variables do not follow Gaussian distribution, Kalman filter obtains poor estimations for those state variables. This disadvantage of the Kalman filter can be overcome with the help of particle filtering [3],[7],[31]

In particle filtering, the posteriori probability density of state $p(X_t \mid Z_t)$ at time t is represented by a set of samples $\{s_t^{(n)} : n = 1, \ldots, N\}$ (particles) with weight π_t. The weights determine the importance of a sample, that is, its observation frequency [56]. Particle filter employs a special sampling scheme also known as importance sampling to find a new set of samples. It consists of three steps viz selection of samples randomly, generation of new samples from previous samples and adjustment of weights corresponding to the new sample.

1.4 Acoustic Source Localization

Acoustic source localization is the task of locating a sound source when measurements of the sound field are known . The sound field can be described using physical quantities like sound pressure and particle velocity. These parameters can be instrumental in obtaining an accurate estimate of direction of arrival of sound. Sound Pressure is generally measured using omnidirectional microphones. Although microphones with directional characteristics are available too but they help us in obtaining an estimate of the direction. The source direction information can be obtained by acoustic particle velocity as well. Other approaches using multiple sensors is also possible. Most of these approaches use the time difference of arrival (TDOA) technique.

1.4.1 Methods of Sound Source Localization

Different methods for obtaining either source direction or source location are possible.

1. Particle Velocity or Intensity Vector

Particle Velocity Probes can be used to experimentally determine the particle velocity vector, which contains sensitive directional information.

2. Time difference of arrival

With a sensor array (for instance a microphone array) consisting of at least two microphones it is possible to obtain the source direction using the cross-

correlation function between each microphone's signal. The cross-correlation function between two microphones is defined as

$$R_{x_1,x_2}(\boldsymbol{\tau}) = \sum_{\boldsymbol{n}=-\infty}^{\infty} \boldsymbol{x_1(n)\, x_2(n+\tau)}$$

which defines the extent of correlation between the outputs of two sensors x_1 and x_2. A high value of correlation is indicative of the fact that the maxima of the correlation function is closer to the actual time delay

3. Triangulation

Triangulation technique relies on determination of a point in space with known angles to the point from two other distinct points. It is similar to a scenario where third point of a triangle is to be determined from a known side and two known angles .The source can be assumed to be the third point and the two microphones used can be placed at either end of the known side. Using the direction of arrival information, it is theoretically stated that the source can be localized.

1.5 Organization of the Thesis

The thesis has been organized to consist of 5 chapters. The first chapter consists of the introductory concepts required for an exhaustive understanding of the topic. It consists of a description of object detection and tracking methods used till date and then a coverage of the techniques used for acoustic source localization and tracking. Chapter 2 consists of an in depth coverage of particle filters and its application in object tracking. The algorithm for the same is discussed as well. Chapter 3 consists of details of the acoustic source localization and its algorithm and operational procedure. Chapter 4 consists of the simulation procedure, results and discussion followed by conclusion. Chapter 5 consists of conclusion and future work.

Chapter 2

Video-based object tracking

In this chapter attempts have been made to implement a colour based object tracking mechanism which uses particle filter. In many engineering applications one is faced with the problem the state vector of the system is unknown and has to be estimated by known observations. The solution to this estimation problem is obtained by the posterior probability density function (PDF), which attempts at giving the probability distribution of the state vector using the previous information available through observations. In a system with a linear model and gaussian noise, a parameterization of these distributions can be derived using the Kalman filter. However, in the general, nonlinear model, there is no closed form expression available.

Recently, a simulation based method has been proposed which is capable of state estimation in non-linearly modeled systems with non-gaussian noise [3]. Using a class of methods referred to as sequential Monte Carlo methods or, more commonly, particle filters, it is possible to numerically approximate the posterior density. An attractive feature is that these methods provide an approximate solution to the generic real problem instead of an optimal solution to the approximate problem with lot of assumptions. This is the main reason for the improved accuracy of particle filters over Kalman filter based methods. Another reason why particle filter was preferred over kalman filter is that the kalman filter is optimized for gaussian distributed state variables whereas if the state variables do not follow Gaussian distribution it results in a degraded performance.

2.1 Particle Filters

Particle filters or **Sequential Monte Carlo** (SMC) methods are a set of state estimation algorithms which use Bayesian recursive equations to determine the posterior probability density of the state vector. It uses a set of particles to evaluate posterior density. The state space might be modeled non-linearly and state vector and noise may follow any distribution.It provide a very effective approach of sampling

from a distribution without making any assumptions to the state space model or noise distribution. However, this method is computationally expensive and does not perform well when applied to high-dimensional systems. The Bayesian recursion equations are applied via an ensemble based approach where a group of particles are sampled to represent the distribution and weights are assigned to each denoting the chances of that particle being sampled from the PDF.A resampling step is used to avoid weight collapsing due wide variation in the weights of the particles, where the low weight particles are replaced by particles closer to the ones with heavier weights. The following figure shows the state space model:-

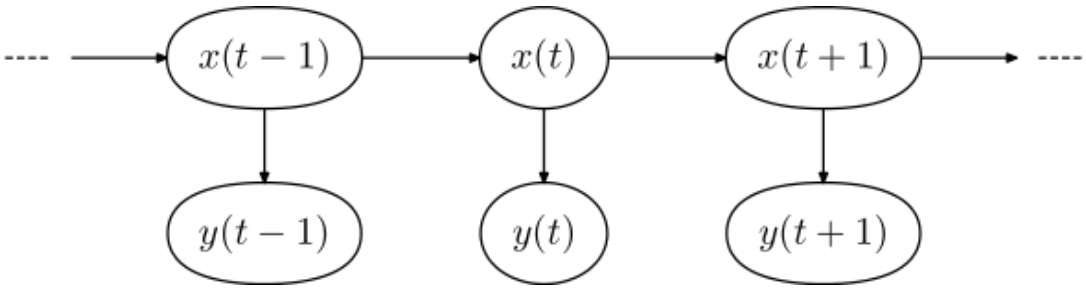

Figure 4 The state space model . The hidden states are denoted by vector **x** and observation states are denoted by vector **y**.

The particle filter is designed for a Hidden Markov Model, where the system consists of hidden and observable variables. The objective of the particle filter is to estimate the values of the hidden states ***x***, the values of the observation process ***y*** is known**.** It aims to estimate the sequence of hidden parameters, x_k for k = 0, 1,2,3,…, based only on the observed data y_k for k = 0,1,2,3. Particle methods assume x_k and the observations y_k can be modeled in this form:

1. x_0, x_1, x_2 ....is a first order Markov process that evolves according to the distribution $p_{x_k|x_{k-1}}$ and with an initial distribution $p(x_0)$.

2. The observations y_0, y_1, y_2 ....are conditionally independent provided that x_0, x_1, x_2 .... are known. Each observation state y_k depends only on the corresponding hidden state x_k.

An example system with these properties is:

$$x_k = g(x_{k-1}) + w_k$$
$$y_k = h(x_k) + v_k$$

If the functions g(.) and h(.) are linear, and if both w_k and v_k are gaussian, the Kalman filter finds the exact Bayesian filtering distribution. If the above conditions are not satisfied the Kalman filter provides an approximate solution to the problem and with sufficient number of particles, particle filter provides a much accurate solution.

2.2 The Implementation of the Model.

1. The model implemented is a colour based tracking model which works robustly in case of rapidly moving objects and handles occlusions satisfactorily.
2. It is an adaptive model in which we enter the video as the input and the gray scale indices of red, green and blue plane of the object to be tracked as target.
3. The video is then divided into a number of frames followed by generating a set of random particles distributed all over the frame.
4. The gray scale indices of the red, green and blue planes at all such points is stored.
5. The Euclidean distance of the stored coordinates from the target coordinates is evaluated and stored as d.
6. The log likelihood estimate otherwise known as Bhattacharya Coefficient [3] obtained by the expression $\log(\frac{1}{\sqrt{2\pi}\sigma}(e^{-\frac{d^2}{2\sigma^2}}))$ The estimate is otherwise known as Bhattacharya Coefficient. As it can be seen greater the similarity with the target, greater is the log likelihood estimate.
7. Based on the likelihood score weights are assigned to the particles and the positions of these particles are updated.
8. The state vector consists of position and velocity information of the particles. The state update equation is denoted by

$$X = F * x + \text{No}$$

Where x is the state vector obtained from the log likelihood function, No is the randomly distributed noise and F is the state transition matrix.The weight updation also takes into account factors like noise and motion of the object.

2.3 The simulation details

The simulation is performed using Matlab 2012 edition software. The video used in the simulation has a resolution of 640*480. It is a short video sequence running for 4 seconds at a frame rate of 25 frames per second. The number of tracking particles is 300 and parameter σ, the variance of the state vector's distribution is set as 10. The target coordinates are set as (161,231,161). The following steps are followed in simulation:-

1. The video file is the input to the system
2. The video file was segmented into frames and each frame was divided into red, green and blue planes with each plane containing the respective gray scale indices.
3. The state vector was randomly initialized as first hypothesis.
4. The frames were captured, RGB index values recorded, The Bhattacharya Coefficient or normalized log likelihood is evaluated and cumulative distribution function is generated.
5. Random variable similar to normalized log likelihood function generated.
6. The state vector is updated with state transition matrix and Gaussian noise.
7. The iterative process is repeated for all the frames.
8. End.

It may be inferred that a larger number of particles leads to a faster convergence of the algorithm since there would be a greater probability of tracking the target with more number of particles. The variation in the parameter σ too would affect the tracking selectivity. The tracking selectivity is expected to be inversely proportional to σ since a greater σ would spread the state vector's particles over a larger space due to greater variance and a smaller value would keep them restricted to a smaller space.

2.4 Assumptions and Limitations

- Since the above technique is a colour based approach, it is not applicable in scenarios where a background similar in colour to the target is involved.\
- It can track multiple similarly coloured objects simultaneously but the accuracy is affected since the number of particles tracking each is reduced.
- Objects that move too suddenly and swiftly might not be tracked by the particles since there is a limit to their sampling frequency due to the frame rate of the video.

Chapter 3

Object tracking via sound source localization

In this chapter the entire object tracking mechanism with audio input has been vividly described and analyzed.

3.1 Geometry of the problem

The geometry of the problem is very critical in sound source localization problems because it is prone to confusions arising due to the presence of aliases. Aliases are points in space from where the microphones receive identical signals. They are caused due to insufficient number of microphones or due to their improper geometrical arrangement which renders some of them redundant. To solve this, the geometry of the problem needs to considered so that that least number of microphones are used for a known situation, thereby eliminating redundancy.

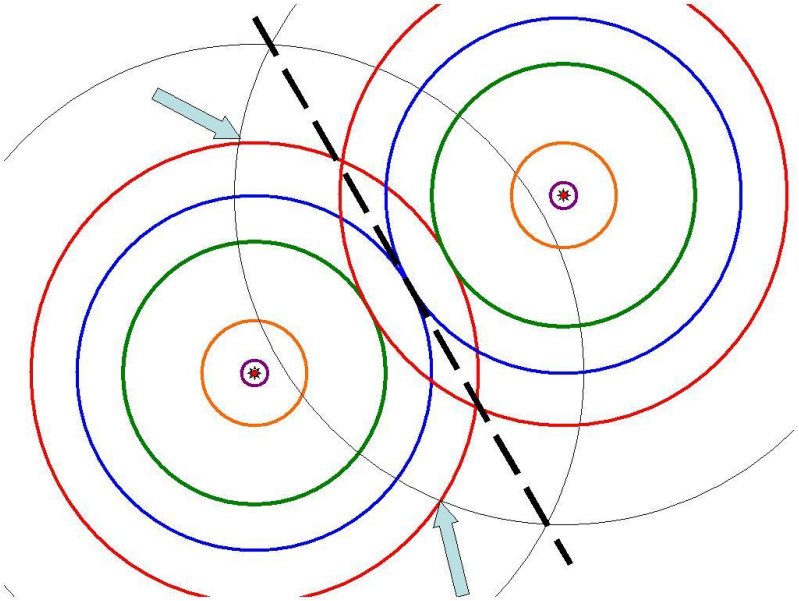

Figure 5 Geometry of two microphones in a plane. Black dashed line is the locus of aliases for zero time delay. The arrows point to a set of aliases.

Let us consider the geometrical condition of 2 microphones and a source. For a known time difference of arrival of the sound signal at the microphones, the path difference is constant since speed of sound is assumed in the homogeneous medium to be constant. That means the sound source must lie on the intersection of two circles having the microphones as the centre. For any two known circles, there are two such points which are each other's aliases .Thus in order to determine the location of a source to the precision of one point, the source must be constrained to move on a line joining the two microphones. In such a case, the two aliases coincide and the location of the source can be determined to a single point.

Now, let us consider the locus of the sound source in the situation above in a general scenario. The signal after being emitted at the source reaches one of the microphones faster than the other, and hence there is a time difference of arrival between them. That, when multiplied with the acoustic velocity in air the path difference between them is obtained. Now, the condition reduces to locus of a point such that the difference in distances between two fixed points remains a constant. This is a standard equation of a hyperbola and hence the source is constrained to a hyperbolic curve in 2 dimensions.

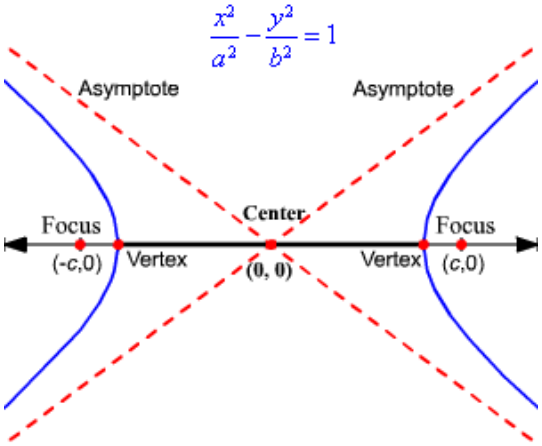

Figure 6 Locus of the sound source with two microphones.

Here, we have the two microphones located at the two foci and the source on one

of the branches. From here, it is understood that in a two microphone system, the source cannot be localized to a single point without having to put some additional constraint on its location. So in order to eliminate this constraint, a third microphone is added to the arrangement. Aim is to add another dimension to the localization problem and it is seen that adding a microphone anywhere on the line joining the existing microphones makes the third microphone redundant. So the three microphones are kept non-linear. Now it may so appear that placing the microphones on the three vertices of a triangle would localize the source to one point. However, it's not always true. Each pair of microphones will lead to a distinct hyperbolic locus. Even if two hyperbolas are taken, each branch may intersect another branch at one, two or at most four points and in reality there are three such hyperbolas. Thus, there can't always be a unique solution in such a case. Such a system is said to be overdetermined and a solution may or may not exist. To further eliminate redundancy, a fourth microphone is added to the system which adds another dimension to the system. In such a case the hyperbola becomes a hyperboloid and solution of the system can be determined by their intersection point. It shall be seen further that to accurately determine the source location, one must place the microphones on the vertices of a tetrahedron. Placing the fourth microphone on the plane containing the previous three microphones degrades the precision of the system by adding further aliases.

3.2 Time delay estimate methods and their comparison

The term cross correlation is indicative of the similarity between two signals as a function of the time lag between them. It is also known as sliding dot product or sliding inner product. This concept can be used to find the time delay between any two versions of the same signal received at the corresponding pair of microphones. For two real time discrete signals, the cross correlation can be defined as:-

$$r_{xy}(l)=\sum_{n=-\infty}^{\infty} x(n) * y(n-l)$$

where n is the time index, l is the time delay and x & y are the signals. In order to account for the reverberation and multipath interference, different methods for calculating the time delay are considered. The setup is that there are two microphones separates by a distance and there is a sound source. The whole system can be modelled by the following equations:-

$r_1(t) = s(t) + n_1(t)$

and

$r_2(t) = s(t - D) + n_2(t)$

where $s(t)$ is the desired signal, D is the time delay, $r_1(t)$ and $r_2(t)$ are the signals received at the microphones and $n_1(t)$ and $n_2(t)$ are AWGN signals.

The methods used for the estimation are as follows:-

3.2.1. Cross correlation (CC) method

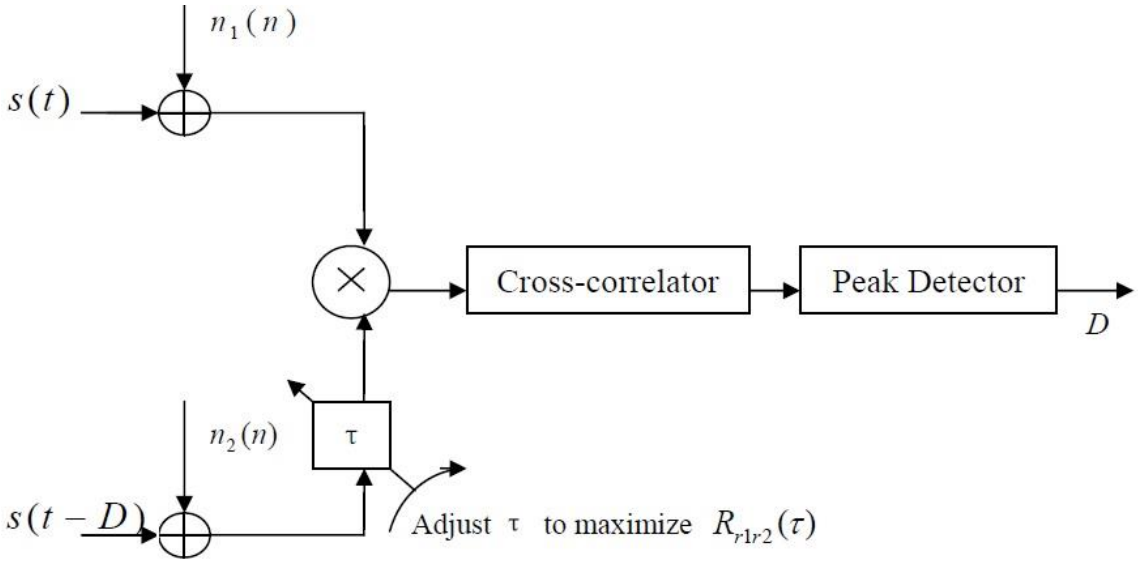

Figure 7 Cross Correlation Processor schematic

The entire process can be modelled by the following equations:-

$R_{r1r2}(\tau) = E[r_1(t)r_2(t-\tau)]$

and

$D_{CC} = arg_\tau \, max[R_{r1r2}(\tau)]$

It is known that any pair of microphones receive two different versions of the same signal from the source or are delayed by different constants **τ**. Thus the cross correlation lets us determine the delay between them by assigning a score called correlation coefficient to each value of delay. The delay for which the score is maximum is the requisite time delay D_{CC}.

3.2.2. Phase Transform Method

The entire method can be modelled by the following set of equations:-

$$R_{r1r2}(\tau) = \int_{-\infty}^{+\infty} \Psi_P(f)\, G_{r1r2}(f)\, e^{j2\pi f\tau} df$$

where

$$\Psi_P(f) = \frac{1}{|G_{r1r2}(f)|}$$

and

$$D_P = arg_\tau\, max[R_{r1r2}(\tau)]$$

Where $\Psi_P(f)$ is the PHAT weighing function.

This procedure is used to whiten the input signals by multiplying their cross spectral density with a weighing function as a result of which the peaks of the correlation function do not spread. Only phase information of the spectrum is preserved after multiplying the weight. It then reduces to inverse Fourier transform of unity with a time shift of τ. In Output a sharp peak is obtained at appropriate delay.

3.2.3.Maximum Likelihood Correlator Method

It is a method similar to that of GCC-PHAT except for the fact that a different weighing function called Maximum Likelihood correlator is used. It improves the time delay estimation by attenuating the signal in the spectral regions where SNR is low and accentuating it where it is high. It gives good results for uncorrelated gaussian signal and noise in single path environments for long observation intervals. The Process can be described by the following equations :-

$R_{r1r2}(\tau) = \int_{-\infty}^{+\infty} \Psi_{ML}(f)\, G_{r1r2}(f)\, e^{j2\pi f\tau} df$

where

$\Psi_{ML}(f) = \frac{1}{|G_{r1r2}(f)|} \frac{|\gamma_{r1r2}(f)|^2}{(1-|\gamma_{r1r2}(f)|^2)}$

and $|\gamma_{r1r2}(f)|^2 = \frac{|G_{r1r2}(f)|^2}{G_{r1r1}(f)G_{r2r2}(f)}$, $D_{ML} = arg_{\tau}\, max[R_{r1r2}(\tau)]$

where $|\gamma_{r1r2}(f)|^2$is the magnitude coherence squared and , $D_{ML}$is the delay. The frequency where coherence is 1 is with highest SNR so they are emphasized and the ones with low values are attenuated.

3.2.4. Average Square Difference Function

The Process can be modelled by:-

$R_{ASDF}(\boldsymbol{\tau}) = \frac{1}{N} \sum_{n=0}^{N-1}[r_1(n) - r_2(n+\boldsymbol{\tau})]^2$

and

$D_{ASDF} = arg_{\boldsymbol{\tau}}\, min[R_{ASDF}(\boldsymbol{\tau})]$

This method evaluates the minimum error square $R_{ASDF}(\boldsymbol{\tau})$ between the signals received and locates the point of minima to obtain the estimated time delay D_{ASDF}.The advantages of this method are that it obtains perfect estimation in the absence of noise.

Computational complexity is low since only additions are involved. Knowledge of input spectrum is not essential here since cross spectral density is not involved.

3.3 Problems encountered in Time Delay Estimation

One of the major challenges faced by TDE techniques is the reverberant environment and background acoustic noise. It creates a problem since several temporally and spatially correlated versions of the same signals are received at the microphones and there is difficulty in achieving a distinct peak for the cross correlation. However its interference may be quantified by considering an

equivalent value of SNR .The more the reverberation, lesser is the SNR. Another issue with ML and PHAT methods is that to get an accurate value of cross spectral density a very long observation interval needs to be taken. However, that is not practically possible so an estimate of the quantity is obtained which has a variance that can lead to large errors at times. But the problem with long intervals is that the signal might not be stationary or have a constant delay over the entire interval. So, there is a tradeoff between SNR and observation interval. The higher the SNR shorter is the need for a long observation interval. Thus the main issue is the low value of SNR. There exists a threshold value of SNR for each of the methods below which the time delay estimation fails. Thus the threshold SNR is an important parameter that can be used to compare the approaches under different conditions. One which has the lowest threshold SNR must be the preferred one.

3.4 Source Localization Technique

This technique basically relates the time delay estimate to the geometry of the problem. In real time scenario, the inputs are the signals captured at the microphones, then the time delay estimates are calculated for each pair of microphones, then the obtained data is processed and the position of the source is determined. The algorithm for the entire process can be described by the following steps:-

- The number of microphones used for the tracking is specified (say k). All their positions are specified in terms of their Cartesian coordinates (x_i, y_i, z_i) where i = 1,2,3,4,……..k. All the microphones are kept in a non-coplanar fashion keeping in mind the geometry of the problem so as to avoid aliases.

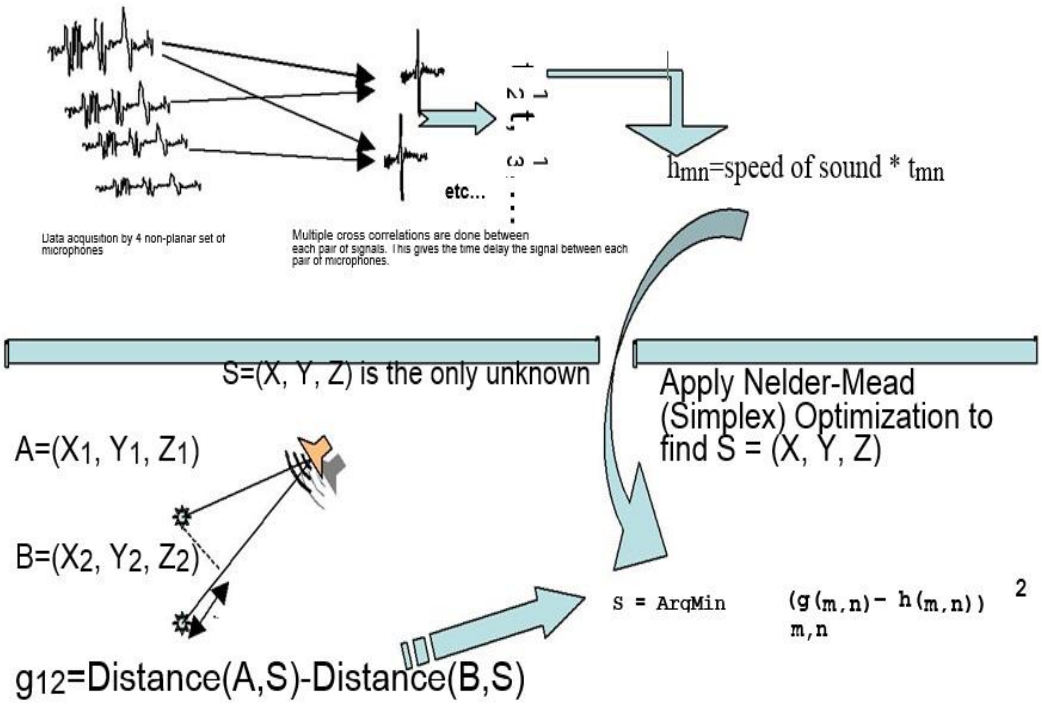

Figure 8 The Source localization routine

1. The signals obtained from each source is recorded and the time delay between the mth and nth microphone is stored as $t_{m,n}$where m & n are microphone indices and vary from 1 to k.
2. The corresponding path delays are determined by multiplying each value of the time delay with the speed of sound which is assumed to be a constant c for a homogeneous medium. The difference between path differences of mth and nth microphones from the source is stored as $h_{m,n}$.
3. The position of the source is then assumed to be S(x,y,z) which is unknown. Then its geometrical distance from each of the microphones is represented algebraically as $\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}$ for ith microphone for i varying between 1 and k.
4. The path difference between the mth and nth microphone from the source is then stored in variable $g_{m,n}$.Then the difference between $g_{m,n}$ and $h_{m,n}$ is calculated and is squared. This constitutes the objective function which is to be minimized by using an algorithm called Nelder-Mead Algorithm or

Simplex Algorithm.

5. The minima of the objective function is recorded and the coordinates at that point denote the estimated value of the source.
6. The same algorithm is repeated for each sampled time index and the position of the source is successfully tracked.

3.5 Nelder Mead Optimization technique:-

. The Nelder Mead algorithm, or the simplex algorithm is a widely used direct search method for optimization problems in which the multiple variables involved with the non-linear objective or cost function have no constraints on them. The algorithm uses a sequence of simplices which degenerate gradually to converge at a solution.

The Direct search methods are those methods which do not use the derivative information of the cost function to find the optimal point. Instead it simply starts from a point and looks for another point where the value the value of the function is lower. Its advantage over derivative based methods is that it can be applied in cases where the function is non-differentiable and sometimes discontinuous as well.

Figure 9 A 3-dimensional simplex

A simplex (plural *simplexes* or *simplices*) is a generalized version of a triangle

or a tetrahedron in multi-dimension. A simplex may be defined as the smallest convex set containing the known vertices. A k-simplex is a *k*-dimensional polytope which is the convex hull of its $k + 1$ vertices. Suppose there are k+1 affinely independent points with position vectors u_i such that i=0,1,2,3,…..k, then the vectors $u_i - u_0$ will be linearly independent. For example a point, a line segment, a triangle, a tetrahedron and a 5 cell are 0- simplex,1-simplex,……4-simplex respectively. Other direct search methods other than Nelder Mead also maintain a non- degenerate simplex at all steps which finally degenerates to a point.

The Nelder Mead algorithm was proposed as a method for minimizing a real-valued function $f(x)$ for $x \in R^n$. Four scalar parameters must be specified to define a complete Nelder Mead method: coefficients of *refection* (ρ), *expansion* (χ), *contraction* (γ), and *shrinkage* (σ). According to the original Nelder Mead paper[32], these parameters should satisfy

$$\rho > 0; \quad \chi > 1; \quad \chi > \rho \; ; \; 0 < \gamma < 1; \text{ and } 0 < \sigma < 1:$$

The nearly universal choices used in the standard Nelder Mead algorithm are

$$\rho = 1; \qquad \chi = 2; \qquad \gamma = \frac{1}{2} \qquad and \qquad \sigma = \frac{1}{2}$$

Statement of the algorithm:- At the beginning of the *k*th iteration, $k \geq 0$, a non-degenerate simplex Δ_k is defined, along with its $n + 1$ vertices, each of which is a point in R^n. It is always assumed that iteration *k* begins by ordering and labeling these vertices as $x_1^{(k)}, \ldots, x_{n+1}^{(k)}$, such that

$$f_1^{(k)} \leq f_2^{(k)} \ldots\ldots\ldots\ldots\ldots\ldots f_n^k \leq f_{n+1}^{(k)};$$

where $f_i^{(k)}$ denotes $f(x_i^{(k)})$. A different simplex is defined in each subsequent iterations such that $\Delta_{k+1} \neq \Delta_k$. The point $x_1^{(k)}$ is considered as the *best* point or vertex, to $x_{n+1}^{(k)}$ as the *worst* point and to $x_n^{(k)}$ as the *next-worst* point since the motivation is to minimize the objective function *f*. Similarly, $f_n^{(k+1)}$ is referred as the

worst function value, and so on. A single generic iteration is specified, the result of which is either a single accepted vertex which replaces the worst vertex to form the next simplex or a new simplex formed with the best vertex and n other vertices generated from the algorithm in case a shrink is performed.

One iteration of Algorithm NM (the Nelder Mead algorithm).

- **Order.** Order the $n + 1$ vertices to satisfy $f(x_1)$ $f(x_2)$ $f(x_{n+1})$, using the tie-breaking rules described below.
- **Reflect.** Compute the *reflection point* x_r from the following formula:-

  $x_r = \bar{x} + \rho(\bar{x} - x_{n+1}) = (1 + \rho)\bar{x} - \rho x_{n+1}$;

  where $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$ is the centroid of n best points i.e all vertices except x_{n+1}. Then $f_r = f(x_r)$ is calculated. If $f_1 \leq f_r < f_n$, accept the reflected point and x_r and terminate the iteration.

- **Expand** :-If $f_r < f_1$, calculate the expansion point x_e, where it is denoted by the following relation:-

  $$x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \rho\chi(\bar{x} - x_{n+1}) = (1 + \rho\chi)\bar{x} - \rho\chi x_{n+1}$$

  and evaluate $f_e = f(x_e)$. If $f_e < f_r$, accept x_e and terminate the iteration; otherwise (if $f_e \geq f_r$), accept x_r and terminate the iteration.

- **Contract:-** If $f_r \geq f_n$, perform a *contraction* between $\bar{x}$ and the better of x_{n+1} and x_r.

  Outside. If $f_n \leq f_r < f_{n+1}$ (i.e., x_r is strictly better than x_{n+1}), perform an outside contraction and calculate

  $x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \gamma\rho(\bar{x} - x_{n+1}) = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}$

  and evaluate $f_c = f(x_c)$. If $f_c \leq f_r$, accept x_c and terminate the iteration;

  Inside :- If $f_r \geq f_{n+1}$, perform an *inside contraction*: calculate

  $$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma\bar{x}) + \gamma x_{n+1}$$

  and evaluate $f_{cc} = f(x_{cc})$.If $f_{cc} < f_{n+1}$, accept x_{cc} and terminate

iteration otherwise, go to step 5 (perform a shrink).

- **Perform a shrink step**:- Evaluate *f* at the *n* points $v_i = x_1 + \sigma(x_i - x_1)$, $i = 2, \ldots, n+1$ The (unordered) vertices of the simplex at the next iteration consist of $x_1, v_2, \ldots, v_{n+1}$.

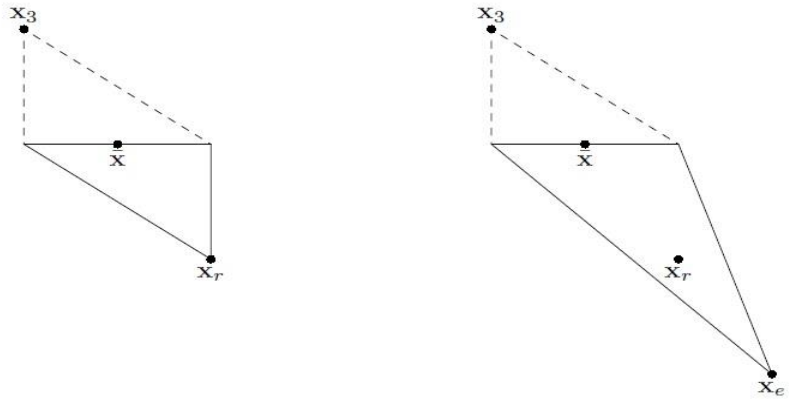

Figure 10 Nelder Mead simplices after a reflection and an expansion step. The original simplex is shown with a dashed line.

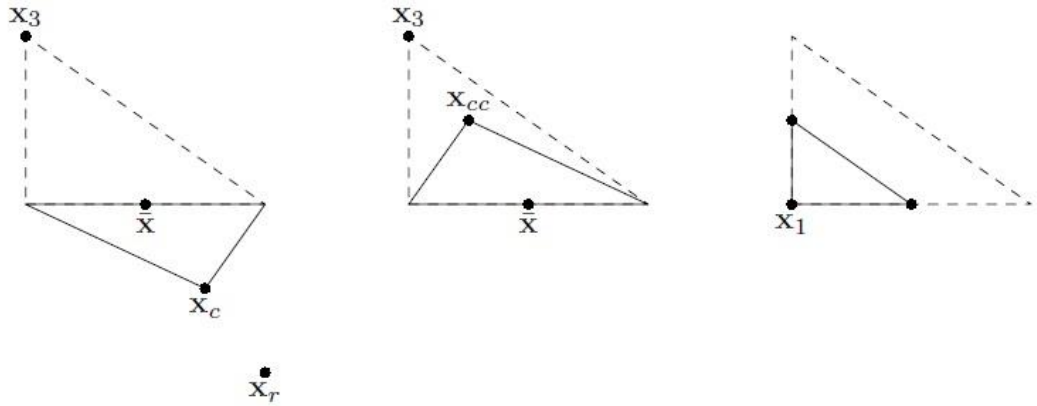

Figure 11 Nelder Mead simplices after an outside contraction, an inside contraction, and a shrink. The original simplex is shown with a dashed line

Figures 9 and 10 show the effects of reflection, expansion, contraction, and shrinkage for a simplex in two dimensions (a triangle), using the standard coefficients $\rho = 1$, $\chi = 2$, $\gamma = \frac{1}{2}$, and $\sigma = \frac{1}{2}$. Observe that, except in a shrink, the

one new vertex always lies on the (extended) line joining x and x_{n+1}.Furthermore, it is visually evident that the simplex shape undergoes a noticeable change during an expansion or contraction with the standard coefficients.

The following tie-breaking rules are adopted, which assign to the new vertex the highest possible index consistent with the relation $f(x_1^{(k+1)}) \leq f(x_2^{(k+1)}) \leq$ $\ldots\ldots\ldots\ldots \leq f(x_{n+1}^{(k+1)})$.

Nonshrink Ordering Rule**:-** When a nonshrink step occurs, the worst vertex x_{n+1}^{k} is discarded .The accepted point created during iteration k, denoted by $v^{(k)}$ and takes j+1 th position in the vertices of Δ_{k+1} where

$$j = max_{0 \leq l \leq n}\{l \mid f(v^{(k)}) < f(x_{l+1}^{(k)})\};$$

all other vertices retain their relative ordering from iteration *k*.

Shrink ordering rule. If a shrink step occurs, the only vertex carried over from Δ_k to Δ_{k+1} is $x_1^{(k)}$. Only one tie-breaking rule is specified, for the case in which $x_1^{(k)}$ and one or more of the new points are tied as the best point: if

$$\text{Min } \{f(v_2^{(k)}), \ldots\ldots\ldots, f(v_{n+1}^{(k)})\} = f(x_1^{(k)});$$

then $x_1^{(k+1)} = x_1^{(k)}$. Beyond this, whatever rule is used to define the original ordering may be applied after a shrink.

3.6. Assumptions and limitations

The real time scenario for the sound source is very complicated for modeling due to various factors like multipath, clutter, background noise etc. The environment in which the simulations are performed are subject to various assumptions and limitations mentioned as follows

1. It is assumed that the sound source is the lone sound source ,has minute physical dimensions and radiates uniformly in all directions.

2. Multipath interference due to reflection from the ground, surrounding buildings, objects and clutter is minimal.

3. There is no interference due to any other noise source present.

4. The sound source is assumed to be emitting stationary signals when signals are captured.

5. The medium is assumed to be homogeneous and variations in sound's velocity is neglected

6. The microphones must be placed in proper geometrical configuration as required by acoustic processing techniques

7. Although perfect solutions to the localization problem is not possible, some improvements are possible if certain factors mentioned below are taken care of:-

- Geometrical setup of the acoustic receivers.
- Accuracy of the acoustic receiver setup.
- Uncertainties in the location of the microphones.
- Synchronization issues with the receivers.
- Error in calculation of time delays.
- Interference due to noise sources.

Chapter 4

Results and Discussions

4.1 Particle filter based video tracking

The outputs and observations of the given program are as follows:-All total 5 frames distributed over a span of 100 frames have are showed here and inferences drawn are stated along with that.

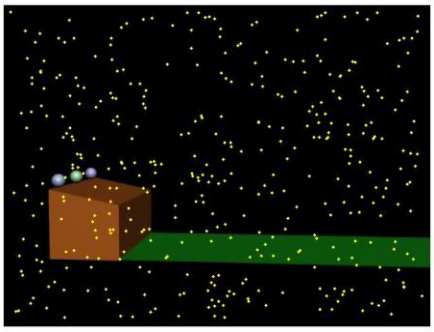

Figure 12 Initial hypothesis

The above video is of 3 different coloured moving balls viz. blue, violet and green with green being the target. In first hypothesis 300 random points are distributed across the frame which are denoted by blue dots. This is the first endeavor at sampling of points to generate the likelihood estimate. Note that blue dots are randomly distributed across all parts of the frame. This is frame no. 1.

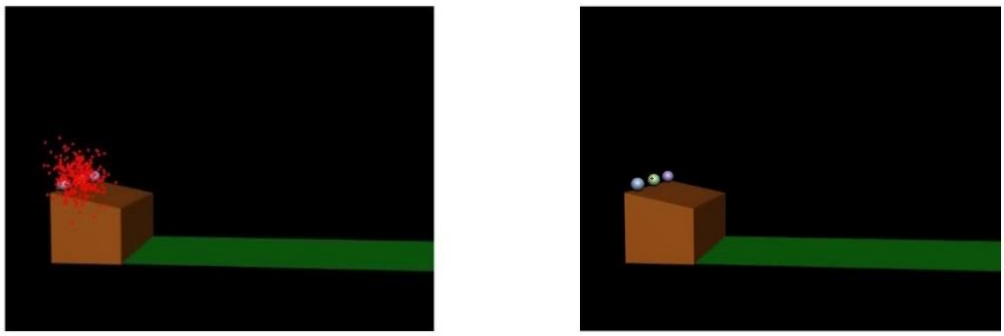

Figure 13 Frame 5 particles and their centroid

Now the object tracking has already begun in frame no. 5. The video frame rate is 25 frames per second .So it is the scene after $\frac{1}{5}^{th}$ of a second. The red particles are tracking the green ball. One thing to noted is that the centroid of all these particles is the best estimate of the object's location. However we prefer cluster of particles because it gives us a range over which tracking can be performed. The second figure shows the centroid of the particles.

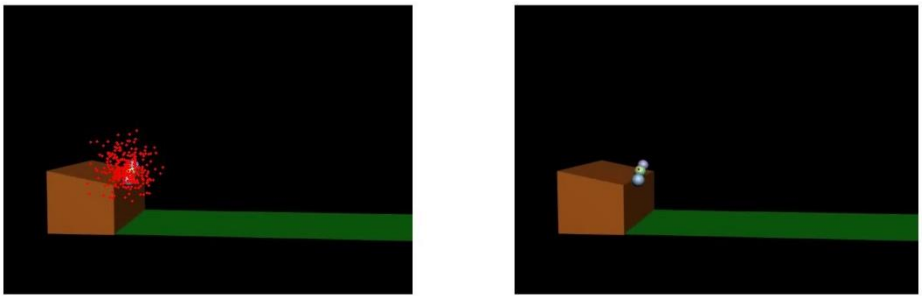

Figure 14 Frame 21 particles and Centroid of the particles

The green ball is being tracked progressively in frame 21 as well. In this observation we try to observe the change in the target's position and the best estimate of it's position as indicated by the red coloured particle indicating

the centroid. Since the video is a collection of 100 frames, some of those frames are sampled and presented for observation.

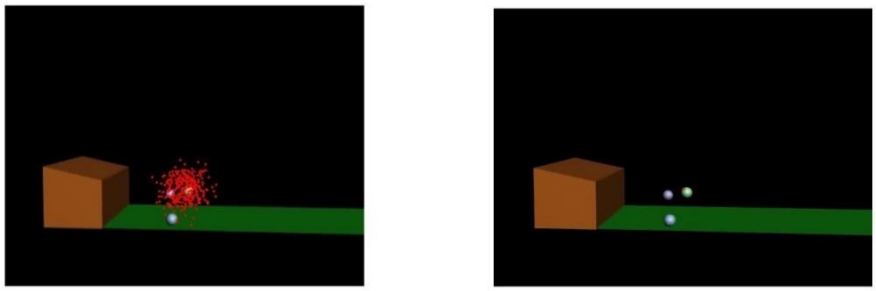

Figure 15 Frame 31 particles and their centroid

Even after a quick movement down the ledge the ball is tracked in frame no. 31. The centroid gives the most likely estimate of the particle's position in the frame. It is denoted by the single red dot. This shows the importance of parameter σ which helps in tracking sharp movements and swiftly moving objects. Greater the σ, greater is the variance in particle distribution and greater is the range over which tracking is performed.

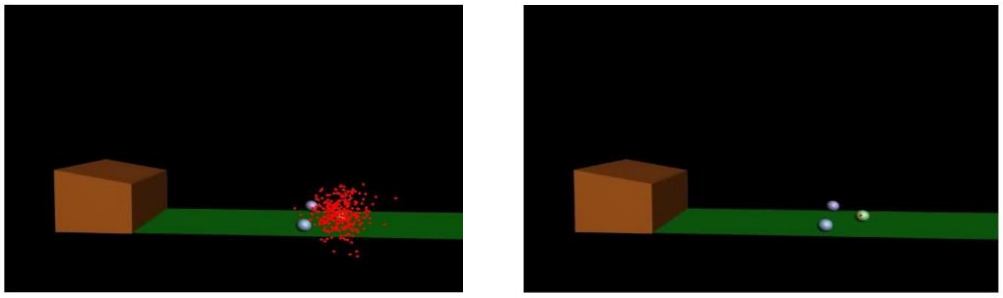

Figure 16 Frame 57 particles and their centroid

Till the end close to frame 57 the ball is being tracked. It is to be noted that the tracker perform satisfactorily even if the background is somewhat similar in colour to the object being tracked.

There are two major parameters involved in the tracking process. The first is the dimension of the state vector and the second is the tracking selectivity parameter σ.The effect of the dimension of state vector is discussed in the following figure:-

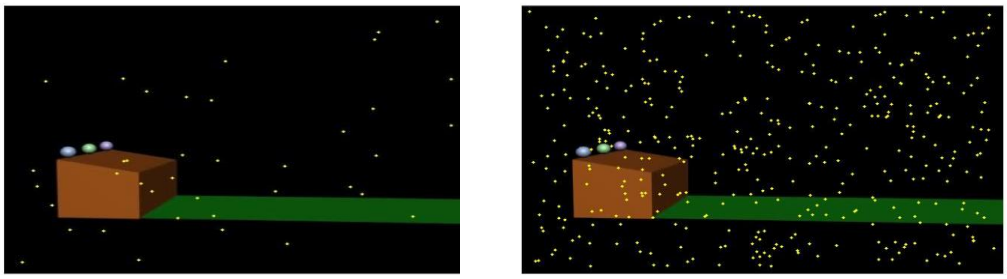

Figure 17 The first hypothesis with (a)40 particles and (b)400 particles

The set of figures illustrated above show the first hypothesis for two cases when the dimension of the state vector is 40 in fig 17(a) and 400 in fig 17(b). As is evident from the figures, in first case ,since the number of particles is low and is spread randomly over the frame, it is less likely that a point would be close to the target since they are spread over a large area whereas the object to be tracked is small, whereas in second case the large area is sampled by a proportionately large number of particles, thereby accentuating the probability of sampling a point close to the target in the Euclidean plane.

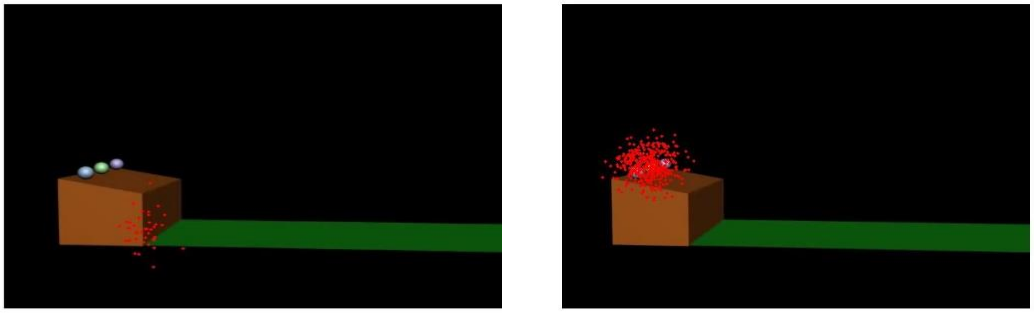

Figure 18 Tracking performance with (a)40 particles and (b)400 particles

The above set of figures compares the tracking performance of 40 particles and 400 particles .Both depict the frame number 7 of 100 frame sequence. As it can be observed, the tracker fails at converging when the number of particles is low and successfully tracks the target with 400 particles.

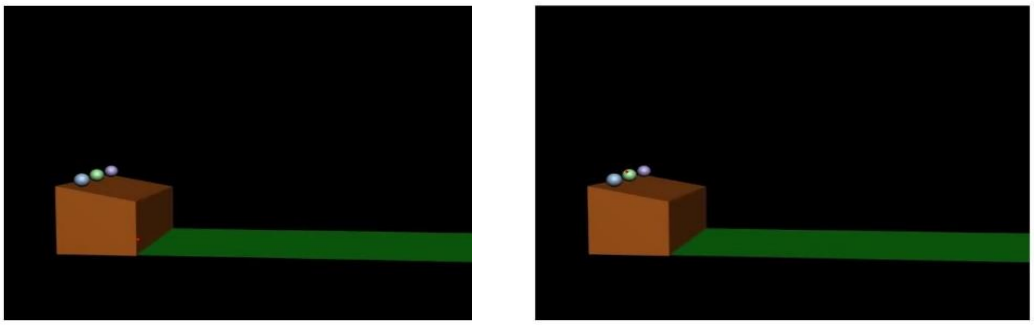

Figure 19 Centroid position with (a)40 particles and (b)400 particles

The set of figures shown above are indicative of the best estimate of the target's position in case of 40 and 400 particles. The difference in the position of red dots indicates that higher the number of particles in a tracker , higher is the probability of a successful tracking.

Now, we assess the effect of variation of the tracking selectivity σ on the performance:-

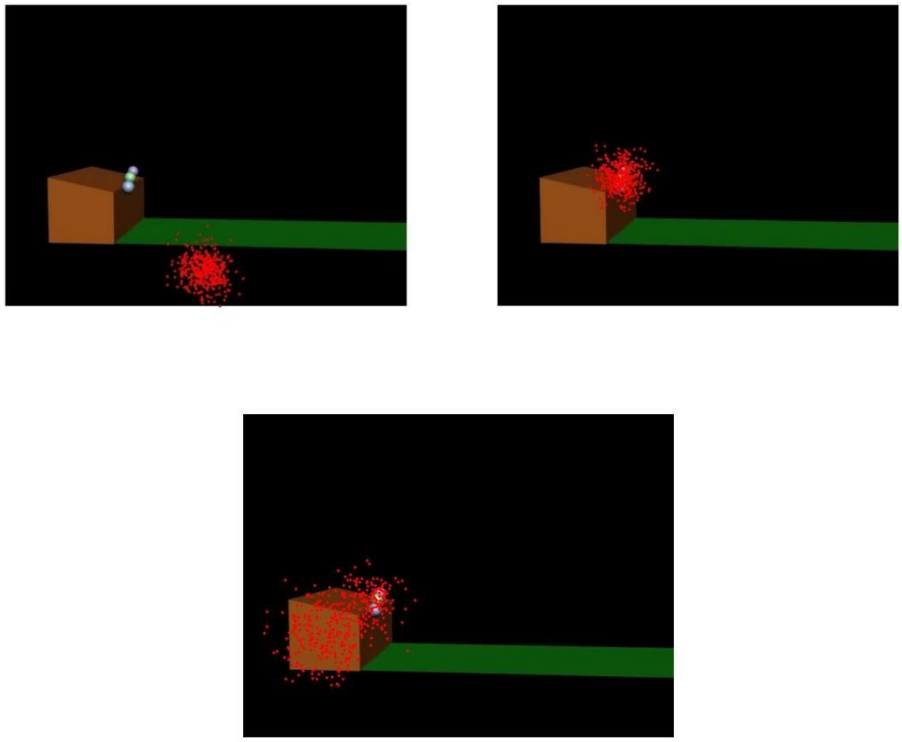

Figure 20 Tracking performance with σ = (a) 1,(b) 10,(c) 100

All the above observations are for frame number 40 of the video sequence .As is pretty evident from the observation, for a successful tracking result and fast convergence, σ should neither be too less nor be too large. If it is too small e.g 1 in fig 20(a) it does not shift it's position in case it picks a wrong point initially since it searches for a correct point in its vicinity over a very small area, whereas if it is too large in fig 20(c), the particles are spread over a very large area and hence proper resampling near the target cannot be done .So both the cases either converge slowly or might not converge at all. In either of these case the number of particles have to be increased to ensure confirmed fast convergence. However if there is a balance and it is somewhere in between these two values viz. 10 then there

we get a proper set of particles stacked within an appreciable area with good selectivity as well. Therefore, the tracking performance depends on choice of optimal set of values for the state vector size and σ.

4.2 Time Delay Estimation

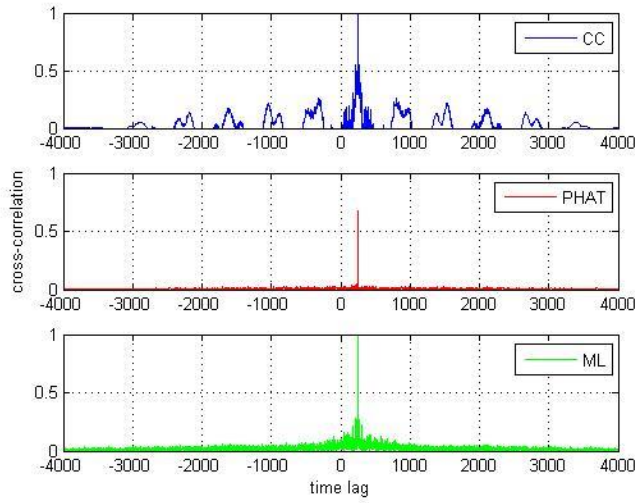

Figure 21 Time Delay Estimated given by CC, PHAT & ML method

The above figure gives a comparison of time delay estimation performance of CC, PHAT and ML correlator method. The time delay for a simulated pair of signals is calculated. The sampling frequency of the signal is 7.418 Hz and sample time is 0.13 ms. The time delay introduced in the simulation is 243T. All the three methods have peaks at that delay. However the ones in PHAT and ML correlator have much better primary to secondary peak separation.

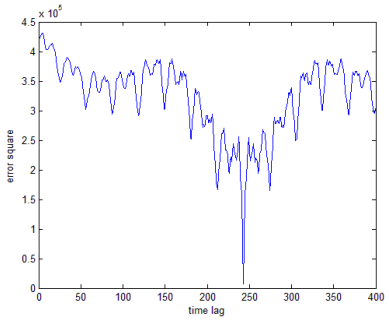

Figure 22 The Average Square Difference Function v/s Time Lag

As can be seen from Fig 22 the Average Square Difference Function is minimum at 243T. This correctly estimates the time delay

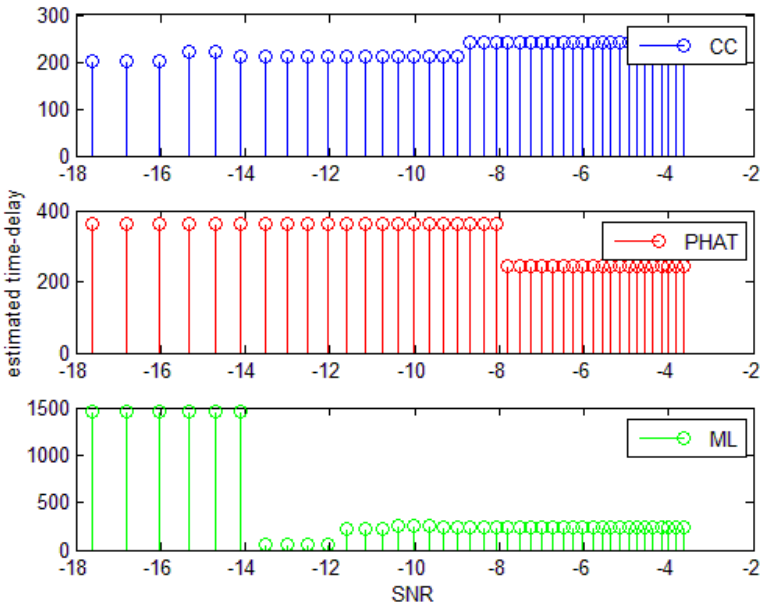

Figure 23 SNR comparison of CC,PHAT & ML methods

As evident from the figure 23 ,the CC and the PHAT methods have their threshold SNR at close to -8.5 dB and -8 dB respectively. But the ML method has a threshold of -12 dB i.e it can perform satisfactorily for lower SNR values. Hence it is preferred in low SNR conditions.

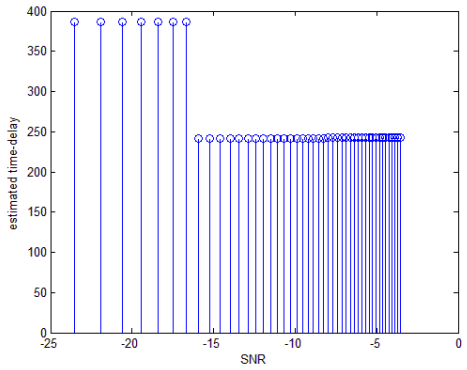

Figure 24 SNR performance of ASDF method

The Average Squared Difference Function method exhibits a threshold SNR of -15 dB and hence is the most suitable one for low SNR conditions. It is computationally efficient as well.

4.3 Acoustic Source Localization Simulations

The simulations were performed for acoustic source localization by entering the source's coordinates as input .The only difference between the simulated and real time conditions is that in real time scenario the time delay is calculated by cross-correlating two signals whereas in simulation it is to be calculated by using geometrical path delays between a pair of microphones and the source point. Then by using simplex optimization, the same point is tracked without knowledge of the source position. The simulation results are presented since the real time data sets for audio could not be found.

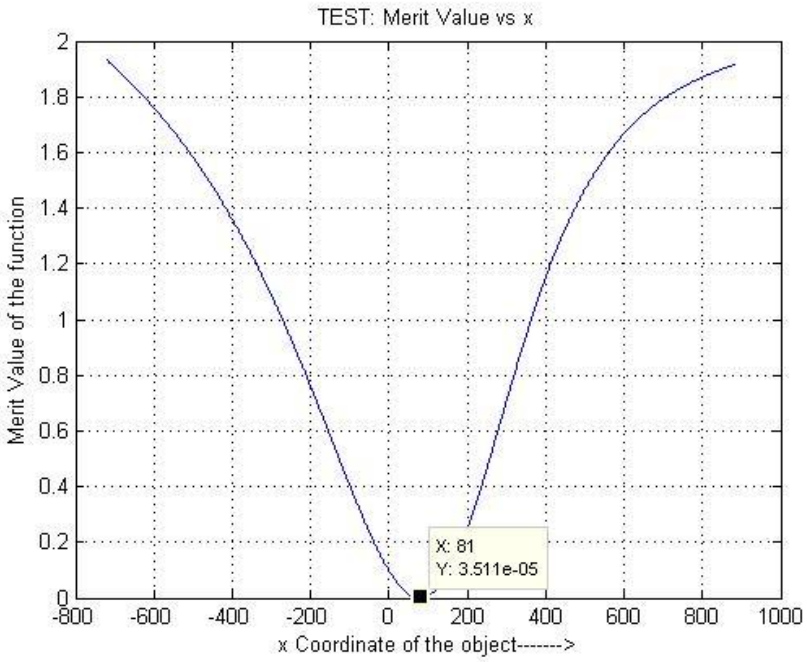

Figure 25 Objective function using simplex optimization for target's X coordinate

The trajectory of the target starts at t = 0 and ends at t= 80 in the simulation .Therefore, the merit value of the function is found to be minimum at the object's X coordinate i.e close to 80.

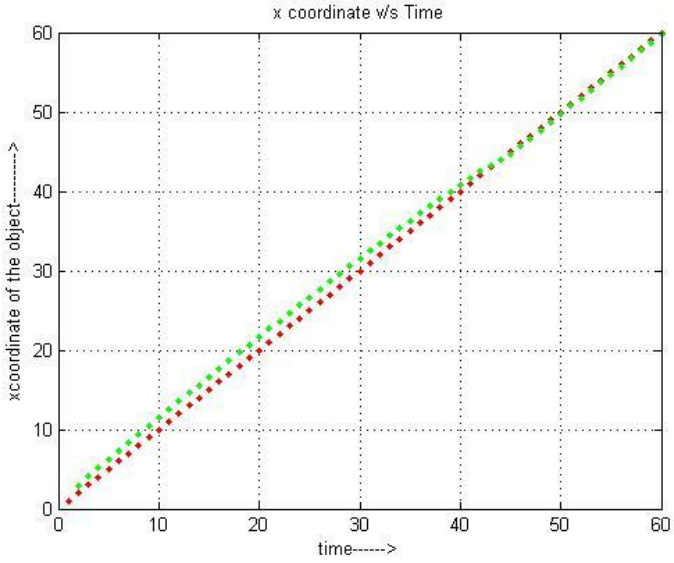

Figure 26 Perfomance of the algorithm in tracking X coordinate

The X coordinate of the object is successfully tracked using the audio tracking technique. The red dots denote the actual object's X coordinates whereas the green dots denote the tracked coordinates. For better scaling the X coordinates are shown from t= 0 to t= 60 in fig 4.14. Following the same convention, the same was done to track the Y and Z coordinates too and results are presented as follows:-

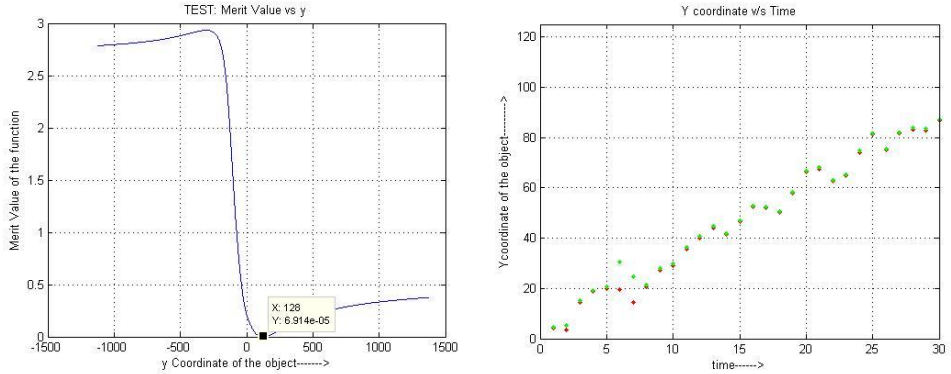

Figure 27 (a)Objective function computed for tracking Y coordinate of the target and (b) Tracking performance for target's Y coordinate

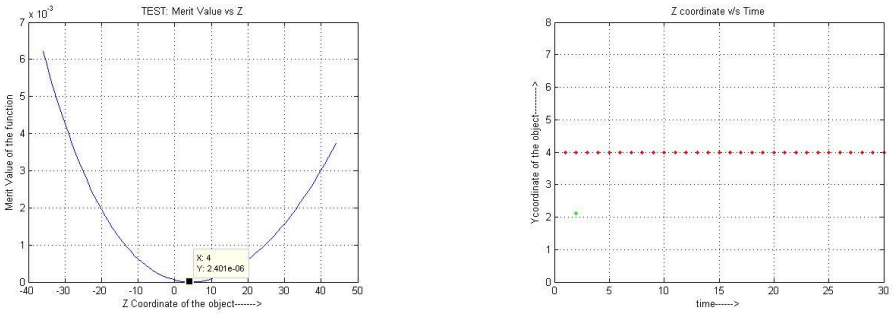

Figure 28 (a) Objective function evaluated for target's Z coordinate and Fig 4.15(b) Tracking performance for target's Z coordinate

Hence, a combination of these x, y and z coordinates can be accurately tracked as well. The simulated results are in accordance with the object's position.

Chapter 5

Conclusion and future work

5.1 Conclusion

By an algorithm based on the application of particle filter an effective color based tracking system is implemented. The effect of variation in parameters like tracking selectivity and state vector size on object tracking accuracy in video is studied. Various time delay estimation techniques are implemented and compared to study what specific advantages each of them offer. A source localization technique combining time delay estimation and simplex optimization is implemented via simulation. Simulation results track the input to appreciable accuracy. An endeavor to use the audio-visual information could be made by using the data obtained from both modalities. However, the information obtained from both modalities were found to be unsynchronized with each other. An audio-visual data set was taken and tracking was performed in both cases separately via audio and video data. However there was huge disparity between both the values obtained.one probable reason for the same could be the different coordinate systems in which the audio and video tracking results exist. If the two coordinate systems could be matched, then a combined output assigning adaptive weightages to audio and video information could be implemented.

5.2 Future Work

- The Audio and Video modalities can be fused or combined via statistical multi-modal fusion techniques to get a result which may perform better in robust conditions.
- The effect of uncertainty in microphone location on source localization can be studied.
- The effect of reverberations and background noise on sound source localization may be studied.

Bibliography

1. Robertson Neil, Hopgood James R. Person Tracking and Audio Visual Fusion Proc. 9th IET Data Fusion & Target Tracking Conference 2012.

2. Aarabi, Parham, and Safwat Zaky. "Robust sound localization using multi-source audiovisual information fusion." *Information fusion* 2, no. 3 (2001): 209-223.

3. Nummiaro, Katja, Esther Koller-Meier, and Luc Van Gool. "An adaptive color-based particle filter." *Image and vision computing* 21, no. 1 (2003): 99-110.

4. Zhang, Yushi, and Waleed H. Abdulla. "A comparative study of time-delay estimation techniques using microphone arrays." *Department of Electrical and Computer Engineering, The University of Auckland, School of Engineering Report* 619 (2005).

5. Taddese, Biniyam Tesfaye. "Sound Source Localization and Separation." (2006).

6. Lagarias, Jeffrey C., James A. Reeds, Margaret H. Wright, and Paul E. Wright. "Convergence properties of the Nelder--Mead simplex method in low dimensions." *SIAM Journal on optimization* 9, no. 1 (1998): 112-147.

7. Hol, Jeroen D. "Resampling in particle filters." (2004).

8. Porikli, Fatih, and Alper Yilmaz. "Object detection and tracking." In *Video Analytics for Business Intelligence*, pp. 3-41. Springer Berlin Heidelberg, 2012.

9. Piccardi, Massimo. "Background subtraction techniques: a review." In *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4, pp. 3099-3104. IEEE, 2004.

10. Zhou, Huiyu, Yuan Yuan, and Chunmei Shi. "Object tracking using SIFT features and mean shift." *Computer vision and image understanding* 113,

no. 3 (2009): 345-352.

11. Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25, no. 5 (2003): 564-577.

12. Weng, Shiuh-Ku, Chung-Ming Kuo, and Shu-Kang Tu. "Video object tracking using adaptive Kalman filter." *Journal of Visual Communication and Image Representation* 17, no. 6 (2006): 1190-1208.

13. Comaniciu, Dorin, Visvanathan Ramesh, and Peter Meer. "Real-time tracking of non-rigid objects using mean shift." In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 142-149. IEEE, 2000.

14. Knapp, Charles, and G. Clifford Carter. "The generalized correlation method for estimation of time delay." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 24, no. 4 (1976): 320-327.

15. Bjorklund, Svante, and Lennart Ljung. "A review of time-delay estimation techniques." In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, pp. 2502-2507. IEEE, 2003.

16. Huang, Yiteng, Jacob Benesty, Gary W. Elko, and Russell M. Mersereati. "Real-time passive source localization: A practical linear-correction least-squares approach." *Speech and Audio Processing, IEEE Transactions on* 9, no. 8 (2001): 943-956.

17. D. Serby, E. K. Meier, and L. V. Gool, "Probabilistic Object Tracking Using

Multiple Features", IEEE Proc. of International Conference on Pattern Recognition Intelligent Transportation Systems, Vol. 6, pp. 43-53, 2004.

18. L. Li, S. Ranganath, H. Weimin, and K. Sengupta, "Framework for Real-Time Behavior Interpretation From Traffic Video", IEEE Transaction On Intelligent Transportation Systems, , Vol. 6, No. 1, pp. 43-53, 2005.

19. P. Kumar, H. Weimin, I. U. Gu, and Q. Tian, "Statistical Modeling of Complex Backgrounds for Foreground Object Detection", IEEE Trans. On Image Processing, Vol. 13, No. 11, pp. 43-53, November 2004.

20. Z Zivkovi, "Improving the selection of feature points for tracking", In Pattern Analysis and Applications, vol.7, no. 2, Copyright Springer-Verlag London Limited, 2004.

21. J. Lou, T. Tan, W. Hu, H. Yang, and S. J. Maybank, "3D Model-Based Vehicle Tracking", IEEE Trans. on Image Processing, Vol. 14, pp. 1561-1569, October 2005.

22. Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computing Surveys, Vol. 38, No. 4,2006.

23. Wei-Bin Yang, Bin Fang ; Yuan-Yan Tang ; Zhao-Wei Shang ; Dong-Hui Li, "Sift features based object tracking with discrete wavelet transform ", International Conference on Wavelet Analysis and Pattern Recognition,ICWAPR,pp. 380-385,2009.

24. Wei-Bin Yang, Bin Fang ; Yuan-Yan Tang ; Zhao-Wei Shang ; Dong-Hui Li, "Sift features based object tracking with discrete wavelet transform ", International Conference on Wavelet Analysis and Pattern Recognition,ICWAPR,pp. 380-385,2009.

25. Hanxuan Yang , Ling Shao, Feng Zheng , Liang Wangd, Zhan

Song,"Recent advances and trends in visual tracking: A review", Elsevier Neurocomputing 74 (2011) pp. 3823–3831, 2011.

26. Faliu Yi, Inkyu Moon," Image Segmentation: A Survey of Graph-cut Methods",International Conference on Systems and Informatics (ICSAI 2012), 2012.

27. Comaniciu, D. And Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Patt. Analy. Mach. Intell. 24, 5, pp.603–619.

28. Vu Pham, Phong Vo, Hung Vu Thanh, Bac Le Hoai, "GPU Implementation of Extended Gaussian Mixture Model for Background Subtraction", IEEE-RIVF 2010 International Conference on Computing and Telecommunication Technologies, Nov. 01-04, 2010.

29. Rittscher, J., Kato, J., Joga, S., AND Blake, A. 2000. A probabilistic background model for tracking. In European Conference on Computer Vision (ECCV). Vol. 2. pp.336–350.

30. Stenger, B., Ramesh, V., Paragios, N., Coetzee, F., AND Burmann, J. 2001. Topology free hidden markov models: Application to background modeling. In IEEE International Conference on Computer Vision (ICCV). pp.294–301.

31. Tanizaki, H. , “Non-gaussian state-space modeling of nonstationary time series.” J. Amer. Statist. Assoc.82, pp.1032–1063, 1987.

32. Nelder, John A., and Roger Mead. "A simplex method for function minimization." *The computer journal* 7, no. 4 (1965): 308-313.