SEMI-AUTOMATIC GROUND-TRUTH TRAJECTORY EXTRACTION ON IMAGE SEQUENCES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT KARABIYIK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2017

Approval of the thesis:

SEMI-AUTOMATIC GROUND-TRUTH TRAJECTORY EXTRACTION ON IMAGE SEQUENCES

submitted by **MURAT KARABIYIK** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Mübeccel Demirekler
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Department, METU _____

Prof. Dr. Aydın Alatan
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Emre Özkan
Electrical and Electronics Engineering Department, METU _____

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Engineering Dept., Hacettepe University _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. 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.

Name, Last Name: MURAT KARABIYIK

Signature :

ABSTRACT

SEMI-AUTOMATIC GROUND-TRUTH TRAJECTORY EXTRACTION ON IMAGE SEQUENCES

Karabıyık, Murat
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Mübeccel Demirekler

February 2017, 77 pages

In this thesis, offline semi-automatic ground-truth trajectory extraction technique is proposed that uses measurements of detector as basis. The unknown camera motion of the videos used throughout the thesis makes the problem even more challenging. The camera motion is estimated by using a novel method which uses a special Kalman filter. Background objects are discriminated from the targets and they are used to estimate the camera motion. Two different trackers are implemented to extract the ground-truth. Measurements of the detector are tracked by using Tracker-1. The tracks resulted from Tracker-1 are associated by using Tracker-2. The velocity difference between the target and the camera is used both for position predictions of Tracker-2. The user of the program gives the true target information for the first frame. The output of Tracker-2 gives the raw ground-truth and it is smoothed via Kalman smoother. The output of the Kalman smoother gives the ground-truth. Finally, an example tracker which is used in real time tracking problems is evaluated by comparing the ground-truth and measurements of the tracker which is evaluated.

Keywords: Ground-truth Extraction, Camera Motion Estimation, Kalman Smoother, Evaluation of Tracker, Offline Tracking

ÖZ

GÖRÜNTÜ DİZİLERİNDE REFERANS TAKİP VERİLERİNİN YARI OTOMATİK OLARAK ÇIKARTILMASI

Karabıyık, Murat
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

Şubat 2017 , 77 sayfa

Bu tezde, bir dedektörün hedef tespit ölçümleri kullanılarak bir çevrimdışı yarı otomatik referans takip datası çıkarma tekniği önerildi. Tez boyunca kullanılan videoların kamera hareketlerinin bilinmemesi problemi daha da zor hale getirdi. Kamera hareketi zamana bağlı değişen bir Kalman filtresi kullanılarak tahmin edildi. Arka plan objelerinden kaynaklanan izler hedef izlerinden ayırt edilerek kamera hareketi tahmin modeline eklendi. İki farklı izleyici referans takip datası çıkarmak için kullanıldı. İlk izleyici kullanılarak dedektörden alınan hedef tespit ölçümleri takip edildi. Bunun sonucunda ortaya çıkan hedef takip dizileri ikinci izleyici kullanılarak birbiriyle ilişkilendirildi. Hedef ile kamera arasındaki hız farkı hesaplanıp ikinci izleyicinin pozisyon tahminlerine eklendi. Hangi izin hedef olduğu bilgisi kullanıcı tarafından verildi. İkinci izleyicinin sonucunda ham bir referans takip dizisi ortaya çıktı ve bu dizi Kalman düzgeci kullanılarak düzleştirildi. Bu işlem sonucunda ortaya çıkan ölçüm dizisi referans takip verisini verdi. En son olarak, gerçek zamanlı izleyicinin hedef takip ölçümleri referans takip datası kullanılarak değerlendirildi.

Anahtar Kelimeler: Referans Takip Datası, Kamera Hızı Tahmini, Kalman Düzgeci, Video İzleyici Değerlendirmesi, Çevrimdışı Takip

To my family and people who are reading this page

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Mübeccel Demirekler for her constant support, guidance, wisdom and friendship. It was a great honor to work with her.
There are a lot of people that were with me throughout the work. It is not possible to write down why each of them is important to me and this work, because it will take more space than the work itself. I am very lucky to have them all. So I'll just give names of some of them; Erhan Poyrazoğlu, Merve Dilek and Merve Özkardeş. I would also like to thank all my colleagues in ASELSAN and my team leader Cumhur Ünlü for his understanding.
Lastly, sincerest thanks to each of my family members for supporting and believing in me all the way through my academic life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

GTES	Ground-truth extraction system
RTT	Real time tracker
VN	Video-N
SoV	Source of the video
IMM	Interacting multiple model
GNN	Global nearest neighbor
TT	Typical tracker

CHAPTER 1

INTRODUCTION

In a video which has got various traces, there is a need to extract true position of the desired object that we call 'target'. This process is called the 'ground-truth trajectory extraction'. The simplest way of obtaining the ground truth trajectory is to mark the position of the target by hand at each frame. Marking the position of the target requires high human labor which is not desirable. Once extracted, ground truth trajectory is used to assess the tracking performance of any video tracking system.

Most of the video tracking systems uses two consecutive frames to associate the object candidate(s) found in one frame to the object(s) of the previous frame [27]. In these systems no a priory information about the possible target motion is assumed. The use of the motion model and the predictions given by the Kalman filter or any other filter is another approach to video object tracking. In the last two decades this approach is applied to both single and multiple video target tracking problems [4, 15, 9, 23, 20]. While using this approach usually a detector gives candidate positions of the target at each frame, called the measurements, and the tracker tracks the object by utilizing the measurements by a simple Kalman filter or much more sophisticated filters like particle filter. In the recent years there are approaches that combine the robust tracking techniques with the stochastic filtering techniques [23, 20] or particle filters [9].

An association between the consecutive frames is necessary for all tracking techniques and any association uses features obtained both from the target and the background. The features extracted consist of color, contour information, intensity etc. Color, which is very useful in discriminating feature, cannot be used in IR applications. Shape of the target is also an important discriminating feature. For small targets of few pixels this feature cannot be used either.

Tracking of objects in an image sequence depends on the motion of the camera. Camera may be stationary, moving according to some unknown external source with unknown aim or may try to track the target. In all different cases the motion of the target as observed on the scene would be different. So for all these cases motion models of the target should better be different.

The first aim of this thesis is to fully or partially automate the extraction of the true target position on an image sequence that is already recorded. So this is an offline operation. The second aim is to generate an evaluation system for a video tracker. The evaluation system evaluates the tracks or track parts generated by any tracker as true, false, unobserved etc.

The videos used in this study are IR videos. We assume that the camera that records the image sequence is trying to track a target although not always successful. Since the video is recorded by a tracking camera the tracked object is at the center of the scene unless the tracking system makes a mistake.

The block diagram of the system developed in this thesis is given in Fig. 1.1. System has a detection part which extracts possible target positions. This block extracts at most N candidate target positions for each frame of the video. Tracker-1 is an on line multi target tracker which uses Kalman filtering. Data association is done by Global Nearest Neighboring method. The output of Tracker-1 is a set of tracks. A typical output of Tracker-1 is shown in Figures 3.4 and 3.5. Tracker-2 uses the output of Tracker-1 and mainly associates the tracks. Tracker-2 uses the estimated motion of the camera while making associations. Tracker-2 makes off line operations for this purpose. The final block is the smoothing block that which is again requires off line operation is done.

In this thesis, the detection block (Fig. 1.1: detector) is not implemented. This block is taken as ready to use. It is designed to detect small IR targets. Although the performance of this block directly affects the performance of the ground truth extraction system in our evaluation we consider only the image sequences that the detector gives the true target position, if available, as one of the candidates.

The camera is assumed to be a tracking camera, i.e. it tries to track the target. However the motion of the camera is assumed to be unknown. During tracking the camera makes a smooth motion to keep the target at the center of the scene. Whenever the camera loses the target it makes arbitrary motions that may be large. These time in-

tervals usually correspond to target occlusions. During and/or at the end of such time intervals for the tracking system it is important to find the target again. Unknown camera motion creates a problem for the associations made in these cases. So in this thesis camera motion estimator is designed. Camera motion estimator uses the tracks that are classified as background tracks. A time varying single tracker model is used to track all the objects that belong to the background. The output of this system is not only the trajectory of the target but also the information about the existence of the target at all frames.

The system shortly described above is given in Fig 1.1. is called Ground Truth Extraction System (GTES).

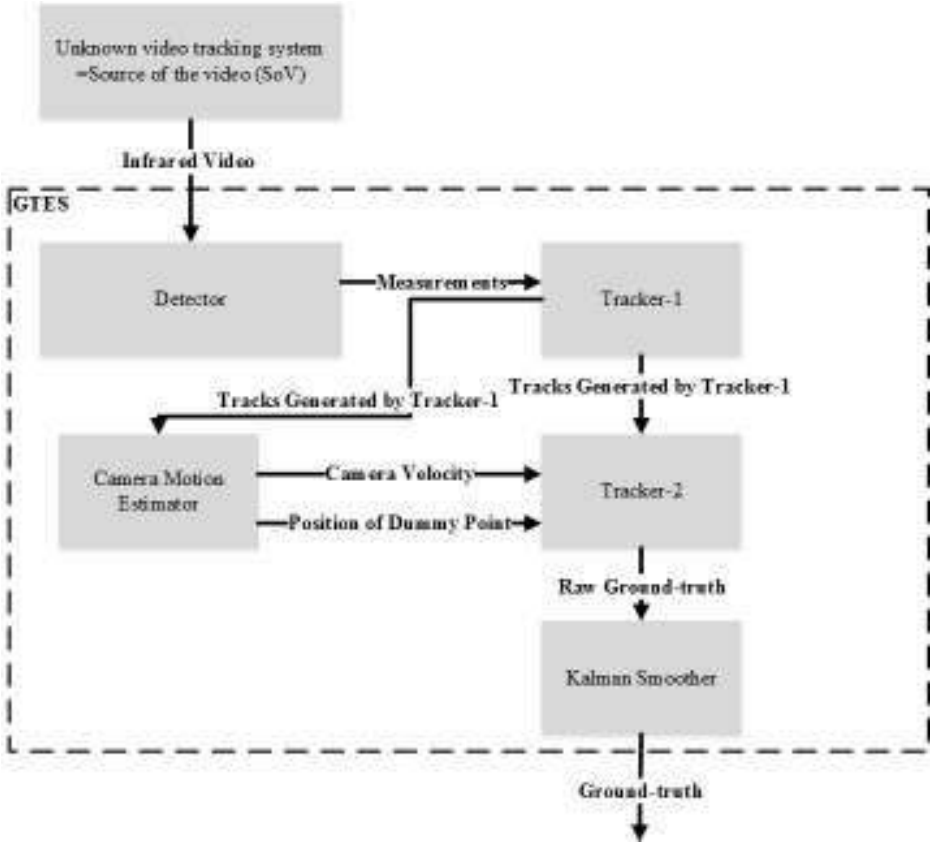


Figure 1.1: Block Diagram of the System

The second aim of the thesis is to generate an evaluation system for the assessment of the performance of video trackers. The ground truth extracted by the above system is compared with the trajectory of the target produced by the tracker of a tracking system under investigation. The assessment indicates the time intervals that the tracker

makes some decision errors like ‘wrong object is tracked’, ‘observable target is not found’ etc. Furthermore it gives the RMS error between its smoothed output and the output of the tracker.

We have used 7 infrared videos in this study. The length of the videos changes between 250-1000 frames (400 frames in avarage). In 5 of them the object that is tracked is an airplane and in the other two it is a helicopter. The 7 videos are selected among 200 videos and they are the ones that are problematic in some way. All videos are recorded by a tracking camera, i.e., while an unknown tracking system is tracking the target. This unknown tracking system is called ‘Source of the Video (SoV)’. In the videos,most of the time the camera turns into the direction of the target and the target remains at a fixed position which is the center of the frame. However, there is no guarantee that the camera moves along with the target all the time since ‘source of the video’ has an unknown performance.

The helicopter videos are selected not because that they are not airplanes but helicopters release flare more than once. For such a case it is important for a tracker not to track the flare. For the flare releasing videos some changes are made in the decision logic.

The second aim of the thesis is to assess the performance of a tracker. The tracker that we used to assess is an IR video tracker that is designed to track relatively small objects in real time. This tracker is named as Real Time Tracker (RTT). RTT tracked the targets of the 7 videos that are used in this study and its tracking performance is evaluated.

In the literature, it is possible to evaluate a tracking system without ground-truth information [8, 24]. In [24], automatic performance evaluation technique which does not require ground-truth is proposed. The evaluation of tracker is done according to the changes of tracking bounding box. This method requires too much prerequisites such as motion smoothness, direction and speed consistency, no occlusion etc. Its performance degrades if the input video contains complexity. In [8], authors proposed a method which uses synthetically constructed videos as reference. These videos can be constructed in a complex way (contains occlusions, multiple targets, dramatically change of speed etc.). However, it is not possible to use real world videos in this system.

There are facilities which provide ground-truth data sets periodically. These datasets

are extracted with hand labeling [1], [2].
There are some commercial toolkits that provide ground-truth trajectory such as the Video Performance Evaluation Resource (VIPER) [3]. For VIPER the human labor increases proportionally as the complexity of videos increase.
In [11], authors propose semi-automatic method for ground-truth extraction. First, moving objects are detected by foreground segmentation algorithm and they are tracked by a tracking algorithm. Operator interferes if necessary and corrects tracks.

1.1 Outline of the Thesis

The thesis is organized as follows:
Chapter 2 explains the camera motion estimation which is crucial for our aim since the videos used throughout the thesis have no camera motion information. Camera motion estimation algorithm consists of 3 parts. In the first part all tracks of the video are computed by Tracker-1. In the second part these tracks are classified as tracks of background objects and others. In the third part the background tracks are used to estimate the camera motion.
In Chapter 3, the decision mechanism which used in order to extract the ground-truth is explained. Two trackers are implemented for this purpose. They use Interactive Multiple Model (IMM). IMM model is explained in Appendix B. The measurements of the detector are tracked by a multi target tracker, Tracker-1. Then the true track is initialized by a human operator. The tracks obtained as the output of Tracker-1 are utilized to generate the correct track. This is done by associating the track outputs of Tracker-1 to the true track whenever such an association is necessary. Camera motion estimation is used to associate in the association process. The output of the second tracker, Tracker-2, gives the raw ground-truth information. The raw ground-truth is smoothed by Kalman smoother and its output ia assumed to be the ground-truth.
Chapter 5 gives the experimental results of the videos used throughout the thesis. The evaluation of an existing tracker named as Real Time Tracker (RTT) is explained on the examples.
Chapter 6 gives the conclusions and the future work.

CHAPTER 2

CAMERA MOTION ESTIMATION

Camera motion estimation which is close to the notion of background motion estimation is a well-studied subject in the literature. The need for the estimation of the background motion is to detect the objects moving with respect to a stationary background. The earliest works assume stationary camera and estimate the stationary parts of the video frames by comparing the statistical representation of each pixel in the consecutive frames [14]. In the early works of background motion estimation with a moving camera homography is used to compensate the camera motion. However in these works camera motion is restricted to be pan, tilt and zoom or plane background approximation [18, 17]. Later the plane background approximation is relaxed to the existence of a 'dominant plane' [13, 25].

There are few studies about the background subtraction for freely moving cameras [19, 16] mostly done in the last decade. Use of multiple homography is also a popular method in these studies [21, 22, 16]. [26] uses a different methodology. It estimates the epipolar geometries induced by a moving camera. [7] generates a sparse optical flow algorithm as an initial processing stage. Then they apply a probabilistic filter in the post processing. In [14] a sparse model consisting of trajectories of salient features is built. Background is subtracted by removing trajectories that lie within the space spanned by the basis.

In this study our aim is not to work on the pixel level but to find a general motion of the background generated by the camera motion. So we used the technique of tracking the background objects.

2.1 Problem Statement

All infrared videos which are used as an input to the detector throughout the thesis were recorded while an unknown tracking system tracks the target. This unknown tracking system is called as 'source of the video (SoV)'. The motion of the camera of this system is unknown. To extract the ground-truth trajectory for the target it is important to estimate the motion of the camera. Camera motion is smooth while 'source of the video' is in the track mode but may be quite arbitrary at the other modes.

There are some possibilities for camera motion:

1) Source of the video is in the track mode and for that reason camera moves with almost constant velocity. That means target remains stable at the center of the scene but the background objects make a relative motion. Furthermore, the velocity vectors of these objects are almost same.

2) Source of the video is in the coast mode and camera makes arbitrary movements to search for the target or to center it after finding the target again.

3) Operator manipulates the motion of the camera.

Figure 2.1 given below shows the x(t) plot of the measurements taken by the Detector for a real video (t is the time or equivalently the frame number). This is an example for possibility 1. Green points correspond to the background objects which could be the clouds, the trees, handmade objects etc. while camera is making almost constant movement to track the target. Blue points are representing the true target and black points are the detections that come from another object which is moving. Note that the x positions of the true target are at almost zero since it is kept at the center of the frame during tracking.

Figure 2.2 represents the possibilities 2 and 3. The color codes of this figure are the same as the previous one. The figure shows the x(t) plot of the detections. Note that when the camera makes an abrupt motion in the interval 190-220 background objects generate parallel trajectories. However their positions change rapidly. As a consequence of this instantaneous movement, false target's position (black color) also changes rapidly but in parallel to the background objects. This is because of the negligible velocity of the moving object compared to the huge camera velocity.

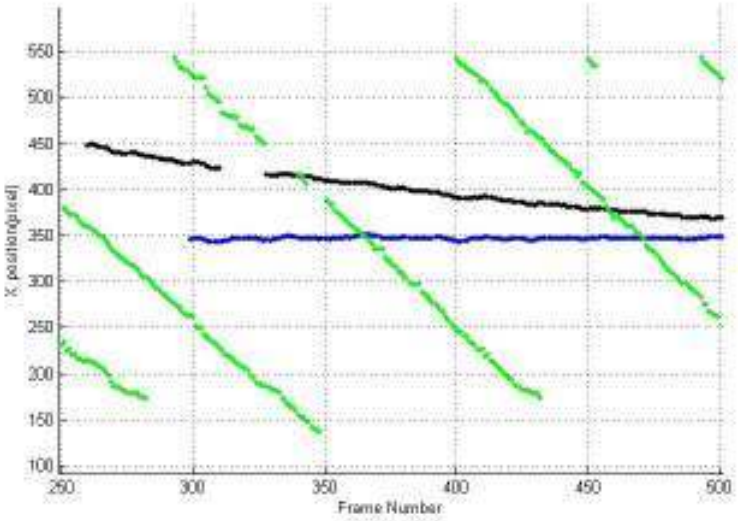


Figure 2.1: Standard Camera Motion

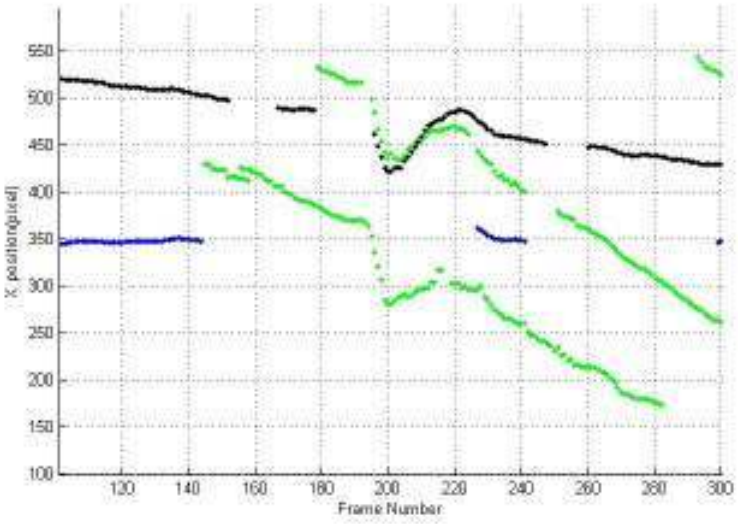


Figure 2.2: Instantaneous Camera Movement

In this study, tracks are generated by the Tracker-1 which is explained in chapter 3 via tracking the measurements of the detector. Background objects are discriminated from the moving objects by using velocities of the generated tracks. Once the tracks belonging to the background objects are found, camera motion is estimated by aggregating all background tracks.

Constant velocity model is used to track the background objects. However since all background objects have the same velocity we have used a time varying single model to track the background. Next section gives the details of the model that we used.

2.2 Camera Motion Estimation Model

Camera motion estimation part of this study consists of three consecutive blocks. In the first block all possible tracks are generated by using Tracker-1. In the second block, these tracks are classified as i) background tracks ii) tracks that belong to a moving object. The schematic diagram of these operations is given below. Since the aim is to estimate the motion of the background objects the output of the classification block is concentrated to the background tracks. The output consists of starting and ending frame numbers of each track and the measurements associated to it. Note that the classification is an off line process so the decisions are made by considering the complete trajectory of a track. After the classification is done a novel background tracking algorithm is run on the background tracks.
In the remaining part of this section we first give the details of the classification algorithm then explain the background motion estimation using the background tracks.

Figure 2.3: General Flowchart of the Camera Motion Estimation Model

2.2.1 Classification of Tracks

Classification is an off line process. The tracks generated by Tracker-1 are classified to extract the ones that are belonging to the background objects. An example of Tracker-1 output is given in the Figures 2.5 and 2.6. These figures show the x and y positions of the detections with respect to time(x(t), y(t) plots). Figures are taken from Video 1 which will be named as V1 from now on. In V1 camera motion is mostly the x direction. Therefore x position given in Figure 2.5 is more informative for the classification purposes. Various tracks are generated as the output of Tracker-1. These tracks belong to either a background object or the target or another moving object. Decisions about these categories are made by considering the velocity vectors of tracks. Classification by velocity requires first the global decision about the camera motion: Camera is tracking the true target or camera is making arbitrary abrupt motion.

Case 1: Camera makes smooth motion

For this case the assumption is: Norms of the velocity vectors of the track belonging to the target is close to zero and norms of the velocity vectors of the tracks belonging to the background objects are greater than a constant value. However, there could be other targets beside the true target. For them we made the assumption that norm of the velocity vector of tracks belonging to the false target is not equal to zero but small if its motion is in the same direction with the true target. However if the false target makes an opposite direction motion compared to the true target its velocity vector usually becomes much greater than the velocity of the background objects. These assumptions are consistent with the observations that we made on the videos that we have analysed.

Case 2: Camera makes abrupt movements

For this situation, we consider all of the tracks as background tracks. There are two reasons for this decision: first the information that a track belongs to either target or background cannot be easily extracted from the velocity since the norm of velocity vectors of both target and background are close to each other. Second: The above described fact at the same time allows to compute the background motion without distinguish the target from the background since norms of velocity vectors of both are close to each other and large.

The pseudo code of the classification block is given in Figure 2.4. Note that the algorithm classifies only the tracks with high speeds as background objects. Actually we need not to distinguish the two cases since in the second case all of tracks are considered as background tracks.

For each track generated by Tracker-1 apply the following algorithm

Input: track starting (f0) and ending (f1) frame numbers, norms of the velocity vectors in this interval, measurements used to generate this track

Output: classification of the track, if classified as background starting and ending frame numbers, measurements that generate the track.

1) Set HighVelocityFrameCount=0; LowVelocityFrameCount=0;
2) Repeat for f=f0:f1
    a. Compute the norm of the velocity vector (nv).
        i. If (nv > M) then discard this tack
        ii. If (nv > ε) then HighVelocityFrameCount= HighVelocityFrameCount+1;
        iii. If not; LowVelocityFrameCount= LowVelocityFrameCount+1;
3) end
4) If (HighVelocityFrameCount> LowVelocityFrameCount) this track is labeled as background track.

Figure 2.4: The pseudo code of the classification algorithm

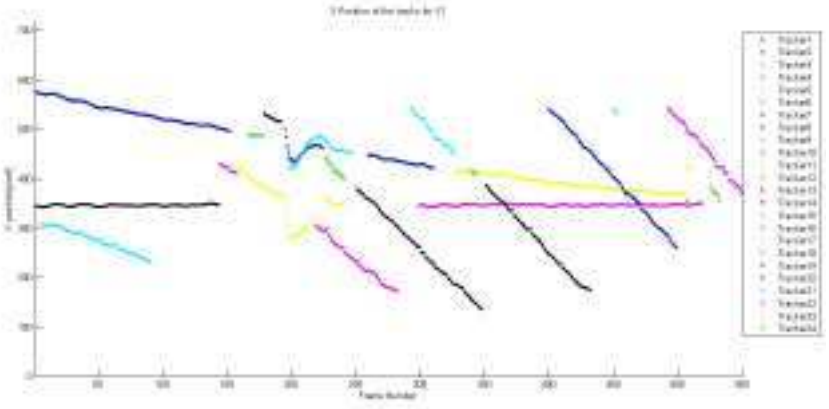


Figure 2.5: X position vs frame number for V1

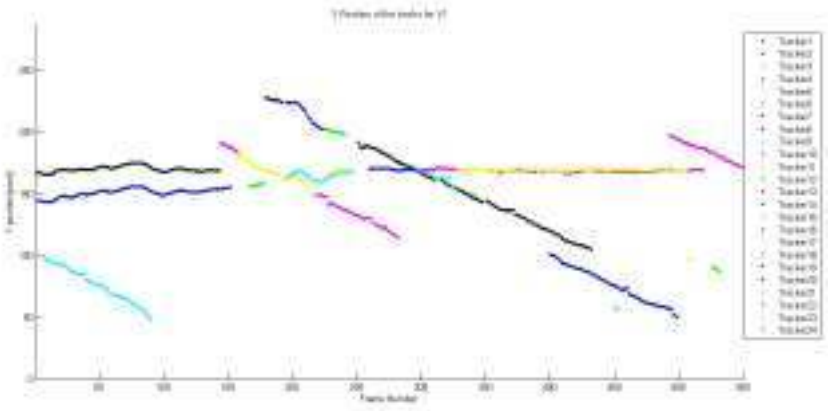


Figure 2.6: Y position vs frame number for V1

2.2.2 Background Tracking

At the end of the classification phase background tracks are obtained. In the background tracking phase they are fused to get more reliable motion estimation. Note that the aim of estimating the motion of the camera is tracking the true target. So very precise algorithms, that associates all pixels of a frame to the pixels of the previous frame are not necessary.

We have treated the background motion estimation problem as finding the position of a dummy point in the consecutive frames by tracking it. The method used here is adopted from [5]. In [5] the procedure explained here is used for group tracking.

The basic idea behind the motion estimation part is to model the motion of the background by a single constant velocity model and use the positions of the background objects that exist as measurements to update the motion at each frame. The interesting point here is that the number of objects change in time and the state vector and also the measurement vector may have different sizes at each time instant. The explanation of this novel algorithm is given below.

2.2.2.1 The Model

We assume that the camera motion, so the motion of the background objects, is smooth so they can be tracked by a constant velocity model. At the time instances

where the motion is abrupt, this model may not be able to track it however for these cases, it ends the track and after ending a track it generates a new track.

Each background object is modeled by a state vector that contains the position of the object on the image frame. The velocity of each background object is almost same if parallax effect is not taken into consideration. So the overall state vector that is modelling the background has the components of x-y positions of each background object and the V_x and V_y values of the common velocity. In addition to these our model contains the position of a dummy point. The position of the dummy point is initialized as the origin of the scene. At the end it gives the total displacement of the origin.

The state and the observation equations have the standard forms as given below.

$$x_{k+1} = Ax_k + Gw_k \tag{2.1}$$

$$y_k = Cx_k + Hv_k \tag{2.2}$$

where $w_k \sim N(0, Q)$ and $v_k \sim N(0, R)$

The compound model is time varying and $x_k, P_k, A_k, G_k, Q_k, R_k, H_k, C_k$ are modified as the number of tracks belonging to background objects is changing.

Assume at frame k we have a set of tracks, with their indexes [i, ... ,n]. The state vector of the background tracking system is defined as

$$x_k = \begin{bmatrix} {X_k}^i \\ {Y_k}^i \\ \vdots \\ {X_k}^n \\ {Y_k}^n \\ {X_k}^* \\ {Y_k}^* \\ {V_k}^{*x} \\ {V_k}^{*y} \end{bmatrix} \tag{2.3}$$

The components of the state x_k are defined as:

${X_k}^i$: X position of i^{th} track at k^{th} frame k^{th} frame

${Y_k}^i$: Y position of i^{th} track at k^{th} frame k^{th} frame

$X_k{}^*$: X position of the origin of the background objects at k^{th} frame

$Y_k{}^*$: Y position of the origin of the background objects at k^{th} frame

$V_k{}^{*x}$: The common velocity of background objects in X direction at k^{th} frame

$V_k{}^{*y}$: The common velocity of background objects in Y direction at k^{th} frame

The state vector contains the position of each object, a single velocity component $\begin{bmatrix} V_k{}^{*x} \\ V_k{}^{*y} \end{bmatrix}$ and a part that is shown by $\begin{bmatrix} X_k{}^{*x} \\ X_k{}^{*y} \end{bmatrix}$ which is the position of a dummy point. The initial position of the dummy point is the origin of the frame so it corresponds to the overall position change estimate of the origin.

The size of the A matrix depends on the number 'n'. Example of A matrices are given in (2.4) and (2.5). (2.4) represents the case of two background objects, while (2.5) is the model for 3 background objects.

G matrix is given in (2.6) and (2.7) for two and three background tracks respectively. C,Q,R,H matrices are explained in (2.8), (2.9), (2.10), (2.11) respectively.

$$A = \begin{bmatrix} I_2 & 0_2 & 0_2 & TI_2 \\ 0_2 & I_2 & 0_2 & TI_2 \\ 0_2 & 0_2 & I_2 & TI_2 \\ 0_2 & 0_2 & 0_2 & I_2 \end{bmatrix} \tag{2.4}$$

$$A = \begin{bmatrix} I_2 & 0_2 & 0_2 & 0_2 & TI_2 \\ 0_2 & I_2 & 0_2 & 0_2 & TI_2 \\ 0_2 & 0_2 & I_2 & 0_2 & TI_2 \\ 0_2 & 0_2 & 0_2 & I_2 & TI_2 \\ 0_2 & 0_2 & 0_2 & 0_2 & I_2 \end{bmatrix} \tag{2.5}$$

$$G = \begin{bmatrix} (T^2/2)I_2 \\ (T^2/2)I_2 \\ (T^2/2)I_2 \\ I_2 \end{bmatrix} \tag{2.6}$$

$$G = \begin{bmatrix} (T^2/2)I_2 \\ (T^2/2)I_2 \\ (T^2/2)I_2 \\ (T^2/2)I_2 \\ I_2 \end{bmatrix} \tag{2.7}$$

$$C = \begin{bmatrix} I_{(2*trackCount)\times(2*trackCount)} & 0_{(2*trackCount)\times(4)} \end{bmatrix} \tag{2.8}$$

$$Q = \begin{bmatrix} {\sigma_x}^2 & 0 \\ 0 & {\sigma_y}^2 \end{bmatrix} \tag{2.9}$$

$$R = \sigma^2 \begin{bmatrix} I \end{bmatrix}_{(2*trackCount)\times(2*trackCount)} \tag{2.10}$$

$$H = \begin{bmatrix} I \end{bmatrix}_{(2*trackCount)\times(2*trackCount)} \tag{2.11}$$

2.2.2.2 Initialization of Camera Motion Estimation Model

If one or more tracks are classified as tracks that belong to the background objects, camera motion estimation model is initialized. Assume that the i^{th} and j^{th} tracks are decided as tracks that belong to the background objects at the k^{th} frame. At this frame state vectors of these background objects are already estimated and the corresponding covariance matrices are known since they are tracks. Initialization parameters of the background motion are selected as:

- $x_{k|k}$:

$\begin{bmatrix} {X_k}^i \\ {Y_k}^i \end{bmatrix}$ part of the x_k^i is the measurement of the i^{th} background obkect. Same rule is applied for other tracks belonging to background objects.
$\begin{bmatrix} {X_k}^* \\ {Y_k}^* \end{bmatrix}$ part of the $x_{k|k}^i$ is selected as $\begin{bmatrix} 360 \\ 144 \end{bmatrix}$ which is the center of the scene if no

initialization is started before. Otherwise, the values where the latest camera motion estimation model ended are taken for initialization.

${V_k}^{*x}$ part of the $x_{k|k}$ is selected as in (2.12). Same calculation is applied for ${V_k}^{*y}$ as well.

$$ {V_k}^{*x} = \frac{({V_k}^{ix}({\sigma^i_{V_x}}^2)_k)^{-1} + ({V_k}^{jx}({\sigma^j_{V_x}}^2)_k)^{-1}}{(({\sigma^i_{V_x}}^2)_k)^{-1} + (({\sigma^j_{V_x}}^2)_k)^{-1}} \tag{2.12} $$

where

$({\sigma^i_{V_x}}^2)_k$: variance of V_x for i^{th} track

${V_k}^{ix}$: Velocity of i^{th} track in x direction at k^{th} frame

- $P_{k|k}$: $\begin{bmatrix} a & e & 0_2 & 0_2 \\ e^T & b & 0_2 & 0_2 \\ 0_2 & 0_2 & 0_2 & 0_2 \\ 0_2 & 0_2 & 0_2 & c \end{bmatrix}$

where

a : $\begin{bmatrix} ({\sigma^i_X}^2)_k & 0 \\ 0 & ({\sigma^i_Y}^2)_k \end{bmatrix}$

b : $\begin{bmatrix} ({\sigma^j_X}^2)_k & 0 \\ 0 & ({\sigma^j_Y}^2)_k \end{bmatrix}$

e : $\begin{bmatrix} ({\sigma^{ij}_X}^2)_k & 0 \\ 0 & ({\sigma^{ij}_Y}^2)_k \end{bmatrix}$

c : $\begin{bmatrix} (((\sigma^i_{V_x}{}^2)_k)^{-1} + ((\sigma^j_{V_x}{}^2)_k)^{-1})^{-1} & 0 \\ 0 & (((\sigma^i_{V_y}{}^2)_k)^{-1} + ((\sigma^j_{V_y}{}^2)_k)^{-1})^{-1} \end{bmatrix}$

$({\sigma^i_X}^2)_k$: Covariance of i^{th} track for x position at k^{th} frame

$({\sigma^{ij}_X}^2)_k$: Cross-Covariance between i^{th} and j^{th} track for x position at k^{th} frame

2.2.2.3 Tracking

Kalman filter is used for tracking the background. Kalman filter equations are given in Appendix A. The difficulty with this model is its unconventional structure. For each frame a new background track may enter to the model or an old one may leave it. Both causes dimension changes in the state equation and in the observation equation. Elimination of ended tracks is easy: one can eliminate the states corresponding to the ended tracks from the state vector. This reduces the dimension of the state vector by two (position of the object on the scene). . Similarly measurement vector is also reduced by two since the position measurements of this background object no more exists. These reductions in the state and the measurement vectors also change the sizes of the related matrices. New matrices can be generated easily by eliminating the parts of them corresponding to the eliminated parts of the vectors.

Adding new parts to the state vector is more difficult. If a new background track enters into the scene its own estimations are appended to the state estimate and its covariance. All matrices are changed accordingly. Example of these matrices are given in Section 2.2.2.1.

2.3 Generating the Reference Data

There is a need to generate a reference data in order to measure the performance of the camera motion estimation model. Hand labeling is applied to V1 for this purpose. Hand labeling is the labeling of a background point throughout the video in a consistent manner. For example; Figure 2.7 is the 100. frame of V1 and red circle is the point where the operator marks a cornet of a cloud as a background object. Operator marks the same object at frame 150 as can be seen in Figure 2.8. If the background object that is labeled leaves the scene then operator selects another point as reference point and continues with that one.

The hand labeled x and y positions of the selected points for V1 are shown in the Figures 2.9 and 2.10. Three red lines means 3 different reference points are selected by the user. A Kalman filter is used to track the positions to further smooth the trajectory. Kalman filter is started for each 3 reference points. The velocity components of state

vector are considered as reference velocities for camera motion estimation model.

Figure 2.7: 100. frame of V1

Figure 2.8: 150. frame of V1

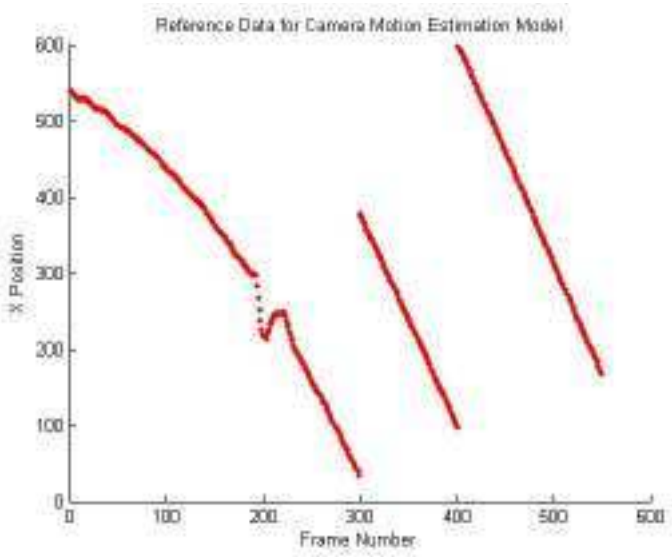


Figure 2.9: Reference X Positions for Camera Motion Estimation Model

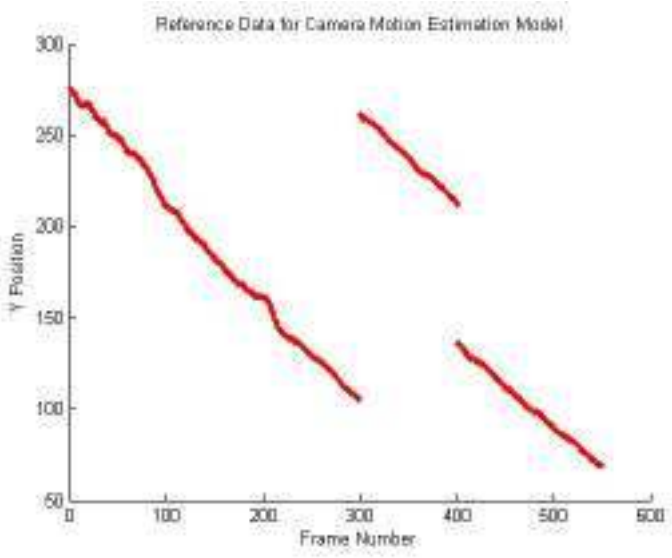


Figure 2.10: Reference Y Positions for Camera Motion Estimation Model

2.4 Testing the Performance of the Camera Motion Estimation Model

Camera motion estimation model is applied to V1 and V2 to evaluate its performance. all videos used throughout the thesis. We have selected these two videos to test the camera motion estimation since they both contain both smooth camera action as well

as abrupt camera motion. Since the position of a background object is independent of the position of the other but their velocities are similar we compare the velocities.

2.4.1 Results of Video-1 (V1)

V1 is a challenging video which contains arbitrary camera motion. Output of Tracker-1 is given in Figure 3.4 and Figure 3.5 in Chapter 3. Figure 2.11 and 2.12 represents the comparison between the reference data and camera motion estimation model results. Blue line corresponds to the velocity of the camera obtained by the camera motion estimation model proposed here and red line corresponds to the reference data. The positions where blue line doesn't exist are the result of non-existence of background tracks.

The figures show that the estimation of the background motion is quite satisfactory in the sense that they give very similar velocities especially when we an abrupt camera motion exits. The difference is less than 1 pixel/frame almost everywhere including the abrupt camera motion region.

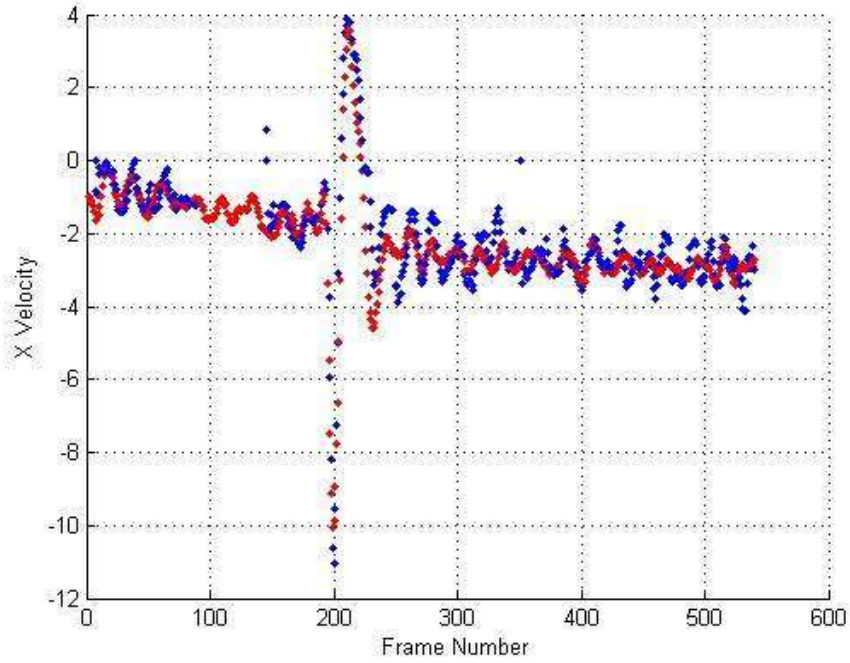


Figure 2.11: Comparison of X velocity between camera motion estimation model and reference data

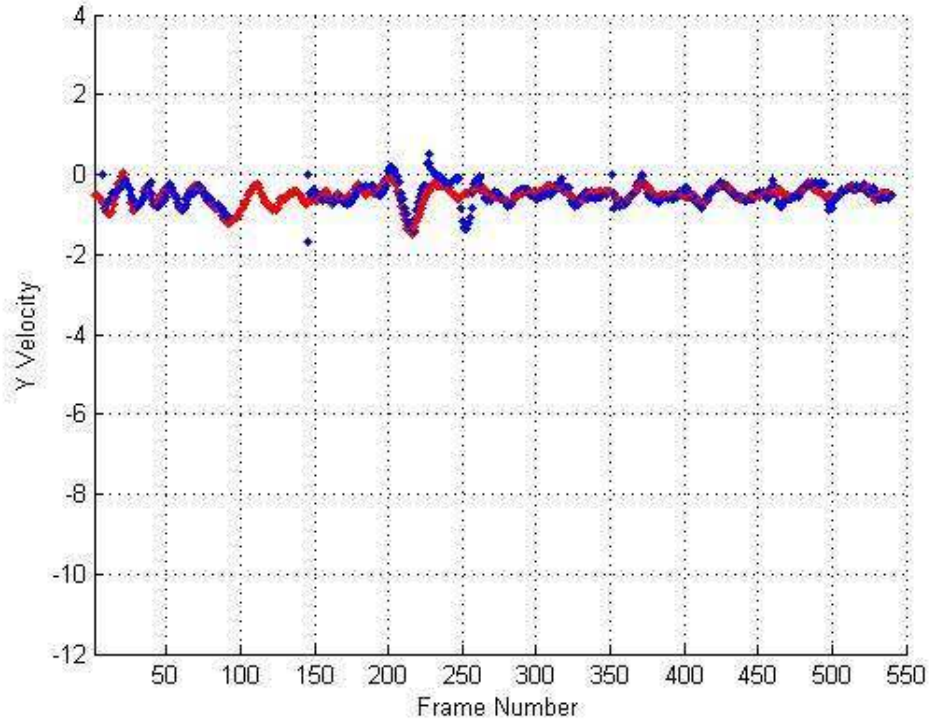


Figure 2.12: Comparison of Y velocity between camera motion estimation model and reference data

2.4.2 Results of Video-2 (V2)

V2 is another challenging video which contains abrupt motion. It is different than V1. Unlike V1, after the abrupt motion of the camera the SoV places another moving object into the center of the scene. Figure 2.13 and 2.14 represents the comparison of the velocities between the reference data and the camera motion estimation model output. Blue line corresponds to the proposed model and red line corresponds to the reference data. The positions where blue line doesn't exist are the result of non-existence of background tracks. The results are similar to the results obtained for V1 and less than 1 pixel/frame in most of the cases.

In both of the experiments the background extraction model seems to generate velocities with larger variances. This can be explained the relatively worse performance of the Kalman filter when the corresponding state is not directly measurable, i.e., the velocity for this case.

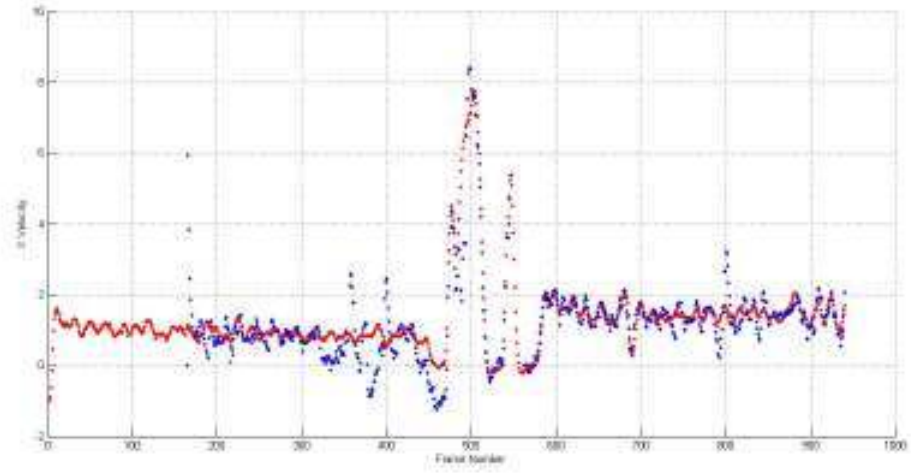


Figure 2.13: Comparison of X velocity between camera motion estimation model and reference data

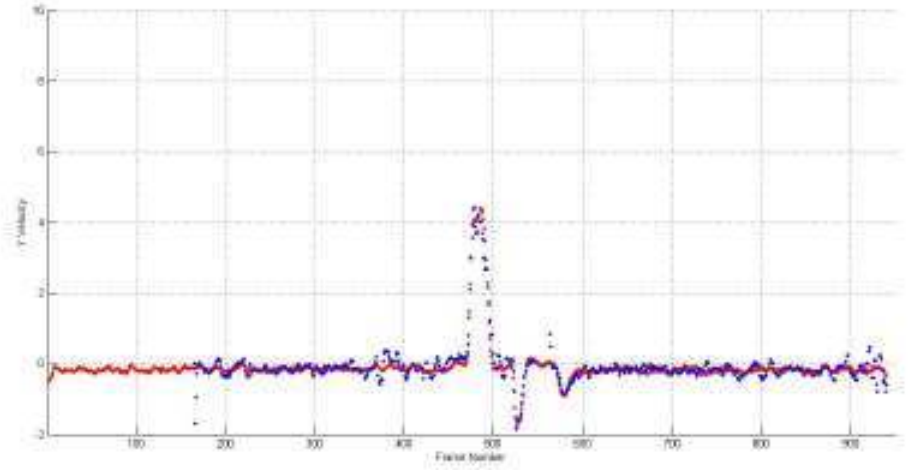


Figure 2.14: Comparison of Y velocity between camera motion estimation model and reference data

CHAPTER 3

DECISION MECHANISM

The aim of this chapter is to build a decision mechanism which extracts the ground-truth trajectory for any video. In this study we have concentrated on the air vehicles. There are two different types of videos that we have worked on. The first type is an air vehicle and the other is an air vehicle that releases flare. The second one is named as 'spawning target'. We assume that the user of the program gives this information as an input to the Ground-Truth Extraction System. Although there are two different approaches for the two different cases the sub-algorithms are not much different. Unless otherwise specified both algorithms use the same methods. Measurements of the detector are used for constructing the decision mechanism.

3.1 Single Air Vehicle

Two trackers are used for this purpose. The complete information about trackers are given in appendix B but some of details are mentioned in this section. The first tracker which is named "Tracker-1" uses measurements of the detector to track multiple targets. Tracker-1 is an on line tracker. It uses Global Nearest Neighbor (GNN) method for the association of the measurements to the tracks. Tracker-1 generates a set of tracks that some belong to the true target. A continuity of the true target trajectory is not expected. The second tracker which is named "Tracker-2" is a single target tracker. Tracker-2 is an off line tracker which uses the future information to make associations. Tracker-1 eliminates some of the measurements and associates the remaining ones to some tracks. These measurements are used by Tracker-2. Tracker-2 tracks only one target for which the related measurements are selected by the deci-

sion mechanism. The track generated by Tracker-2 starts at the beginning of the video stream and ends at the end even if the target is occluded in some regions.

The steps of the decision mechanism are given below. Each step of the decision mechanism is explained on an example video, Video-1 (V1) which is a typical example of an airplane.

1) Measurements of the detector are tracked by Tracker-1. Associations are done by GNN.

2) Camera motion is estimated for the whole video. The output of the camera motion estimation system is the velocity of the background and the position of the dummy point.

3)The user of the program gives the position of the target at the first frame. Therefore the decision mechanism selects the true track among the tracks of Tracker-1.

4) If the true track ends there are two possibilities for the rest of the video:

a) Try to associate the tracks of Tracker-1 with the 'true track'. If an association occurs continue with the measurements of the associated track.

b) If Tracker-1 generates no tracks or no association occurs between the tracks of Tracker-1 and the true track, no measurement update is applied to the true track. However, the covariance matrices are restricted to a nominal value in long intervals.

5)Repeat step 4 until the end of the video.

6) Mark the target trajectory at all frames as either 'occluded' or 'tracked' frame.

7) Apply Kalman smoother to the raw ground-truth generated by Tracker-2 at the tracked regions.

4b is the step where track associations are done. An important point here is that the camera motion must be taken into consideration during the associations. If the camera motion in this interval is almost the same as in the tracked interval we assume that the velocity of the target is equal to the camera velocity. Otherwise, the velocity difference between the camera and the target should be added to the position prediction of the target. The time derivative of the dummy point gives the velocity of the target. If there is no abrupt motion the x position or the y position changes almost continuously at each frame. A huge gap between consecutive positions of the dummy point indicates an arbitrary camera motion. A detailed explanation of step 4 is given by the pseudo code in Figure 3.1. Assume that the tracking of Tracker-2 is stopped at time n due to non-existing data in the interval [n,n+k]. Assume that all the tracks

of Tracker-1 are available. Track availability is the availability of the measurements that generate the tracks. Furthermore assume that each track of Tracker-1 is labeled as ‘background track’ or ‘not background track’.

Inputs:

i) The last filtered value of the target's track and the covariance of its state

ii) ii) All the tracks generated by tracker-1 and their associated measurements (back ground motion is subtracted) together with their tags of belonging or not belonging to the background motion.

Output: The identity of the tracker-1 track that is associated to the target's track.

1) If the measurement of a track generated by tracker-1 is in the gate of the target's track and if it is not a background track then associate this track with the target's track. If there are more than one association, nearest one is selected as the associated track.
2) If a tracker-1 track is associated with the target's track than return back to usual tracking operation with the measurements of the associated track.
3) If at time t no track is associated with the target's track than increase t by 1, apply no measurement update step. If the gate size is larger than a pre specified value keep it fixed equal to the pre specified value.

Figure 3.1: Pseudo Code of Step 4

Example:

The x(t) and y(t) plots of measurements of the detector for video-1 (V1) are given in Figures 3.2 and 3.3 respectively. The x(t) and y(t) plots of the tracks generated by Tracker-1 are given in Figures 3.4 and 3.5 respectively.

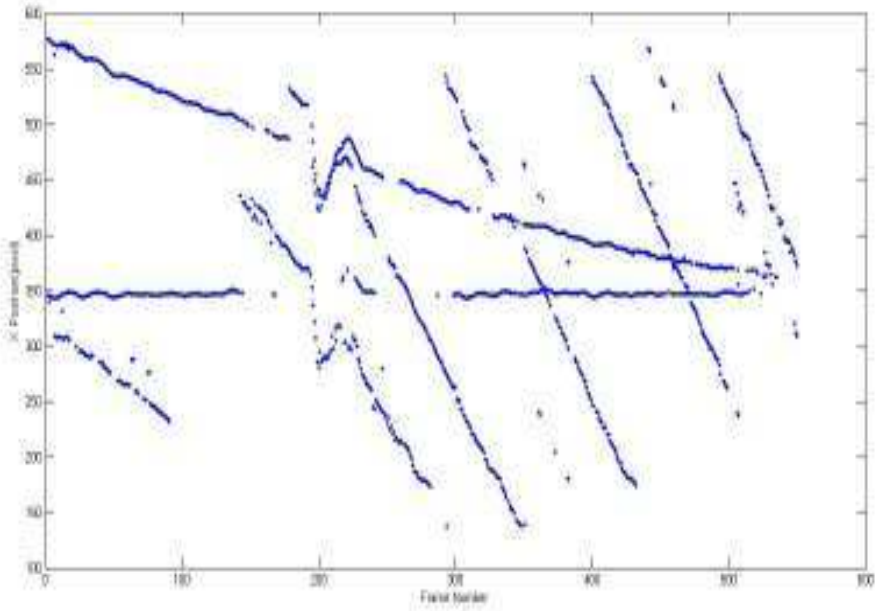


Figure 3.2: X(t) measurements of the detector for V1

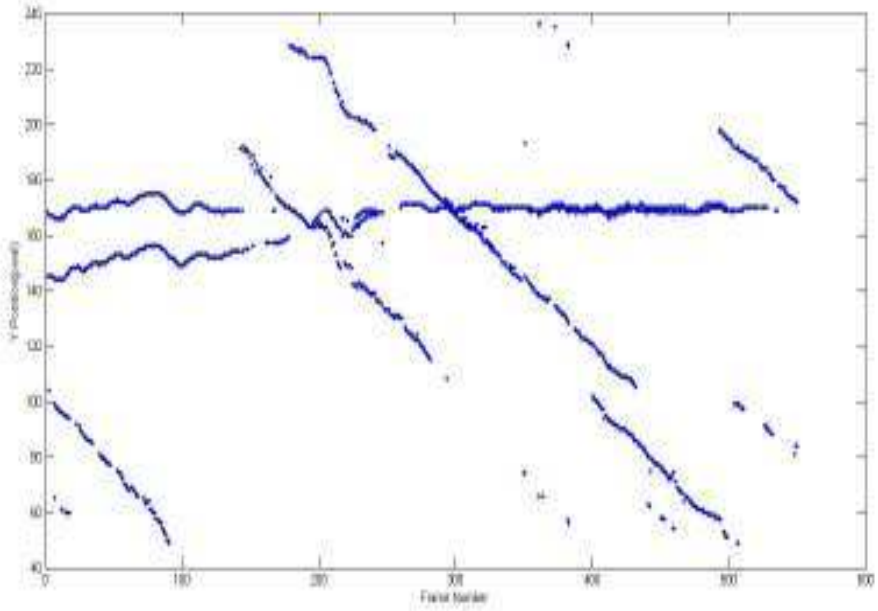


Figure 3.3: Y(t) measurements of the detector for V1

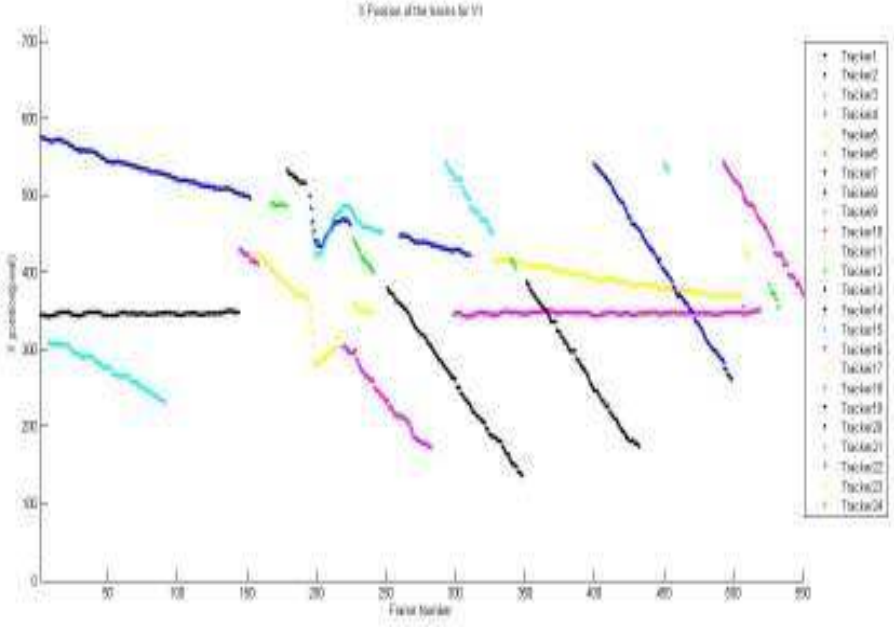


Figure 3.4: X(t) output of Tracker-1 for V1

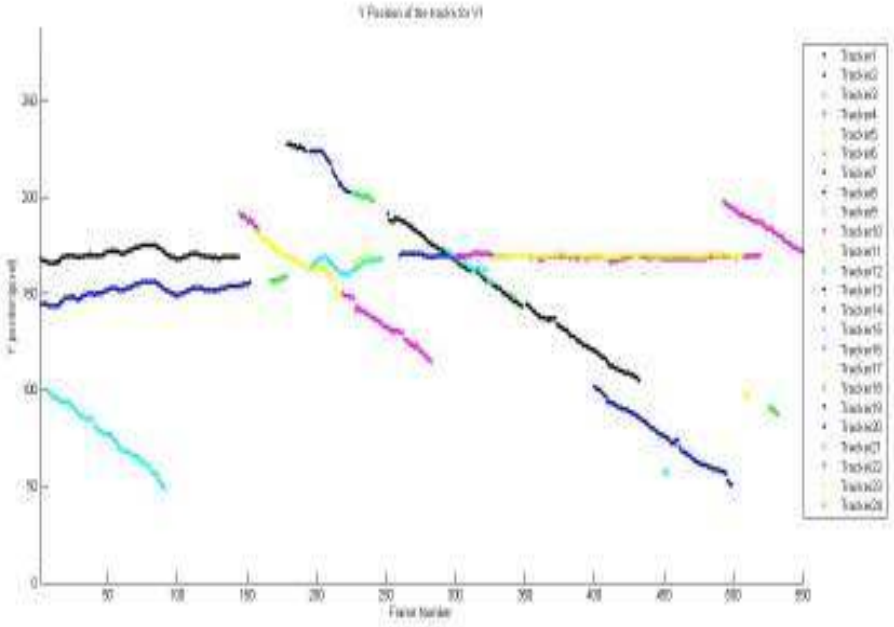


Figure 3.5: Y(t) output of Tracker-1 for V1

The x and y positions of dummy point are given in Figures 3.6 and 3.7

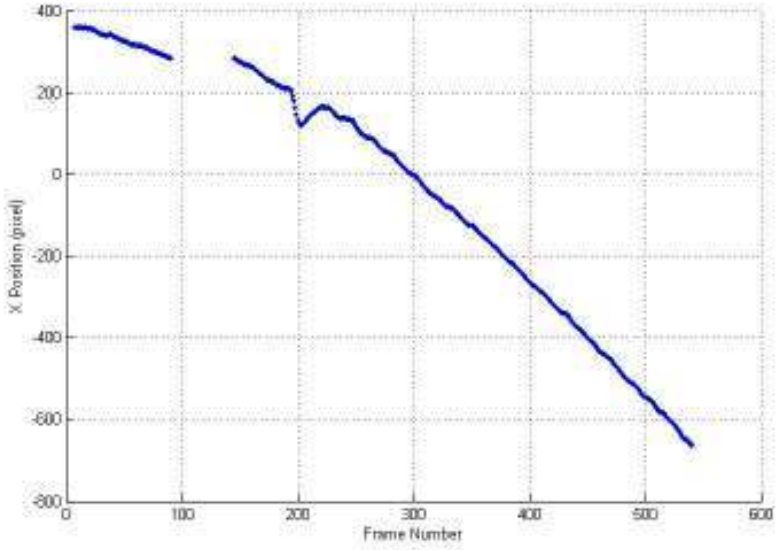


Figure 3.6: X Positions of the dummy point for V1

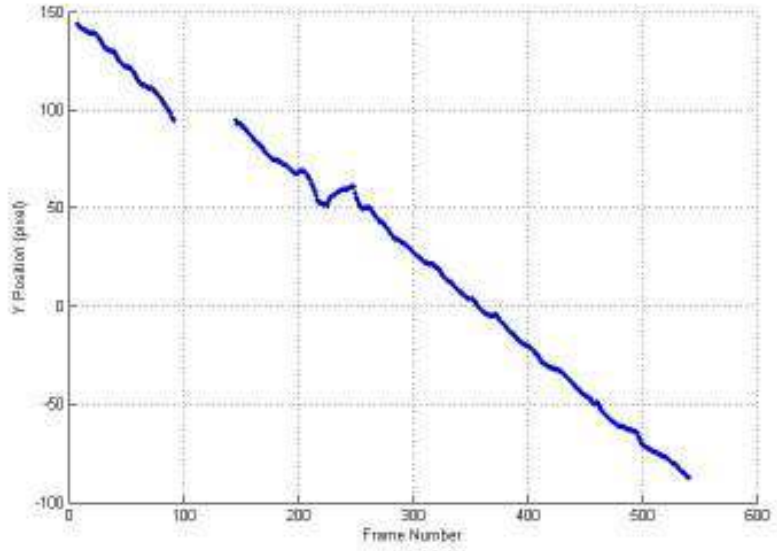


Figure 3.7: Y Positions of the dummy point for V1

Figure 3.8 gives the comparison between smoothed ground-truth and raw ground-truth for V1. Cyan color represents the smoothed ground-truth and black color represents the raw ground-truth. Ground-truth is compared with hand labeled data for 400 frames and the rms error of the position is calculated as 1,3084 pixel.

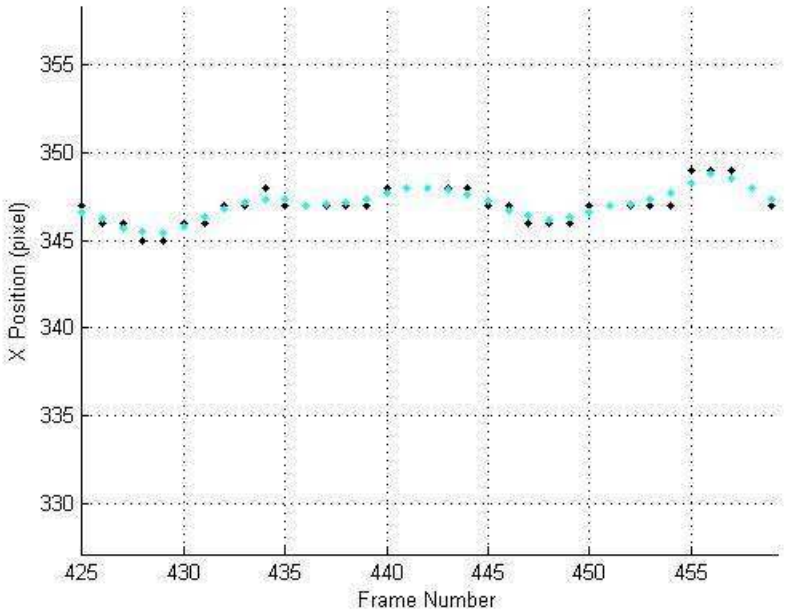


Figure 3.8: Raw ground-truth vs ground-truth

3.2 Spawning Air Vehicle

The aim of releasing flare for an air vehicle is a well-known ECM (Electronic Counter Measure) technique. The system that realized in this study is capable of discriminating flare from the true target. However the system requires the information that the video is a 'flare video'. Although flare introduces some differences to the algorithm, the generated algorithms are still almost same. For single air vehicle type, it is assumed that a track generated from Tracker-1 belongs either to the target or another moving object or a background object. However, for the spawning target videos due to flare release tracks generated by Tracker-1 cannot be trusted. A track could belong to the true target at the beginning but it could belong to the flare after the releasing process. It is essential to find the moment when flare is released by the helicopter. Then, the track belonging to the target must be determined. Figure 3.9 shows the y positions of the tracks of V6 generated by Tracker-1 with respect to time. The two time instances that flare is released by the helicopter are circled in the Figure 3.9. There are two possibilities for flares.

1) If SoV continues to track true target, y position of flares always increases (remember that 'y=0' means top of the scene) due to the gravity. The speeds of background objects and moving objects don't change. The red circled area labeled as '1' in Figure 3.9 is a typical example of this case.
2) If SoV starts to track the flare, the speed of both the target and background seem to be increased dramatically. In addition, the position of flare is almost constant. In this case, the y position of target decreases (it gets closer to the ground). The red circled area labeled as '2' in Figure 3.9 is a typical example of this case.

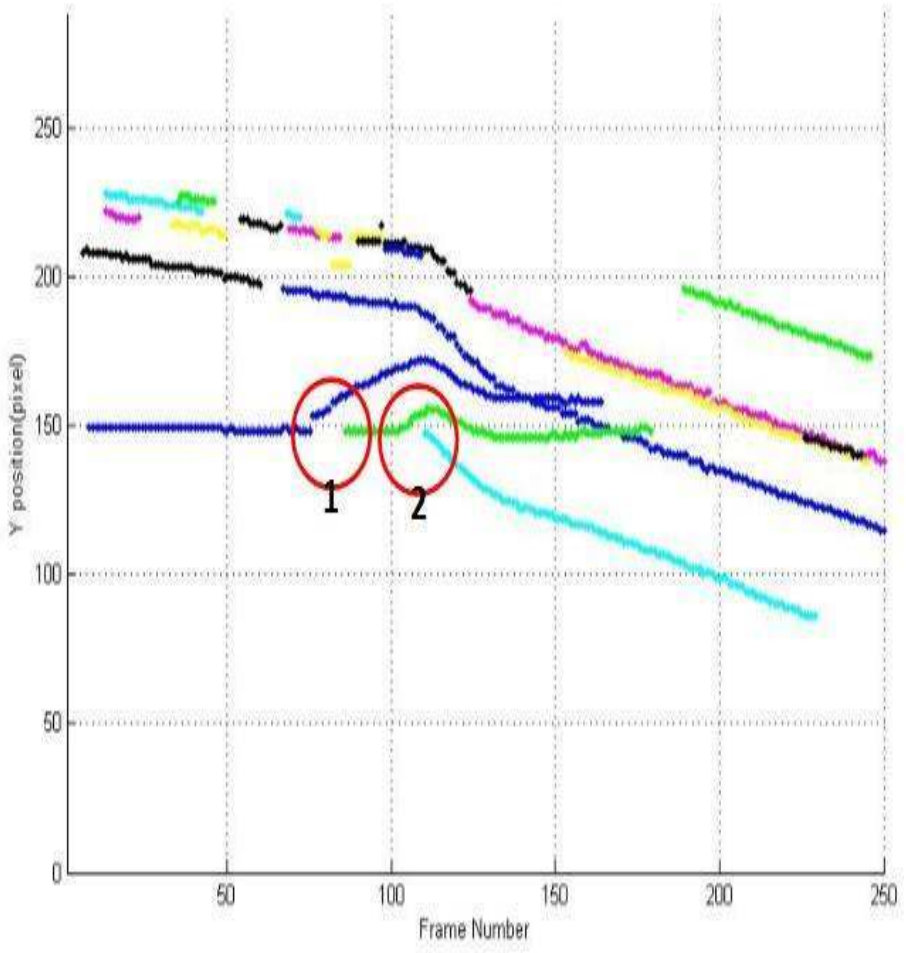


Figure 3.9: Y Positions of the Tracks Generated by Tracker-1

The extraction of flare release moments are explained in the pseudo code given in 3.10.

```
Parameters: frameJump=20; currentFrame;

If (there is a track which has lower y position than true track between the frames
currentFrame and currentFrame+frameJump )

    {

    If (norm of the velocity vector of other tracks changed considerably)

        {Make the track which has lower y position true track} //Possibility 2 occurs

    Else

        {

            If (Norm of the velocity of the true track > ε) //Possibility 1 occurs

            {Make the track which has lower y position true track}

            Else

            { Make measurement update for frameJump frames}

        }

    }

Else

    { Make measurement update for frameJump frames}
```

Figure 3.10: Pseudo Code of Finding Flare Release Moments

CHAPTER 4

EVALUATION OF A TRACKER

The extracted ground-truth trajectory is used for the evaluation of any tracking system. In this chapter we first give the structure of a typical tracker. Then we explain the mistakes that a tracker may do, finally explain the output of our evaluation program.

4.1 A Generic Model of a Tracker

Trackers usually have 3 modes: tracking mode, coast mode and track lost mode. The flow chart of a typical tracker (TT) is given in Figure 4.1.

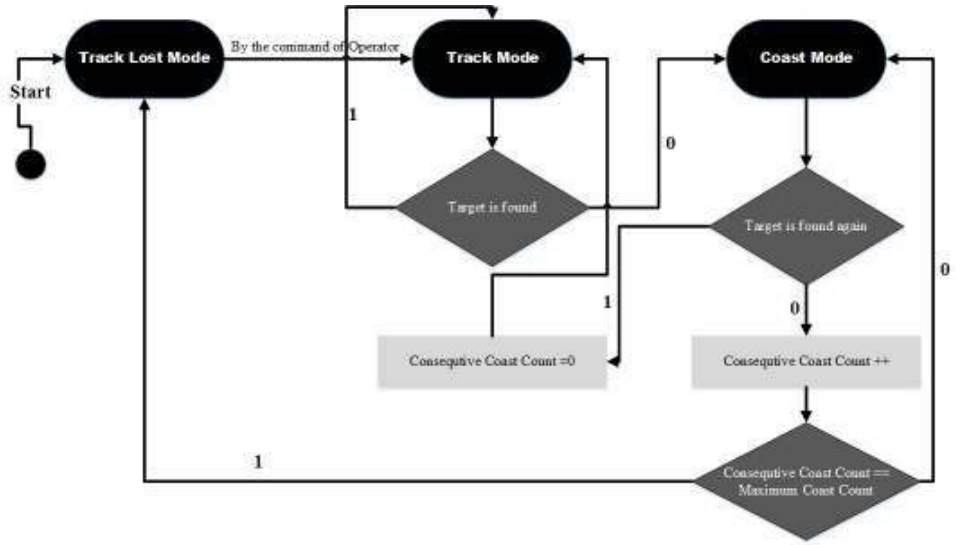


Figure 4.1: TT Mode Transition Diagram

We assume that the initialization of the track is done correctly by some means. If TT

loses the target it enters into the coast mode. In the coast mode, TT searches for the target again. When a target is found, TT returns back to the track mode. However there is no guarantee that the new target is the same as that TT tracks in the previous track mode. There is a time constraint for the coast mode that is called the maximum coast count. Maximum coast count is the total number of consecutive frames allowed in the coast mode. If the target is found within maximum coast count interval TT enters back into the track mode. Otherwise a typical tracker goes to the track lost mode and it needs to be initialized by the user again.

4.2 Possible Modes of the Output of a Typical Tracker

A tracker system makes basically two types of mistakes: wrong association and imprecise tracking. The second type has a meaning only if the track generated belongs to the true target. In this thesis the precision of the so called Ground Truth is obtained by applying Kalman smoother to the parts of the trajectory that belongs to the true target. The first type of the mistakes that a tracker may do may be listed as: it may jump from one track to another, it may lose the track, or it may generate a track although no track exists. Based on these observations we have defined the 'modes of the output of a tracker' to evaluate it. In each frame, the output could be in various modes.
The mode definition needs the evaluation of the scene. We classify the possible scenes as follows.

1. Desired object is in the scene,
2. Desired object is not in the scene temporarily (occluded or by some other reason),
3. Desired object leaves the scene.

Besides that there may be some other moving objects in the scene in all of the above cases. A typical tracker may make false or true decisions, during tracking. We may classify the track decisions as:
T1. True track: There is a track and it corresponds to the true target.
T2. False track: Target is not in the scene but the system tracks something else as the true target.

T3. Missed track: target is in the scene but the tracker gives no track.

T4. False target track: Target is in the scene but the system tracks something else.

T5. Occlusion/no target: Target is not in the scene; hence the tracker gives no measurement. Figure 4.2 illustrates the tracking modes of TT. This figure is a simulation. Blue line corresponds to the true target that TT tries to track and black line corresponds to the false target (another moving object). The expected mode is 'true track' for the area specified as 'T1' because measurements shown by red color and true target position are same. Area 'T5' illustrates the mode 'occlusion' since the true target is occluded by clouds and there is no measurement taken from tracker. Area 'T3','T4' and 'T2' belongs to 'missed track', 'false target track' and 'false track' respectively.

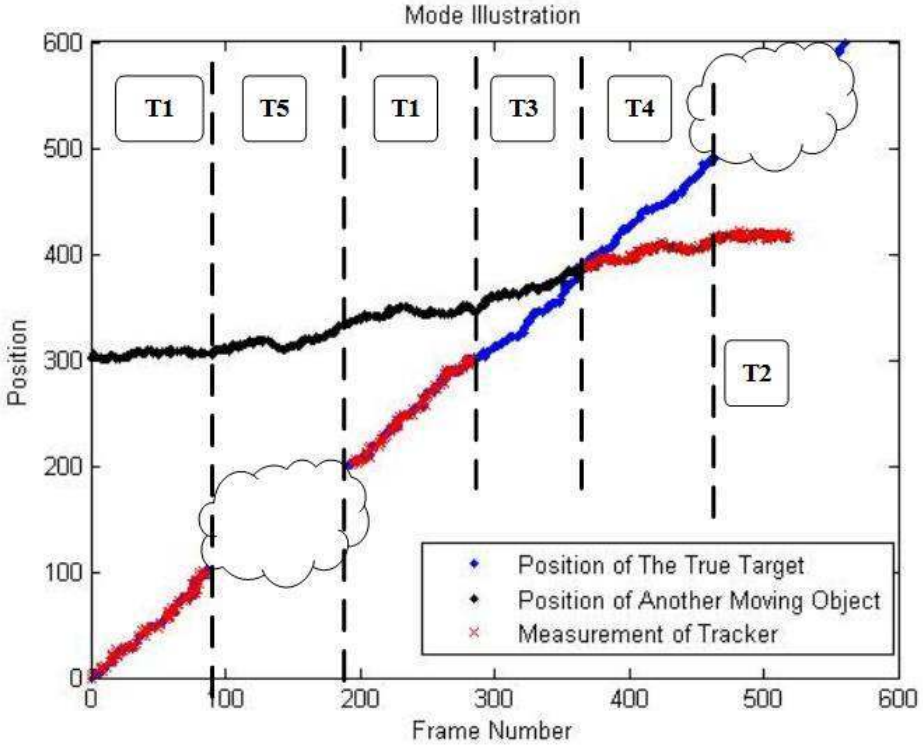
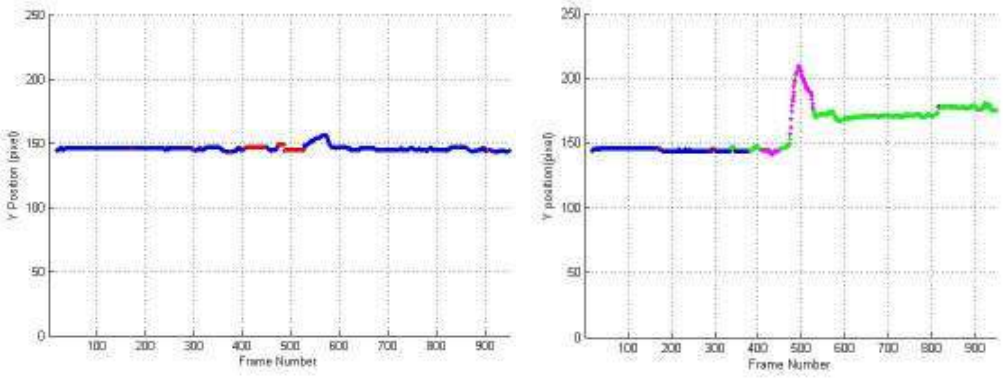


Figure 4.2: Sample Mode Transition Scenario

4.3 Output of the Evaluator

The output of the evaluator gives the frame numbers where a tracker is one of the output modes explained above. The presentations of the results are in the graphical

form. The x(t) and y(t) plots of the target after evaluations are given in the Chapter 5 for all example videos. The graphs given there are color coded. Different colors show different errors made by the tracker under investigation. We include here the result obtained for Video-2 as an example.



(a) Y Positions of the RTT

(b) Y Positions of the output of GTES

Figure 4.3: Output of RTT vs Output of GTES for Y Positions

Blue color in Figure 4.3b indicates that RTT track the correct target, magenta shows the occlusion interval for the target, green region indicates that RTT tracks something although the true target is not in the scene.

CHAPTER 5

EXPERIMENTAL RESULTS

The aim of the study is to obtain i) the true track of a target and also ii) to evaluate a given tracker by comparing its output with the output of the GTES. The performance of the system is examined on 7 videos. All videos satisfy the requirements of the system. They are recorded by a tracking IR camera. In 5 of them an airplane is the target and in two videos the target is a helicopter that is releasing flare. The videos are selected among 200 videos as the ones that create problems during tracking.

The system is first applied to all 7 videos to obtain the 'ground truth'. GTES gives the target position as the Kalman predictions on the occluded regions. It also indicates that the region is an occlusion region. The correctness of the occlusion regions are checked by human. No error is observed on all 7 videos. Also no obvious huge error is observed on the target positions of all sequences. To see the evaluation part of the GTES we have used an available tracker that is aiming to track the object in real time. We call this tracker the Real Time Tracker (RTT).

RTT makes several decision mistakes as classified in Section 4.2. The system generated in this study must give an output that shows these mistakes. Furthermore it must also show the difference between the trajectories generated by RTT and smoothed trajectory of GTES for the time intervals for which the RTT's decisions are correct.

The system GTES is applied to each video. Experimental results for video-1 are already given in the decision mechanism chapter. Experimental results of other videos are given in this chapter. The color codes for GTES and RTT are given in Tables 5.1 and 5.2 respectively. The definitions of 'True Track', 'Missed Track' e tc. are given in chapter 4.

Table 5.1: Color Code for the evaluation

Colour	**Mode**
Blue	True Track
Red	Missed Track
Black	False Target Track
Magenta	Occlusion
Green	False Track

Table 5.2: Color code for the modes of RTT

Colour	**Mode**
Blue	Track
Red	Coast

5.1 Experimental Results of Video-2

Video-2 is the most challenging video among all the videos. It contains two targets that are air planes. False target enters into the scene about at 300^{th} frame. Two targets become occluded by trees at 411^{th} frame and camera makes arbitrary motion during the occlusion interval. The true target remains occluded until the end of the video. But the false target becomes visible at 500^{th} frame and remains visible until the end of the video. RTT tracks false target beginning from the 520^{th} frame until the end of the video. There are so many trees in the scene throughout V2 and this causes many detections. The measurements of the detector are given in the Figures 5.1 and 5.2.
The measurements of the detector are tracked by Tracker-1 throughout the whole video. The track results are given in the Figures 5.3 and 5.4. In these figures, each color represents a different track.
Tracker-2 is applied to the tracks generated by Tracker-1. The output of Tracker-2 is given in the Figures 5.5 and 5.6 together with the output of RTT. The comparison of the two outputs shows the performance of RTT. Examination of V2 and its RTT output shows that the true target disappears at 411^{th} frame. RTT enters into the coast mode at this frame. It re-enters the track mode at 454^{th} frame for a short time, goes to the coast mode again at 474^{th} frame returns back to the track mode and stays there until the end of the video. However, It tracks the false target which is the second

airplane in the last 420 frames.

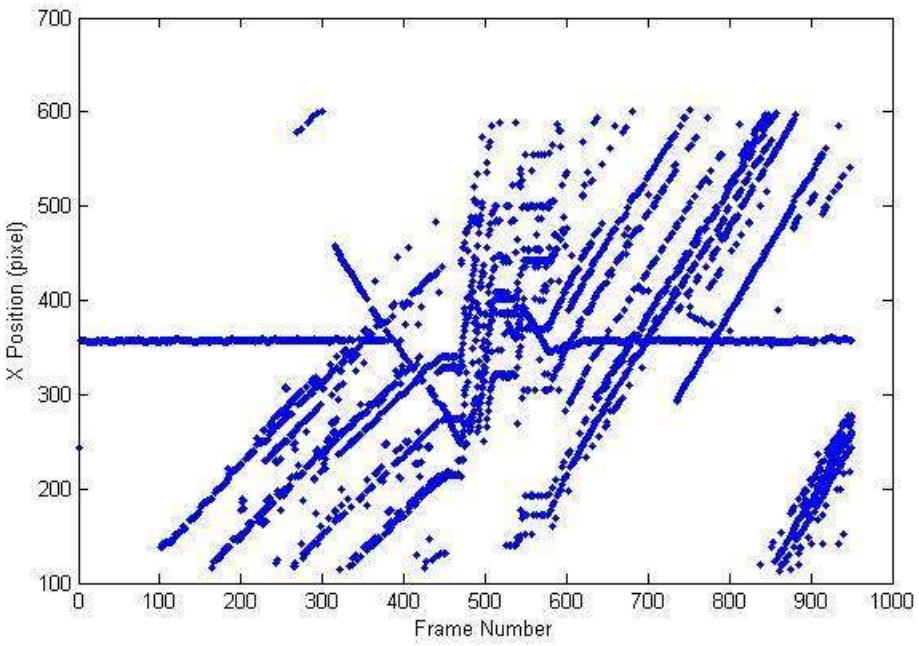


Figure 5.1: X Positions of the Measurements of the Detector for Video-2

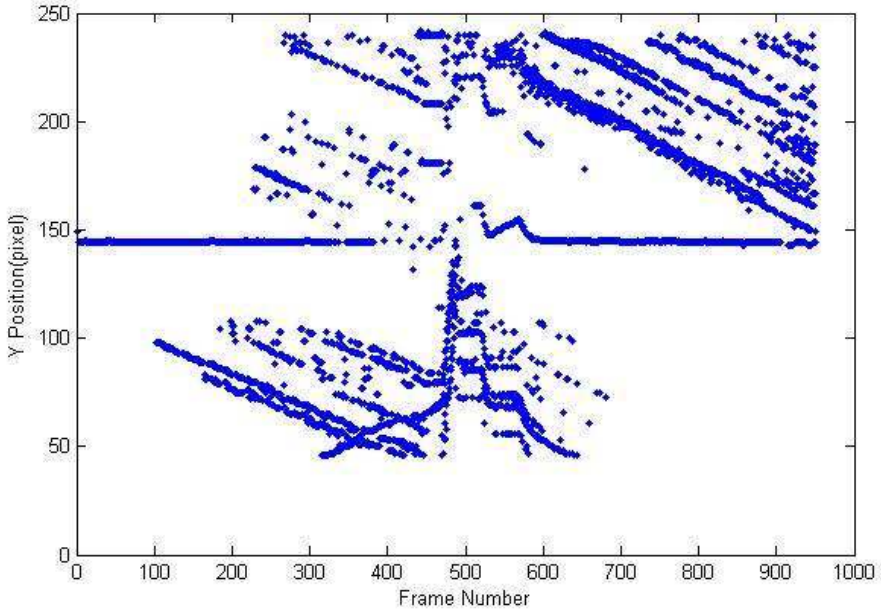


Figure 5.2: Y Positions of the Measurements of the Detector for Video-2

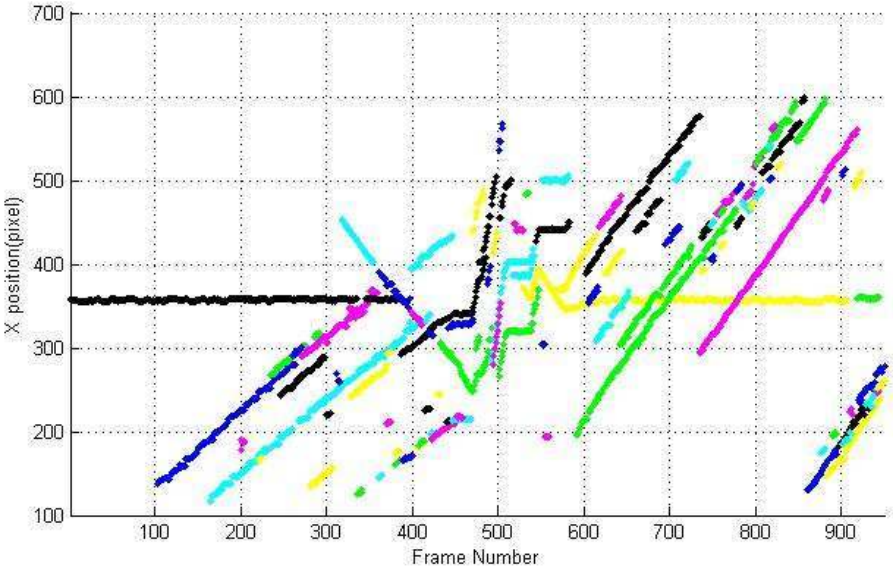


Figure 5.3: X Positions of the Tracks Obtained by Tracker-1

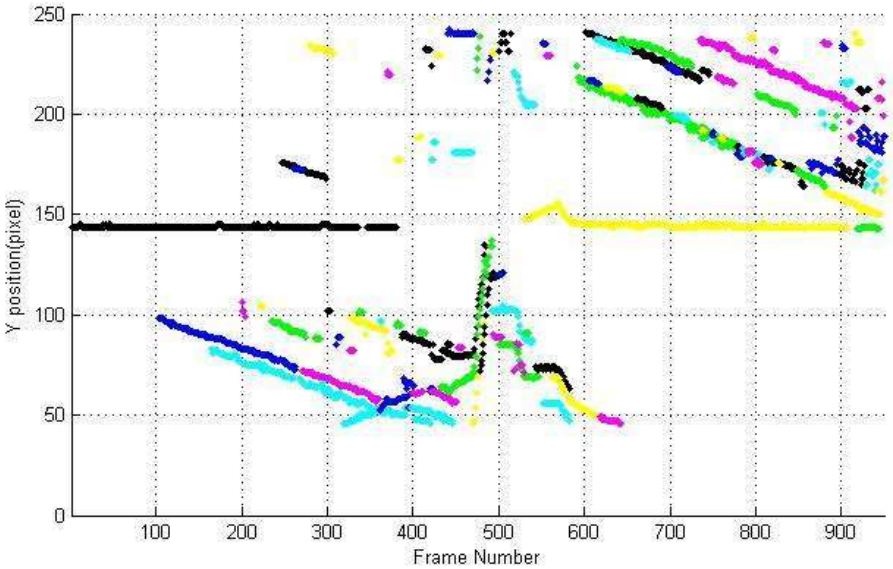
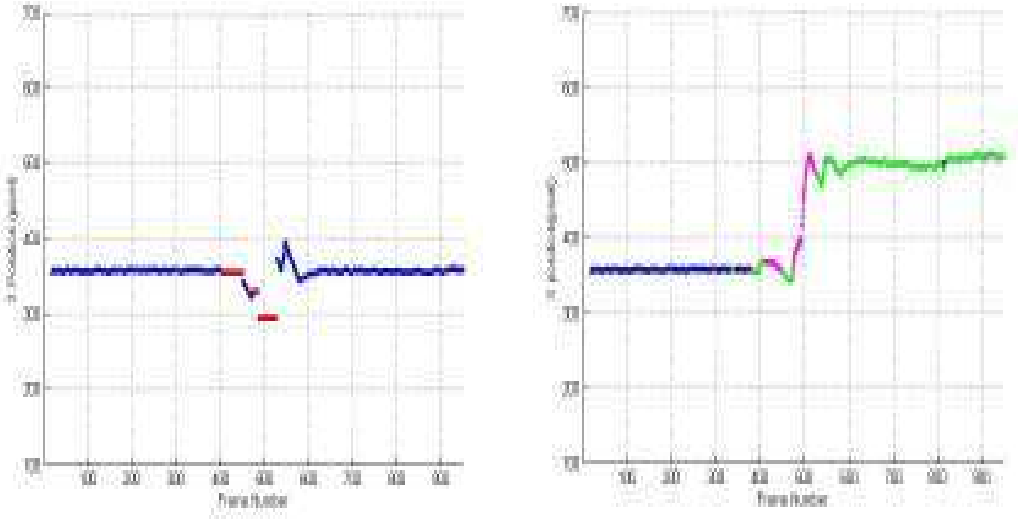
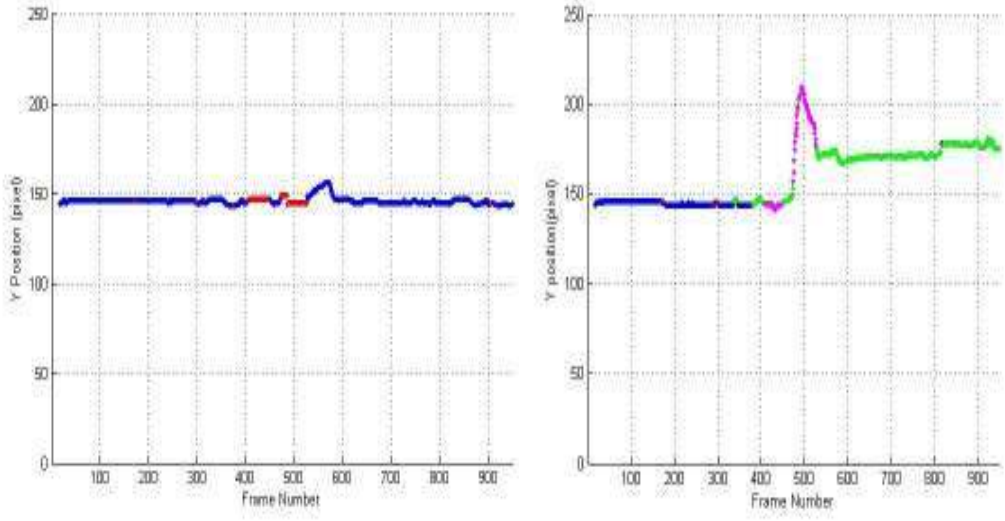


Figure 5.4: Y Positions of the Tracks Obtained by Tracker-1



(a) X Positions of the RTT (b) X Positions of the output of GTES

Figure 5.5: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT (b) Y Positions of the output of GTES

Figure 5.6: Output of RTT vs Output of GTES for Y Positions

Figures 5.5 and 5.6 show the output of both RTT and GTES. The ‘a’ parts of the graphs indicate the tracked target positions and the coast mode of the RTT tracker. The cost intervals are shown by the red color. RTT gives predicted, i.e., ‘no measurement update’ for these intervals. Note that since a tracking camera is used, the claimed target is approximately at the center of the frame throughout the video. The

'b' parts of the figures show the evaluation of the RTT. Blue color in the figure indicates that at these frames the target is correctly tracked by RTT. Magenta color indicates that the target is occluded and RTT is in the coast mode which is consistent with being occluded. In this region GTES indicates the predicted target positions by magenta color by considering the camera motion. The green region is for the part that RTT tracks a wrong target. For this video the true target is occluded in this interval as well but RTT cannot realize that.

V-2 is a typical example that shows the loss of target identity while tracking. After the occlusion region RTT couldn't be able to track the true target but tracks another moving object which is another airplane. The green target position given by GTES is the predicted position of the target.

The rms position error between output of GTES and output of RTT is 1,16 pixels and it is calculated for 370 frames.

5.2 Experimental Results of Video-3

Video-3 contains two targets that are air planes. Both targets enter into the scene about at 50^{th} frame. True target is occluded by clouds between 119^{th} and 159^{th} frame. Before occlusion ends, RTT starts to track a false target and tracks it until the end of Video-3. There is also a huge occlusion interval between 218^{th} and 348^{th} frames. In addition, there are not much background objects throughout the video and this creates camera motion uncertainty. The measurements of the detector are given in Figures 5.7 and 5.8.

The measurements of detector are tracked by Tracker-1 throughout the whole video. The tracks generated by Tracker-1 are given in Figures 5.9 and 5.10. In these figures, each color represents a different track.

Tracker-2 is applied to the tracks generated by Tracker-1. The output of GTES is given in Figures 5.11 and 5.12 together with the output of RTT. The comparison of the two outputs shows the performance of RTT.

This identity loss made by RTT is indicated by GTES and is shown in the figure by green color. The rms position error between output of GTES and output of RTT is 0,30 pixels and it is calculated for 60 frames.

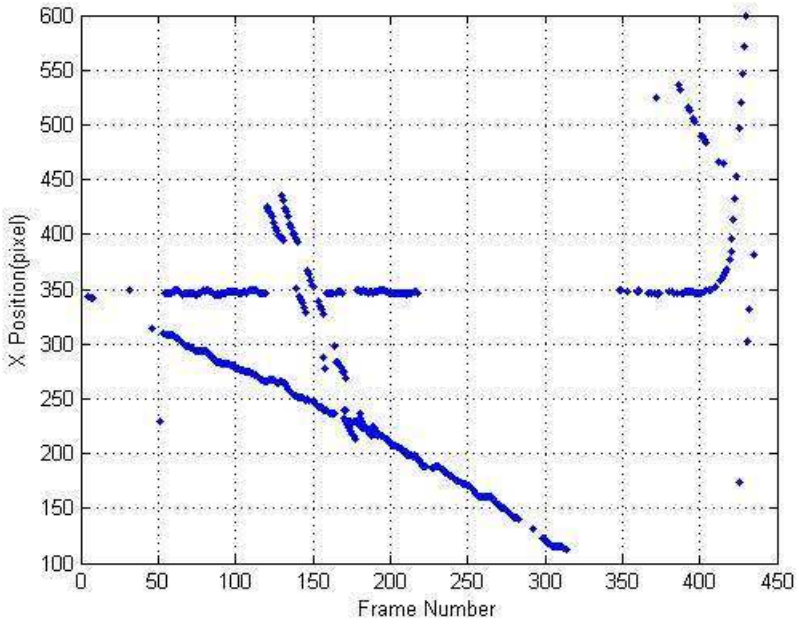


Figure 5.7: X Positions of the Measurements of the Detector for Video-3

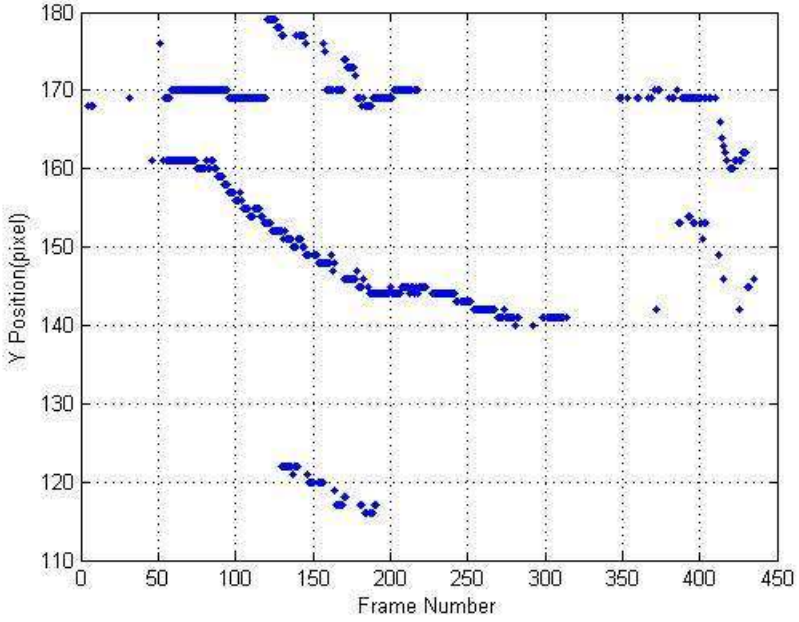


Figure 5.8: Y Positions of the Measurements of the Detector Video-3

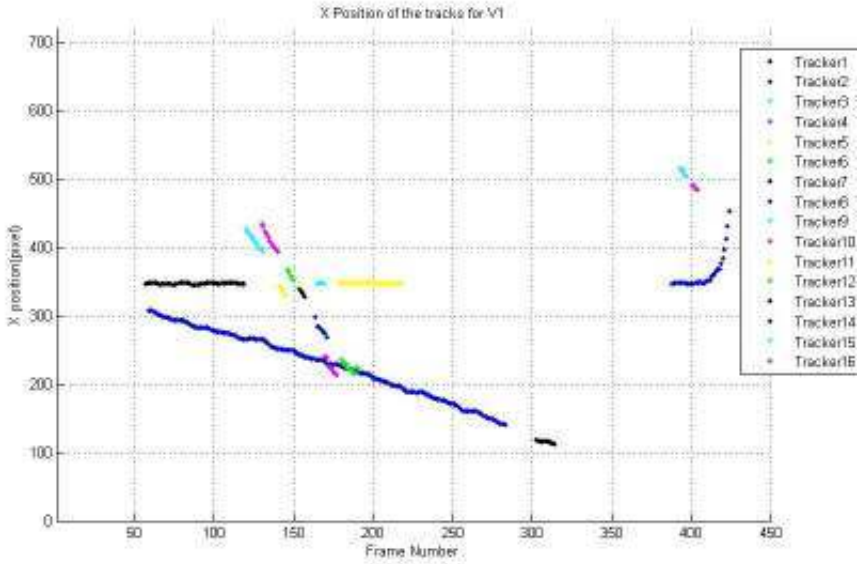


Figure 5.9: X Positions of the tracks generated by Tracker-1

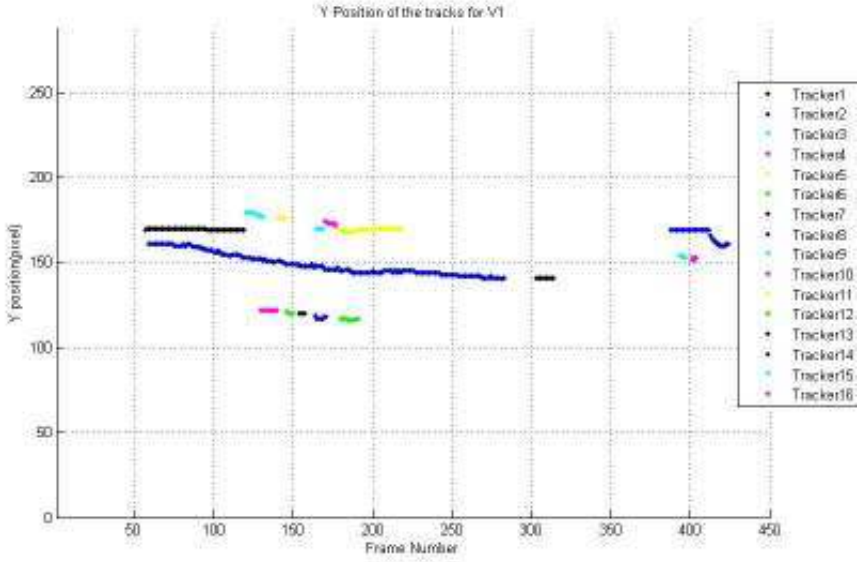
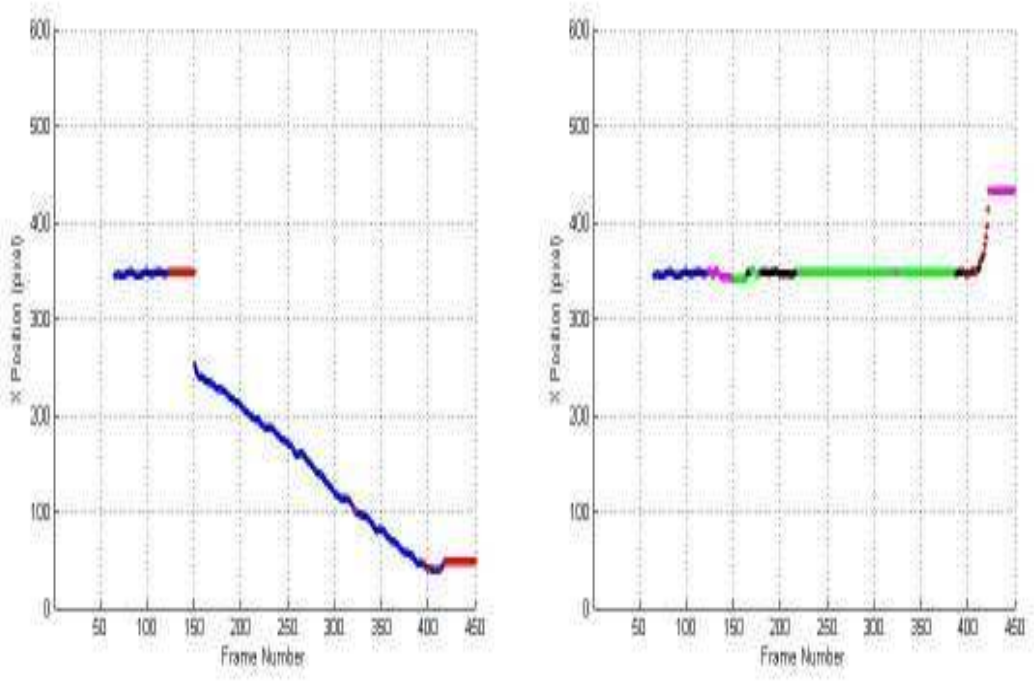
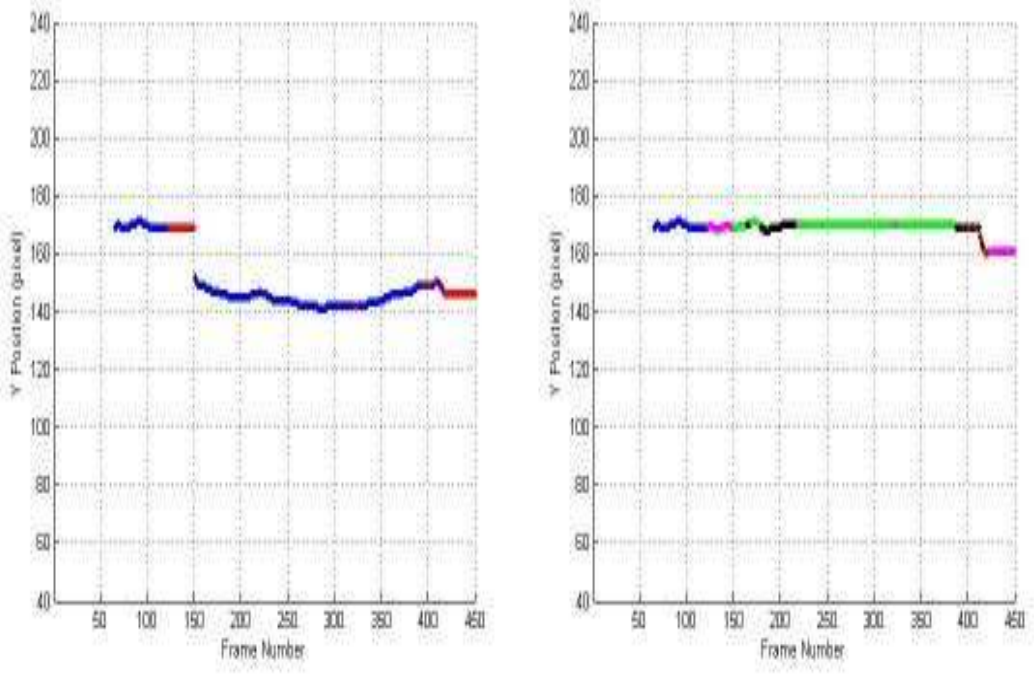


Figure 5.10: Y Positions of the tracks generated by Tracker-1



(a) X Positions of the RTT (b) X Positions of the output of GTES

Figure 5.11: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT (b) Y Positions of the output of GTES

Figure 5.12: Output of RTT vs Output of GTES for Y Positions

5.3 Experimental Results of Video-4

Video-4 contains two targets that are air planes. Both targets stay in the scene from first frame until 160^{th} frame where both are occluded at this frame. Speeds of the two targets are almost same. Analysis of the V4 shows that, RTT loses the track many times altough the target is in the scene. The reason for the loss is probably the high maneuver of the target or the camera. Furthermore, RTT tracks false target between 105^{th} and 146^{th} frames. It starts to track the true target again at 146^{th} frame. In addition, there are not much background objects throughout the video. The measurements of the detector are given in Figures 5.13 and 5.14.

The measurements are tracked by Tracker-1 throughout the whole video. The tracks generated by Tracker-1 are given in Figures 5.15 and 5.16.

Tracker-2 is applied to the tracks obtained by Tracker-1. The output of GTES is given in Figures 5.17 and 5.18 together with the output of RTT. The reason that RTT cannot track but GTES does is the use of IMM which brings flexibility to the tracking system. Note that GTES indicates all of the decision mistakes done by RRT correctly.

The rms position error between output of GTES and output of RTT is 1,25 pixels and it is calculated for 72 frames.

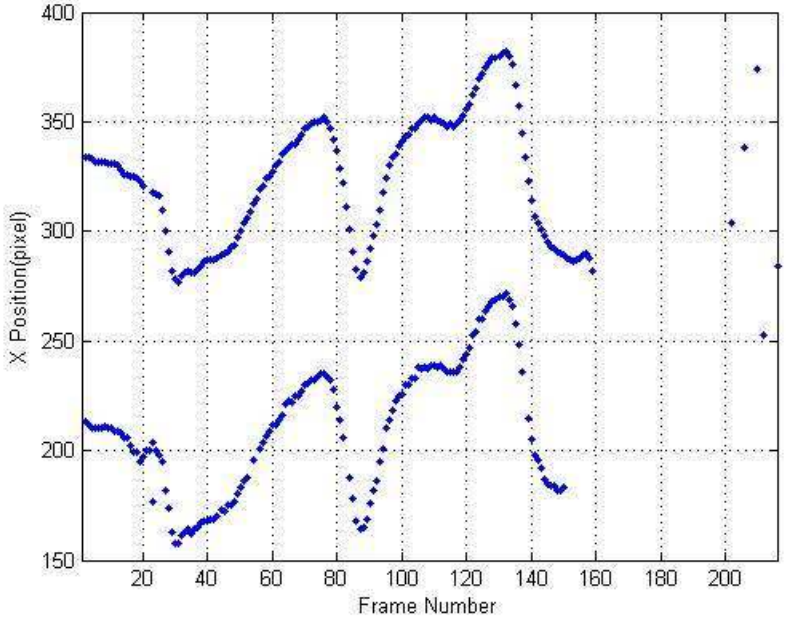


Figure 5.13: X Positions of the Measurements of the Detector for Video-4

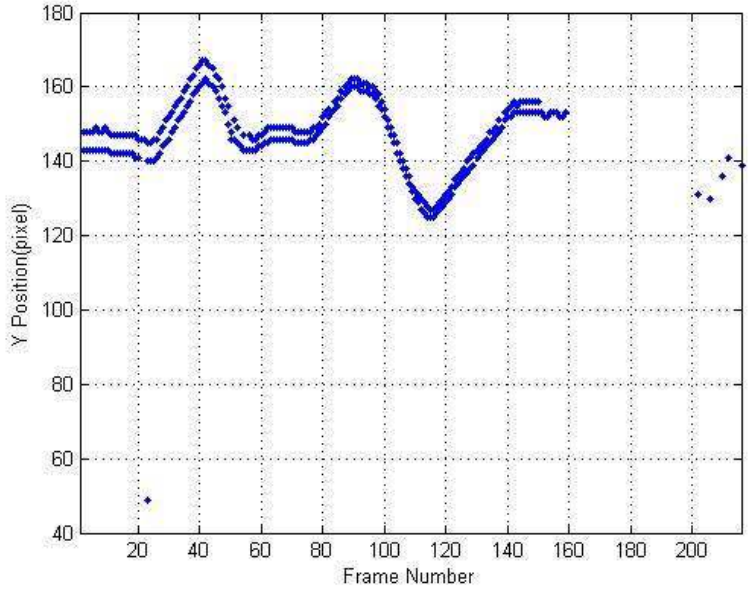


Figure 5.14: Y Positions of the Measurements of the Detector for Video-4

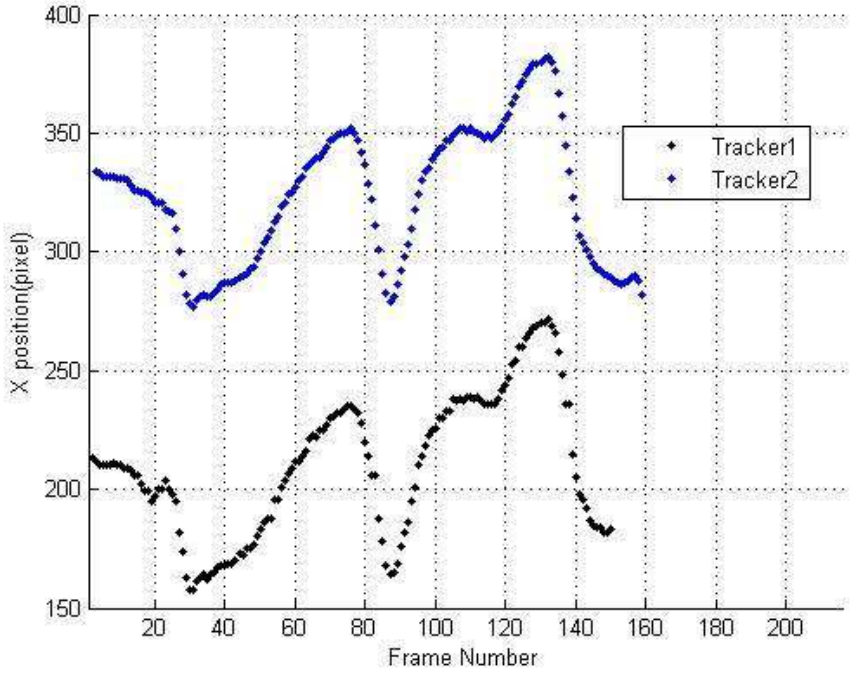


Figure 5.15: X Positions of the tracks generated by Tracker-1

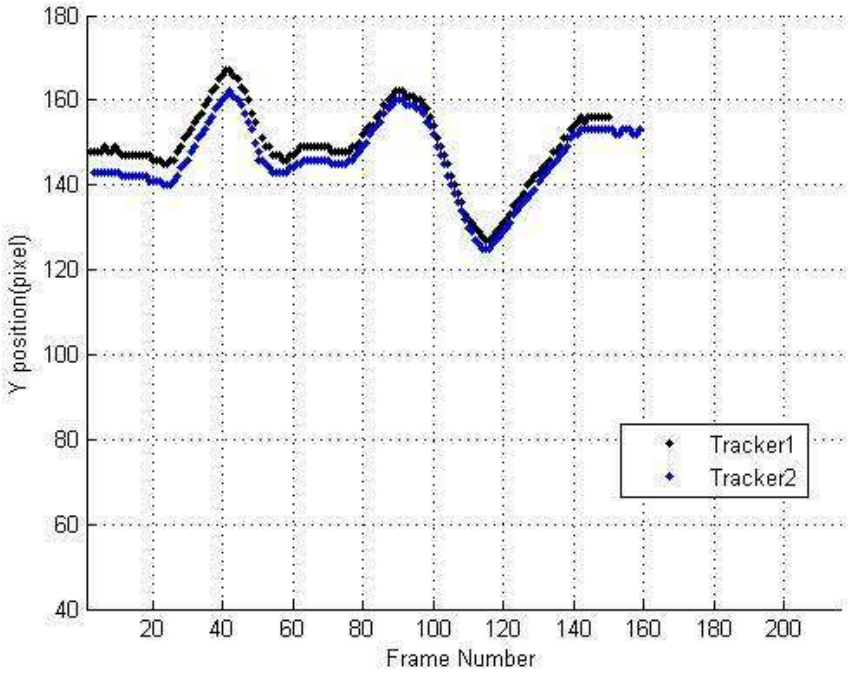
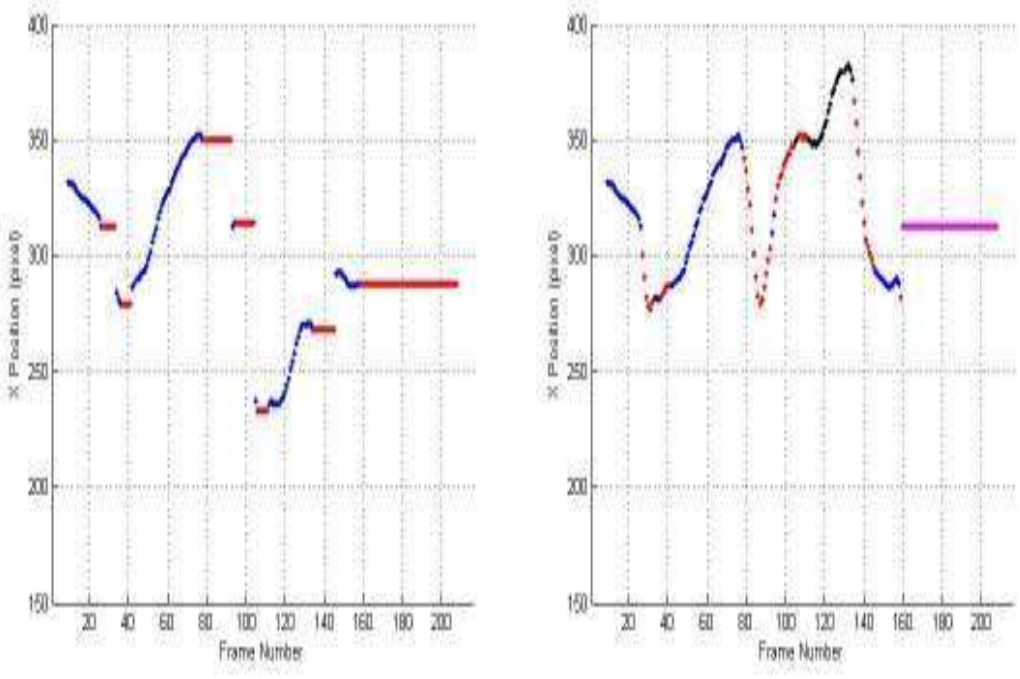


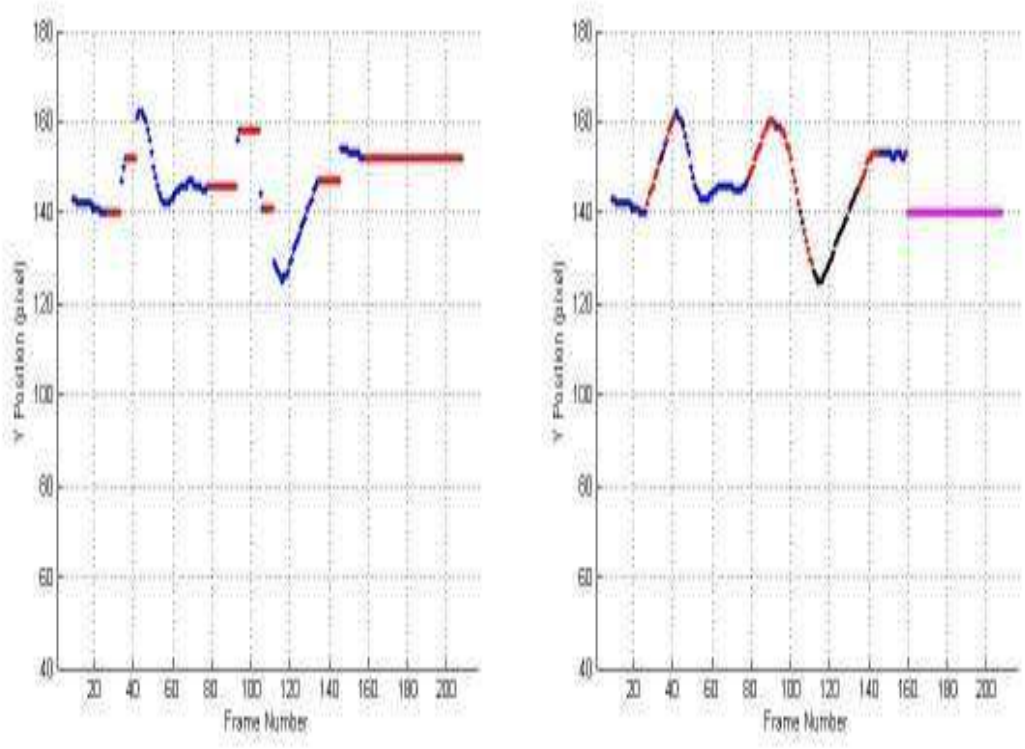
Figure 5.16: Y Positions of the tracks generated by Tracker-1



(a) X Positions of the RTT

(b) X Positions of the output of GTES

Figure 5.17: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT

(b) Y Positions of the output of GTES

Figure 5.18: Output of RTT vs Output of GTES for Y Positions

5.4 Experimental Results of Video-5

Video-5 is simpler compared to the others. It contains one target that is an air plane. The target stays in the scene from the beginning until 303^{rd} frame. There are no background objects so camera motion couldn't be estimated. RTT loses the track although the target is in the scene. The measurements of the detector are given in Figures 5.19 and 5.20.

The measurements are tracked by Tracker-1 throughout the whole video. The tracks generated by Tracker-1 are given in Figures 5.21 and 5.22.

Tracker-2 is applied to the tracks obtained by Tracker-1. The output of GTES is given in Figures 5.23 and 5.24 together with the output of RTT. The comparison of the two outputs shows the performance of RTT.

The rms position error between output of GTES and output of RTT is 0,65 and it is calculated for 271 frames.

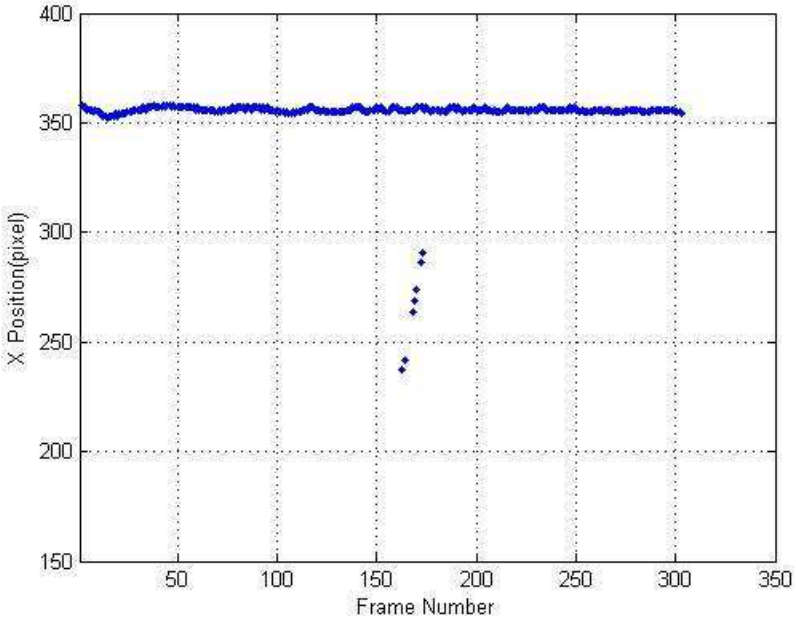


Figure 5.19: X Positions of the Measurements of the Detector for Video-5

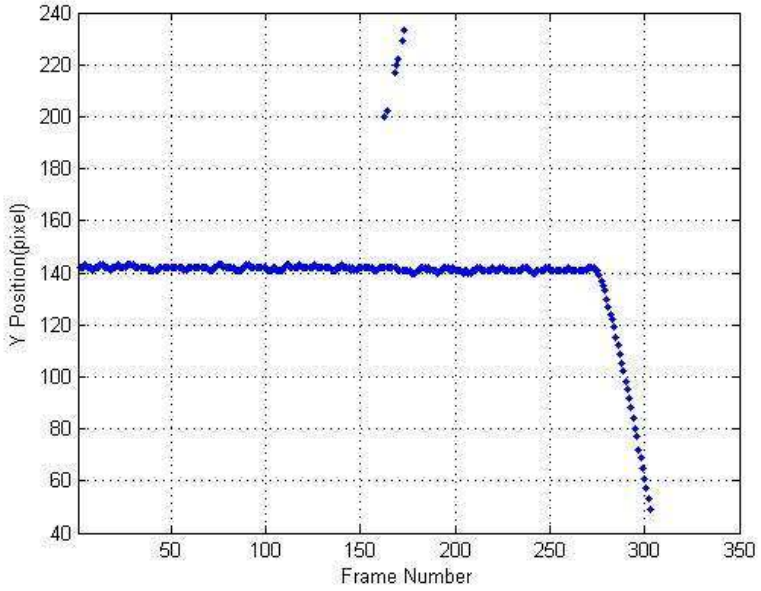


Figure 5.20: Y Positions of the Measurements of the Detector for Video-5

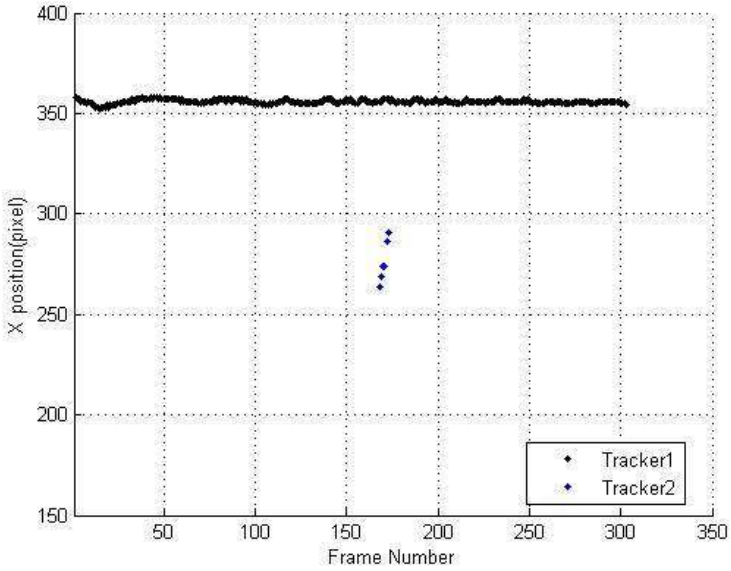


Figure 5.21: X Positions of the tracks generated by Tracker-1

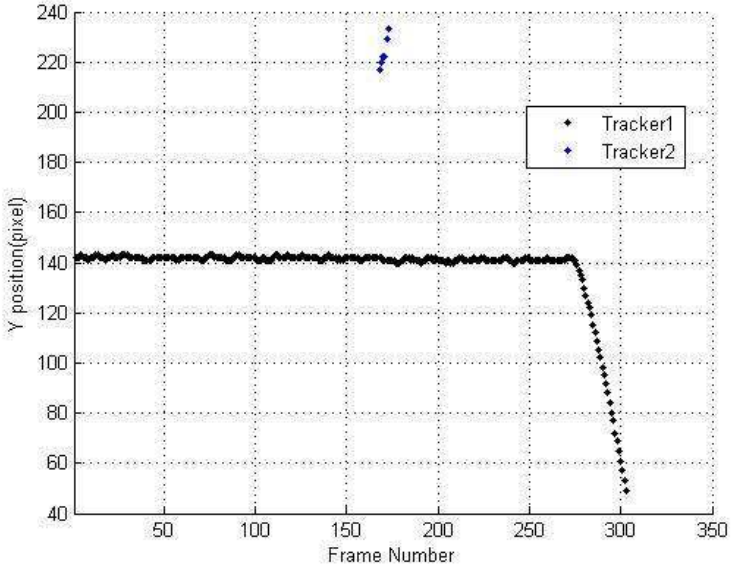
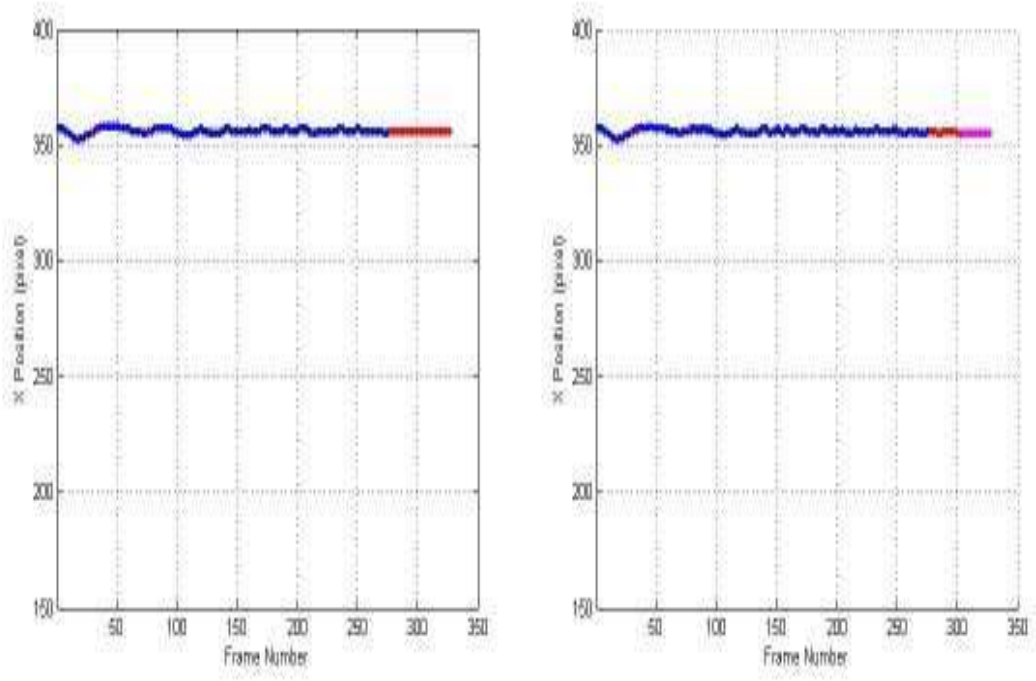
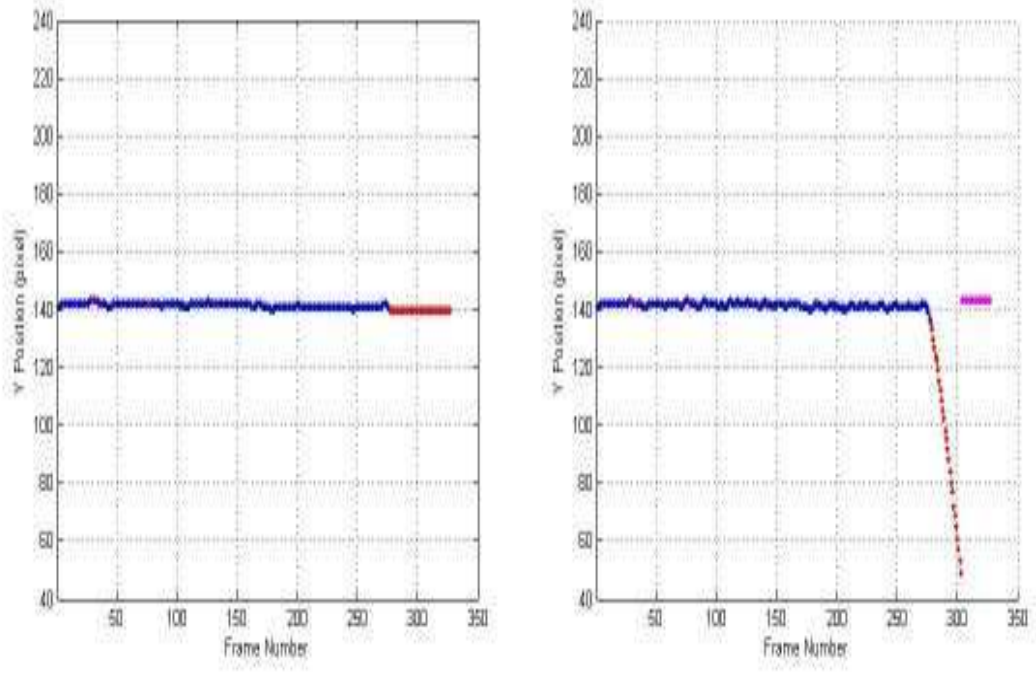


Figure 5.22: Y Positions of the tracks generated by Tracker-1



(a) X Positions of the RTT (b) X Positions of the output of GTES

Figure 5.23: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT (b) Y Positions of the output of GTES

Figure 5.24: Output of RTT vs Output of GTES for Y Positions

5.5 Experimental Results of Video-6

Video-6 contains single target which is a helicopter. The target is almost stationary throughout the video. Helicopter releases flares at 74^{th} and 104^{th} frames. SoV continues to track the target after the first release of flare, however, SoV loses the track of the target and starts to track the flare after second release of flare. The measurements of the detector are given in Figures 5.25 and 5.26. The measurements of the detector are tracked by Tracker-1 throughout the whole video. The tracks generated by Tracker-1 are given in Figures 5.27 and 5.28. In these figures, each color represents a different track.

Tracker-2 is applied to the tracks generated by Tracker-1. The output of GTES is given in Figures 5.29 and 5.30 together with the output of RTT. The comparison of the two outputs shows the performance of RTT.

The figures indicate that RTT cannot track the correct object after the second flare and begins to track the flare. GTES is aware of this mistake done and indicates it by black color.

The rms position error between output of GTES and output of RTT is 0,59 pixels and it is calculated for 87 frames.

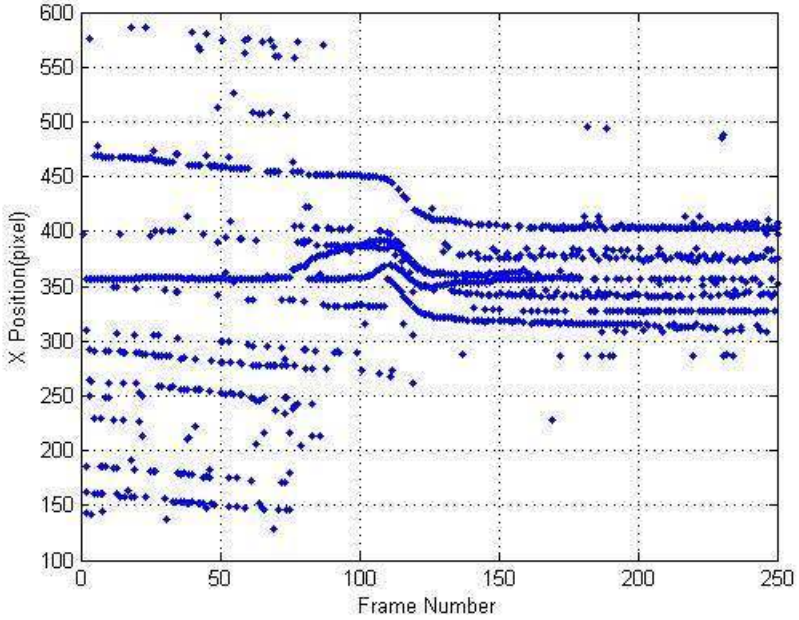


Figure 5.25: X Positions of the Measurements of the Detector for Video-6

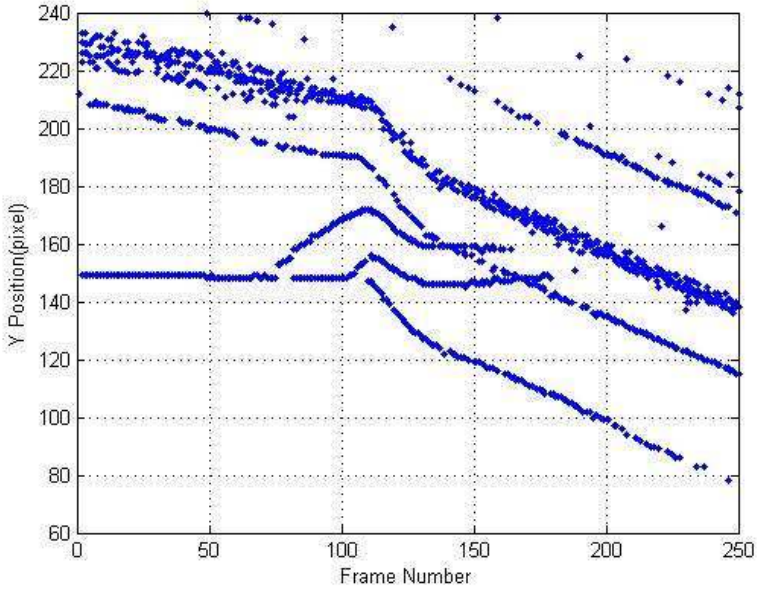


Figure 5.26: Y Positions of the Measurements of the Detector for Video-6

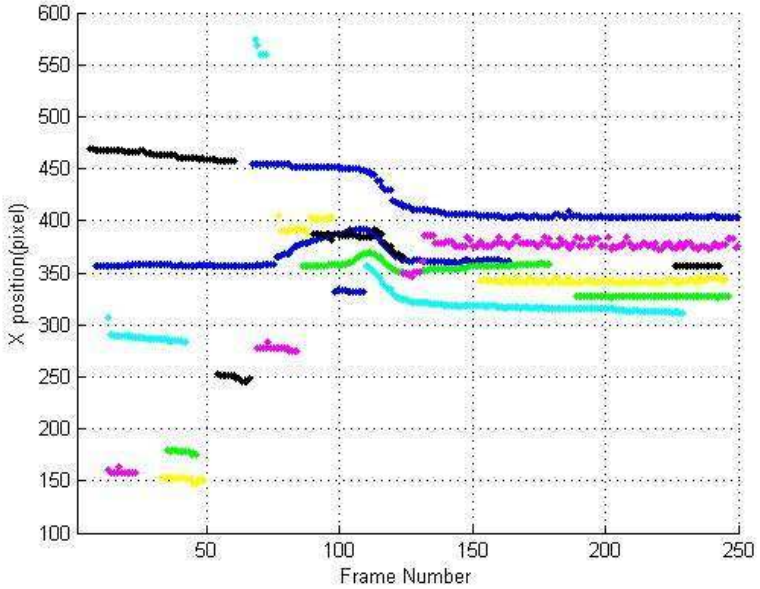


Figure 5.27: X Positions of the tracks generated by Tracker-1

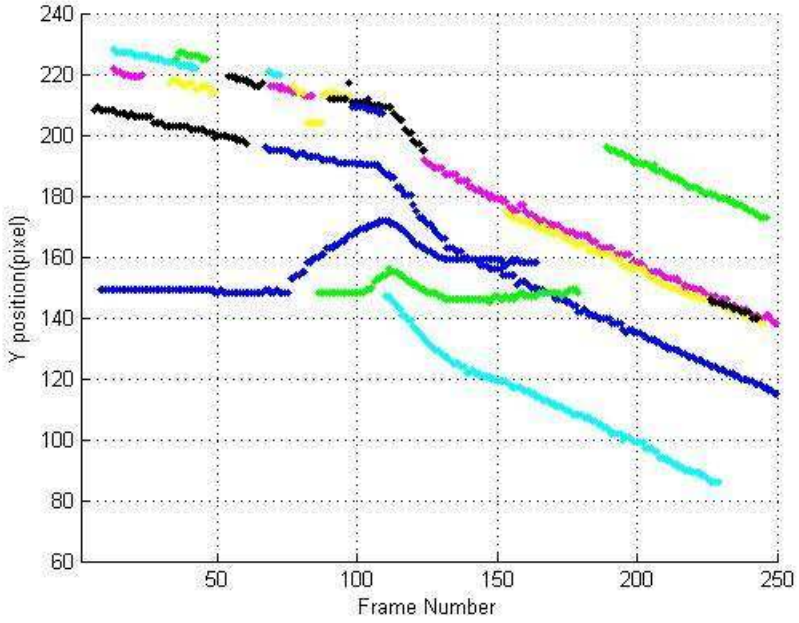
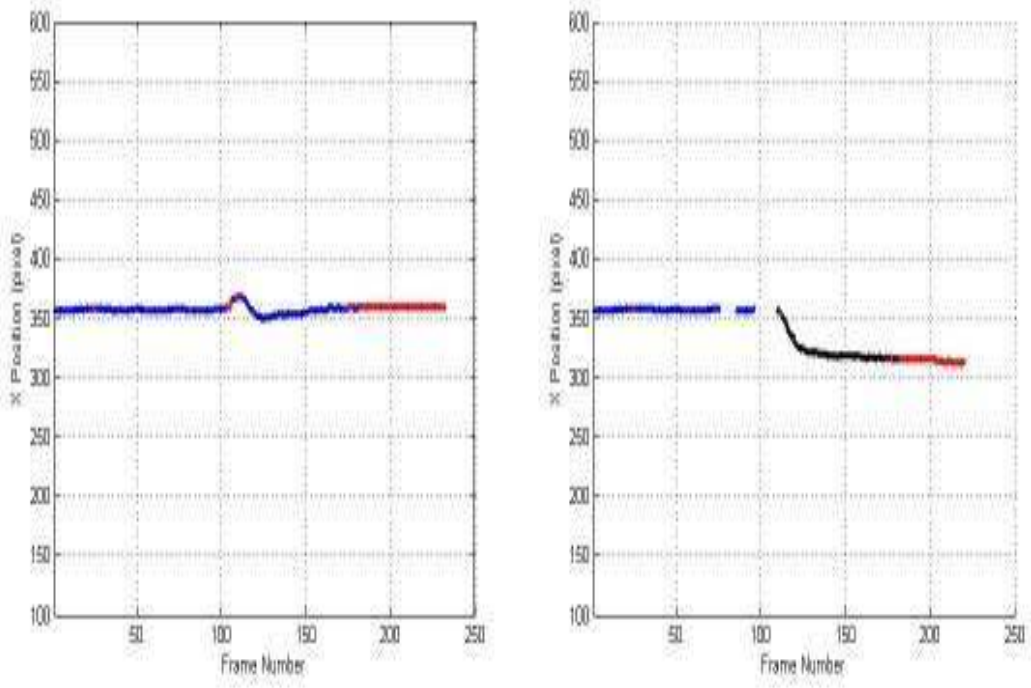


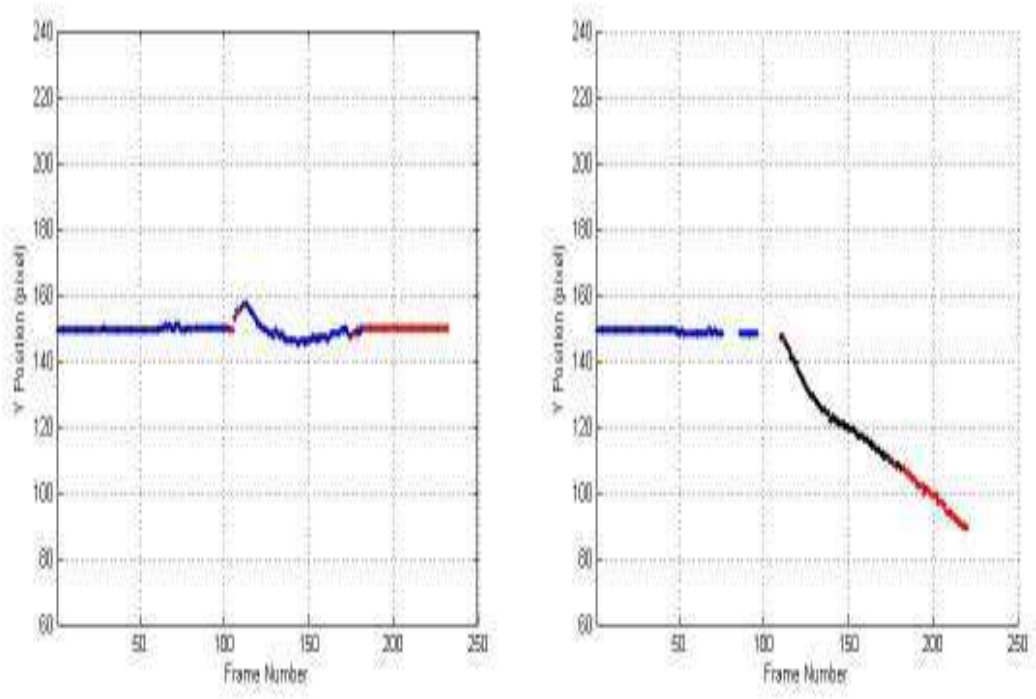
Figure 5.28: Y Positions of the tracks generated by Tracker-1



(a) X Positions of the RTT

(b) X Positions of the output of GTES

Figure 5.29: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT

(b) Y Positions of the output of GTES

Figure 5.30: Output of RTT vs Output of GTES for Y Positions

5.6 Experimental Results of Video-7

Video-7 contains one helicopter target. The target is almost stationary throughout the video. Helicopter releases flares at 154^{th} frame. SoV loses the track of the target and starts to track the flare after the release of flare. Flare disappears at 259^{th} frame from the scene. The camera makes an arbitrary motion between 267^{th} and 279^{th}. Second flare is released at 284^{th} frame. The measurements of the detector are given in Figures 5.31 and 5.32.

The measurements of the detector are tracked by Tracker-1 throughout the whole video. The tracks generated by Tracker-1 are given in Figures 5.33 and 5.34.

Tracker-2 is applied to the tracks obtained by Tracker-1. The output of GTES is given in Figures 5.35 and 5.36 together with the output of RTT. For this video RTT tracks the true target most of the time and it is indicated by the GTES.

The rms position error between output of GTES and output of RTT is 1,23 pixels and it is calculated for 272 frames.

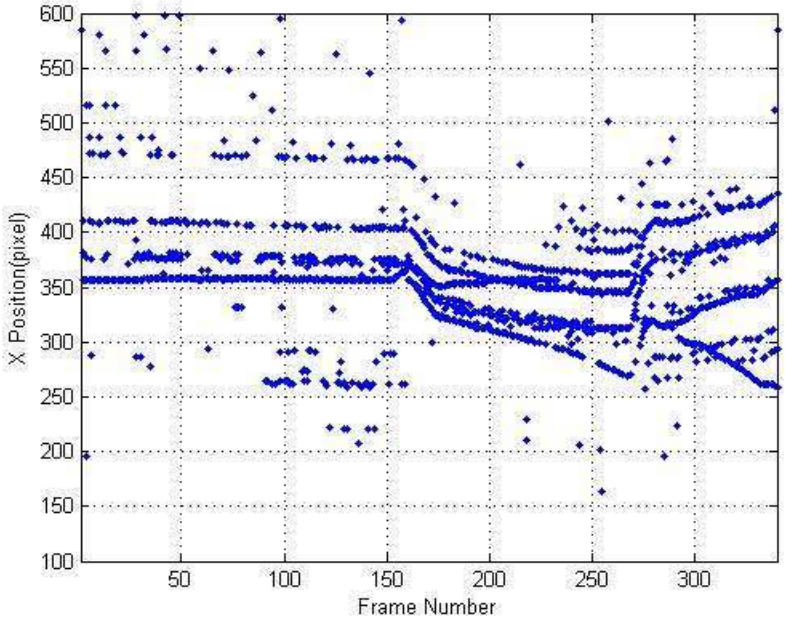


Figure 5.31: X Positions of the Measurements of the Detector for Video-7

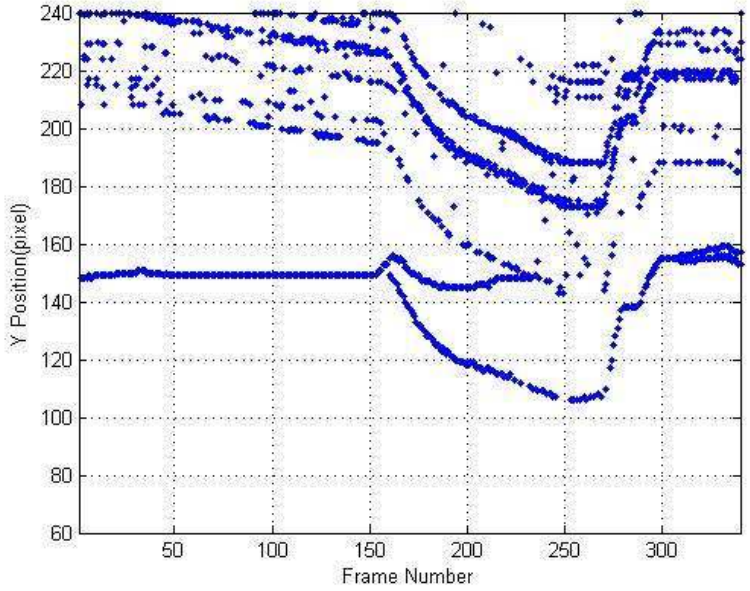


Figure 5.32: Y Positions of the Measurements of the Detector for Video-7

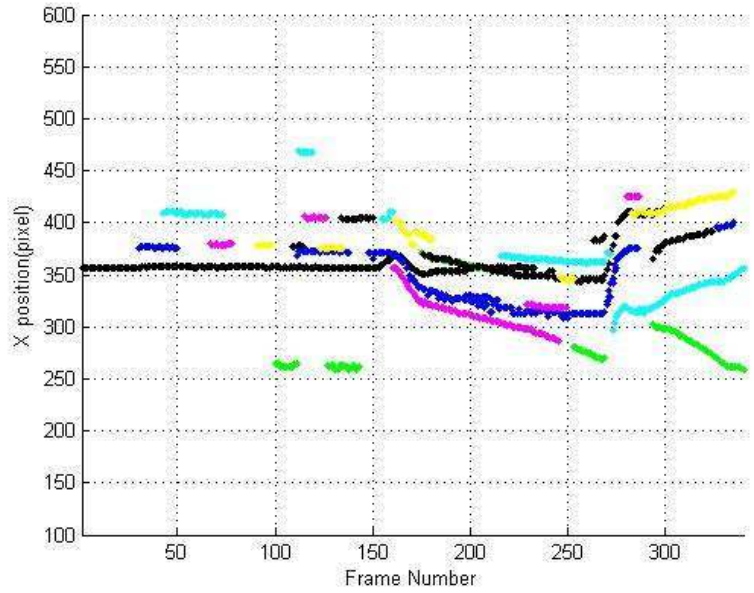


Figure 5.33: X Positions of the tracks generated by Tracker-1

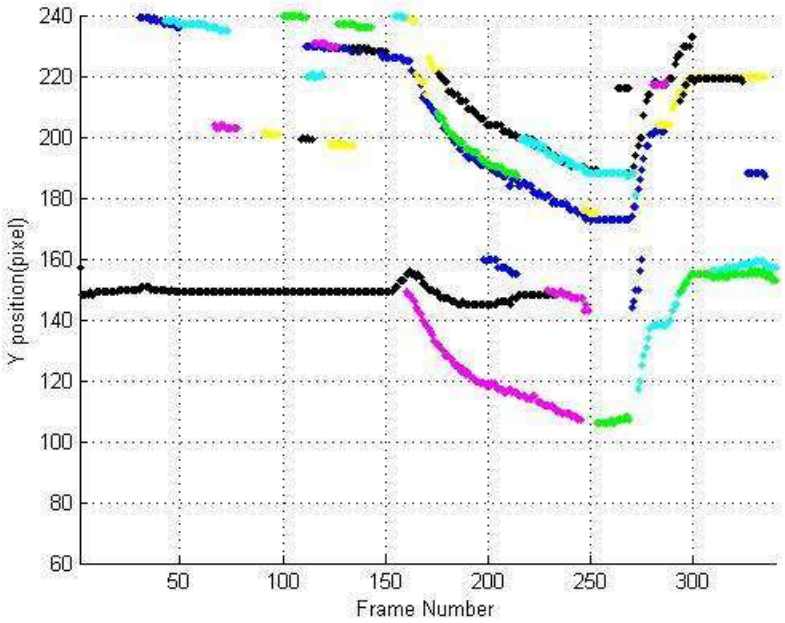
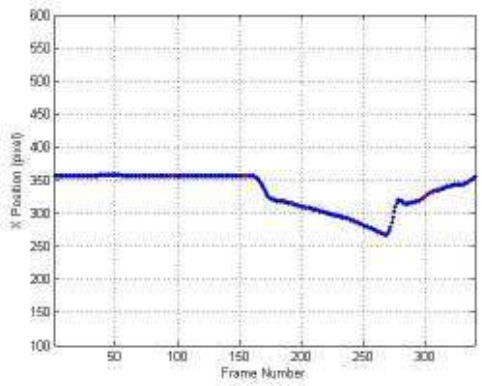
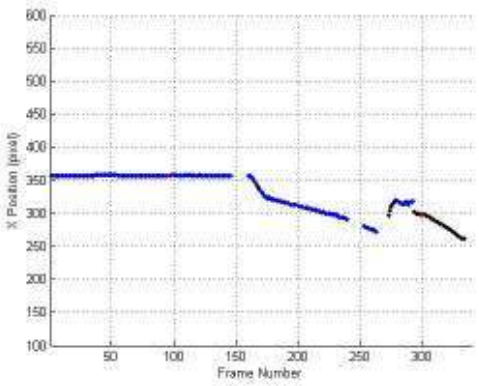


Figure 5.34: Y Positions of the tracks generated by Tracker-1

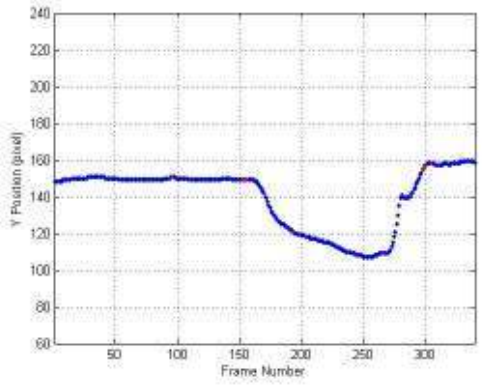


(a) X Positions of the RTT

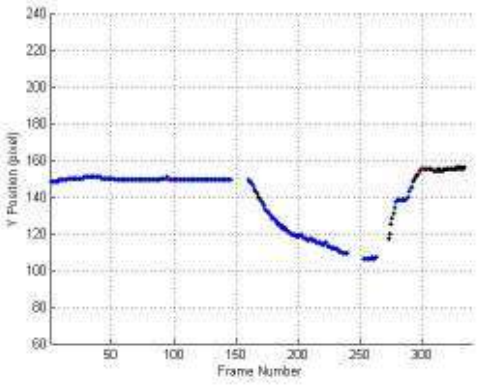


(b) X Positions of the output of GTES

Figure 5.35: Output of RTT vs Output of GTES for X Positions



(a) Y Positions of the RTT



(b) Y Positions of the output of GTES

Figure 5.36: Output of RTT vs Output of GTES for Y Positions

CHAPTER 6

CONCLUSION

The aim of this thesis is to generate an evaluation system for video tracking systems. Video tracking is a broad area so in our work we restricted it in several directions. The video types are restricted as IR videos in this study. This restriction makes the registration problem, so the tracking, more difficult since color information cannot be used. The size of the tracked object is restricted to be small ($\sim$20 pixels). This makes the detection of the object more difficult since additional features, like the contour of the object, its template etc. cannot be used for the detection.

The videos that are aimed to be examined in this study are assumed to be recorded by a tracking camera. This makes the problem a special one due to the relative motion of the target in the scene: the target is at the center of the scene if the tracking camera is tracking the correct object, i.e., the target. At other times the true target position depends on the camera motion. Camera makes some abrupt, unpredictable movements when it loses the target (that is called the 'coast mode' of the camera). The loss of the target is caused by occlusions most of the time. At the end of this period the camera begins to track an object again which may be the correct target or not. Finding the correct object after the abrupt camera motion requires the estimation of the motion.

The system developed in this study called the 'Ground Truth Extraction System (GTES)' mainly generates a trajectory of the true target that is called the 'ground truth', compares it with the trajectory generated by another tracking system and labels the time intervals as 'true target', 'false target' etc.

Since the targets aimed are small the main tracking problem is the association of the objects found in one frame to the true target. Target position changes very little according to different trackers. However we have also implemented a Kalman smoother to smooth the position of the target when it is tracked. The performance of this block

of the GTES is assessed by extracting the hand labeled positions of the target and comparing it with the output of the GTES. Association performance of the GTES is assessed by examining the 7 videos that are tested.

Our system as mentioned above has a camera motion estimation part. There is a huge literature on the camera motion estimation. However here we propose a novel algorithm that is suitable for our purposes. Our aim is to make the target association correctly after the abrupt motion of the camera. Very precise motion estimation in the sub-pixel accuracy are not necessary. Tracker-1 tracks multiple targets for which some correspond to background objects.. The camera motion estimator developed here uses the tracks of the background objects to extract the camera motion. A special Kalman filter is developed for this purpose. The special Kalman filter works on the principle that all background objects have similar velocities and this velocity is different than the velocities of the moving objects. It is a single Kalman filter but the state and the measurement dimensions vary depending on the number of background objects.

To find the trajectory of the true target, two trackers, called Tracker-1 and Tracker-2 are implemented each using the Interacting Multiple Model (IMM). Measurements of the Detector are tracked via Tracker-1. Tracker-1 is a multi-target tracker that uses Global Nearest Neighbor (GNN) method to associate the measurements to the true track. Its output gives various tracks which need to be associated. Tracker-2 is used for the association of the tracks resulted from Tracker-1 to the true target. Camera motion estimation is used while associating the tracks via Tracker-2. The output of the second offline tracker gives the raw ground-truth. The raw ground-truth is smoothed by using Kalman smoother which is the final block of the GTES.

Flare is another problem in for military applications. We have modified our system so it can cope with the flare problem.

We have used 7 videos to test all parts of the system. These videos are selected among 200 videos because they are more problematic compared to the others . GTES makes no association errors in these videos and the rms error of position of GTES is also quite satisfactory .

For the evaluation part we have used a video tracker that is called Real Time Tracker (RTT). The performance of RTT is evaluated by comparing the output of RTT with the output of GTES. In two of the videos more than one flare is released to deceive

the video tracker.

The summary of all the work done in this thesis is given below:

• Implementation of two IMM trackers, one is for multi target, the other is for single target tracking

• Implementation of a Kalman smoother

• Generation of a novel camera motion estimation algorithm and its implementation

• Generation and implementation of an evaluation system that evaluates a given video tracker

The system developed in this thesis can be improved in several ways. One possible direction is to enrich the system by some other target motion models, for example hovering helicopter, ballistic target, etc. The application of camera motion estimation to the videos that contain parallax effect seems to be possible by developing new motion models for the background. Tracking of medium or large sized objects require different techniques but the ideas introduced here may be adopted for them.

REFERENCES

[1] Caviar: Context aware vision using image-based active recognition. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, last visited on January 2017.

[2] A video database for testing change detection algorithms. http://changedetection.net/, last visited on January 2017.

[3] Viper: The video performance evaluation resource. http://viper-toolkit.sourceforge.net/, last visited on January 2017.

[4] P. Arnoul, M. Viala, J. P. Guerin, and M. Mergy. Traffic signs localisation for highways inventory from a video camera on board a moving collection van. *Proceedings of the 1996 IEEE intelligent vehicles symposium*, pages 141–146, 1996.

[5] A. E. Arslan. Visual tracking with group motion approach. Master's thesis, METU, 2003.

[6] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking*. YBS, 1995.

[7] S. Becker, W. Huebner, and M. Arens. Independent motion detection with a rival penalized adaptive particle filter. *Conference on Unmanned/Unattended Sensors and Sensor Networks*, 2014.

[8] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[9] Y. Boers and J. N. Driessen. Particle filter based detection for tracking. *Proceedings of the 2001 american control conference*, 1-6:4393–4397, 2001.

[10] J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.

[11] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. *Proceeding AVSS '09 Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 559–564, 2009.

[12] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland. *Applied Optimal Estimation*. The M.I.T. Press, 1974.

[13] E. Hayman and J. O. Eklundh. Statistical background subtraction for a mobile observer. *ICCV*, 2003.

[14] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans Pattern Anal Mach Intell*, 1:206–214, 1979.

[15] W. Y. Kan, J. V. Krogmeier, and P. J. Doerschuk. Model-based vehicle tracking from image sequences with an application to road surveillance. *Optical Engineering*, 35:1723–1729, 1996.

[16] L. Kurnianggoro, Wahyono, and Y. Yu. Online background-subtraction with motion compensation for freely moving camera. *Lecture Notes in Computer Science*, 9772:569–578, 2016.

[17] A. Mittal and D. Huttenlocher. Scene modeling for wide area surveillance and image synthesis. *IEEE Computer Society Conference on Computer Vision and Pattern Recognation*, 2:160–167, 2000.

[18] Y. Ren, C. S. Chua, and Y. K. Ho. Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24:183–196, 2003.

[19] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. *Computer Vision, 2009 IEEE 12th International Conference on*, 2009.

[20] S. Shtern and B. T. Aharon. A semi-definite programming approach for robust tracking. *Mathematical Programming*, 156:615–656, 2016.

[21] K. Soyeon, D. W. Yang, and H. W. Park. A disparity-based adaptive multi-homography method for moving target detection based on global motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:1407–1420, 2015.

[22] F. Sun, K. Qin, W. Sun, and H. Guo. Fast background subtraction for moving cameras based on nonparametric models. *Journal of Electronic Imaging*, 25, 2016.

[23] H. Wang and S. K. Nguang. Multi-target video tracking based on improved data association and mixed kalman/h_∞ filtering. *IEEE Sensors Journal*, 16, 2016.

[24] H. Wu and Q. Zheng. Self-evaluation for video tracking systems. 24^{th} *Army Science Conference*, 2004.

[25] C. Yuan, G. Medioni, J. Kang, and I. Cohen. Detecting motion regions in presence of strong parallax from a moving camera by multi-view geometric constraints. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 29, 2007.

[26] D. Zamalieva, A. Yilmaz, and J. W. Dawis. Exploiting temporal geometry for moving camera background subtraction. *22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[27] Y. Zhong, A. K. Jain, and M. P. Dubuisson-Jolly. Object tracking using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:544–549, 2002.

APPENDIX A

KALMAN FILTER

Kalman filter is used for the purpose of tracking throughout the thesis. Kalman filter uses measurements taken in the time interval [0 n] to generate an estimate of the state at time n for each n [12]. One of the basic assumptions of the Kalman filter is the Markov property of the state. This requires the noises to be independent. Kalman filter is the optimal filter if the system is linear and measurement and process noises are Gaussian. Under these conditions the state and the observation equations are written below.

$$x_{k+1} = A_k x_k + G_k w_k \tag{A.1}$$

$$y_k = C x_k + H v_k \tag{A.2}$$

where $w_k \sim N(0, Q)$, $v_k \sim N(0, R)$ and $x_0 \sim N(\bar{\mathrm{x}}_0, P_0)$. All of these random variables are independent. $\{w_k\}_{k=1}^{\infty}$ is called the process noise and $\{v_k\}_{k=1}^{\infty}$ is called the measurement noise.

Kalman filter starts with initial state distribution that is parametrized by its mean and covariance matrix and predicts the state and its covariance by using the equations given in (A.3) and (A.4). Then, It uses the measurement which is y_k to update the filter based on the equations given in (A.6) and (A.7).[12]

$$x_{k+1|k} = A_k x_{k|k} \tag{A.3}$$

$$P_{k+1|k} = A_k P_{k|k} {A_k}^T + G Q_k G^T \tag{A.4}$$

$$K_k = P_{k|k-1} C^T (C P_{k|k-1} C^T + H R_k H^T)^{-1} \tag{A.5}$$

$$x_{k|k} = x_{k|k-1} + K_k (y_k - C x_{k|k-1}) \tag{A.6}$$

$$P_{k|k} = P_{k|k-1} - K_k C P_{k|k-1} \tag{A.7}$$

APPENDIX B

INTERACTING MULTIPLE MODEL

Interacting Multiple Model (IMM) is a combination of more than one model to improve the performance of a tracker [6]. In tracking applications usually a 'constant velocity' model is used. This model assumes that the acceleration of the target is a zero mean random process and is modelled as the process noise. If the process noise power is small then the system cannot track maneuvering targets since maneuvers have large accelerations compared to the flexibility introduced to the system by the process noise. On the other hand if process noise is large then the tracking performance of the system degrades when the target has constant velocity. A solution to this problem is to use more than one model, usually two models. One of them models the almost constant velocity movements with a small covariance matrix and the other is selected with a large covariance to overcome the problems introduced by acceleration. The two different models are defined as:

$$x_{k+1}^1 = Ax_k^1 + Gw_k^1 \tag{B.1}$$

$$x_{k+1}^2 = Ax_k^2 + Gw_k^2 \tag{B.2}$$

where $w_k^1 \sim N(0, Q_k^1)$ and $w_k^2 \sim N(0, Q_k^2)$. Q_k^1 is chosen almost 10 times greater than Q_k^2 so that IMM could track the arbitrary position changes.

There are different ways of using more than one model. One very efficient way is to use Interactive Multiple Model (IMM). One cycle of the IMM algorithm is given below.

1) Transition and mode probability matrices are written as in (B.3) and (B.4) relatively.

$$\xi = \begin{bmatrix} \xi_{11} & \xi_{12} \\ \xi_{21} & \xi_{22} \end{bmatrix} \tag{B.3}$$

$$\mu_k = \begin{bmatrix} \mu_k^1 & \mu_k^2 \end{bmatrix} \tag{B.4}$$

2) Initially we have $x_{k|k}^1$, $P_{k|k}^1$, $x_{k|k}^2$, and $P_{k|k}^2$.

3) All parameters below are calculated in order to complete interaction/mixing process.

$$c^1 = \xi_{11}\mu_k^1 + \xi_{21}\mu_k^2 \tag{B.5}$$

$$c^2 = \xi_{12}\mu_k^1 + \xi_{22}\mu_k^2 \tag{B.6}$$

$$\mu^{11} = \frac{\xi_{11}\mu_k^1}{c^1} \tag{B.7}$$

$$\mu^{21} = \frac{\xi_{21}\mu_k^2}{c^1} \tag{B.8}$$

$$\mu^{12} = \frac{\xi_{12}\mu_k^1}{c^2} \tag{B.9}$$

$$\mu^{22} = \frac{\xi_{22}\mu_k^2}{c^2} \tag{B.10}$$

$$x_{k|k}^{01} = \mu^{11}x_{k|k}^1 + \mu^{21}x_{k|k}^2 \tag{B.11}$$

$$x_{k|k}^{02} = \mu^{12}x_{k|k}^1 + \mu^{22}x_{k|k}^2 \tag{B.12}$$

$$P_{k|k}^{01} = \mu^{11}(P_{k|k}^1 + (x_{k|k}^1 - x_{k|k}^{01})(x_{k|k}^1 - x_{k|k}^{01})^T) + \mu^{21}(P_{k|k}^2 + (x_{k|k}^2 - x_{k|k}^{02})(x_{k|k}^2 - x_{k|k}^{02})^T) \tag{B.13}$$

$$P_{k|k}^{02} = \mu^{12}(P_{k|k}^1 + (x_{k|k}^1 - x_{k|k}^{01})(x_{k|k}^1 - x_{k|k}^{01})^T) + \mu^{22}(P_{k|k}^2 + (x_{k|k}^2 - x_{k|k}^{02})(x_{k|k}^2 - x_{k|k}^{02})^T) \tag{B.14}$$

4) Prediction of state and covariance of both model are calculated. The equations of one model are given in (B.15) and (B.16)

$$x_{k+1|k}^1 = Ax_{k|k}^{01} \tag{B.15}$$

$$P_{k+1|k}^1 = AP_{k|k}^{01}A^T + GQ^1G^T \tag{B.16}$$

5) Measurement estimation and its covariance are calculated as given in (B.17) and (B.19). (B.18) is the difference between the measurement and the estimation.

$$y_{k+1|k}^1 = Cx_{k+1|k}^1 \tag{B.17}$$

$$\widetilde{y}_{k+1|k}^1 = z_{k+1} - y_{k+1|k}^1 \tag{B.18}$$

$$S^1_{k+1|k} = CP^1_{k+1|k}C^T + HRH^T \tag{B.19}$$

6) D matrix is defined as in (B.20).Likelihood factors are calculated as in (B.21). Mode probability is updated by using these informations as shown in (B.22)

$$D = \begin{bmatrix} \Lambda^1 & 0 \\ 0 & \Lambda^2 \end{bmatrix} \tag{B.20}$$

$$\Lambda^1 = \frac{e^{(-0.5)(\widetilde{y}^1_{k+1|k})^T (S^1_{k+1|k})^{-1} (\widetilde{y}^1_{k+1|k})}}{\sqrt{2\pi |S^1_{k+1|k}|}} \tag{B.21}$$

$$\mu_{k+1} = \|\mu_k \xi D\| \tag{B.22}$$

7) State and covariance estimates are calculated as shown in (B.24) and (B.25) relatively. Same equations are applied for second filter.

$$K^1_{k+1} = P^1_{k+1|k} C^T (S_{k+1|k})^{-1} \tag{B.23}$$

$$x^1_{k+1|k+1} = x^1_{k+1|k} + K^1_{k+1} \widetilde{y}^1_{k+1|k} \tag{B.24}$$

$$P^1_{k+1|k+1} = P^1_{k+1|k} - K^1_{k+1} C P^1_{k+1|k} \tag{B.25}$$

8) Outputs are calculated as shown in (B.26) and (B.27).

$$x_{k+1|k+1} = \mu^1_{k+1} x^1_{k+1|k+1} + \mu^2_{k+1} x^2_{k+1|k+1} \tag{B.26}$$

$$\begin{aligned} P_{k+1|k+1} = \mu^1_{k+1}(P^1_{k+1|k+1} + (x^1_{k+1|k+1} - x_{k+1|k+1})(x^1_{k+1|k+1} - x_{k+1|k+1})^T) + \\ \mu^2_{k+1}(P^2_{k+1|k+1} + (x^2_{k+1|k+1} - x_{k+1|k+1})(x^2_{k+1|k+1} - x_{k+1|k+1})^T) \end{aligned} \tag{B.27}$$

9) Return to Step 2 and do the same calculations in order to get parameters of next frame.

APPENDIX C

KALMAN SMOOTHER

In the literature there are three types of smoothing methods which are fixed interval smoothing, fixed point smoothing and fixed lag smoothing. Fixed interval smoothing which is used in this thesis uses the measurements of an interval to find better estimations of the state also in this interval. So the aim of fixed interval smoothing is to find $p(x_i|y_0, ..., y_N)$ for all 0≤i≤N for the interval [0 N]. In the literature, two different algorithms which are Rauch,Tung, Striebel and Fraser,Potter algorithms [10] are proposed for fixed interval smoothing . Rauch,Tung and Striebel is used in the thesis. Rauch,Tung and Striebel smoother uses Kalman filter state estimate, state prediction and covariance prediction in order to correct the Kalman filter. Performance of Kalman filter is poor for early frames of measurements et. Kalman smoother uses backward filtering and wipes out this drawback of Kalman filter. Standard Kalman filter equations are given in appendix A. For the time k, forward pass gives $x_{k|k}$,$x_{k+1|k}$,$P_{k|k}$,$P_{k+1|k}$. Backward pass starts at time N, initializes the backward filter with $x^s{}_N = x_{N|N}$ and $P^s{}_N = P_{N|N}$ and apply the backward filtering equations which are given below [10].

$$K^s{}_k = P_{k|k}A^T(P_{k+1|k})^{-1} \tag{C.1}$$

$$P^s{}_k = P_{k|k} - K^s{}_k(P_{k+1|k} - P^s{}_{k+1})(K^s{}_k)^T \tag{C.2}$$

$$x^s{}_k = x_{k|k} + K^s{}_k(x^s{}_{k+1} - x_{k+1|k}) \tag{C.3}$$