BACKGROUND TRACKING OF A VIDEO TAKEN FROM A FRONT CAMERA OF NON MANEUVERING VEHICLE

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

BY

ÖNDER ÜNVER

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

FEBRUARY 2014

Approval of the thesis:

BACKGROUND TRACKING OF A VIDEO TAKEN FROM A FRONT CAMERA OF NON MANEUVERING VEHICLE

submitted by **ÖNDER ÜNVER** 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. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. Mübeccel Demirekler
Supervisor, **Electrical and Electronics Eng. Dept., METU** ________________

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU ________________

M.Sc. Hakan Caner
Aselsan Inc. ________________

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: ÖNDER ÜNVER

Signature :

ABSTRACT

BACKGROUND TRACKING OF A VIDEO TAKEN FROM A FRONT CAMERA OF NON MANEUVERING VEHICLE

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

February 2014, 64 pages

In this study, a novel background tracking technique is proposed that uses extended Kalman Gaussian mixture probability hypothesis density filtering approach. Since the background in a movie, taken from a front camera of a non maneuvering moving vehicle, exhibits a non-stationary nature, tracking the background is usually done by using pixel-wise comparisons in consequent frames. Besides, some methods use features of the background to track it. The proposed method uses the feature tracking approach. The features are chosen as the corner points extracted from each video frame by using Harris corner detector. Linear motion model and non-linear measurement model are developed to predict and update the states of the features. Based on these models, the time varying number of features are tracked by extended Kalman Gaussian mixture probability hypothesis density filter. The method propagates the intensities of the targets based on random set theory and the Kalman filtering approach. MATLAB environment is used to implement the proposed background tracking method. Some simulated results of proposed method are shown for different conditions. The results indicate that the proposed method can be used for background tracking of a video instead of classical background tracking methods under some assumptions.

Keywords: Background Tracking, Random Finite Sets, Probabilty Hypothesis Filter, Harris Corner Detector, Extended Target Tracking,Extended Kalman

ÖZ

İLERİ YÖNDE HAREKET EDEN ARACIN ÖN KAMERA GÖRÜNTÜSÜNDE
ARKA PLAN TAKİBİ

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

Şubat 2014 , 64 sayfa

Bu tez çalışması kapsamında, arka plan takip tekniği olarak kendine has özellikleri olan genişletilmiş Kalman olasılıksal hipotez yoğunluk süzgeci önerilmiştir. Hareket eden kameradan alınmış videolardaki arka planın değişken özellik göstermesi nedeniyle, arka plan takibi genellikle ardışık resimler kullanılarak ve piksel seviyesinde karşılaştırmalar yapılarak gerçekleştirilmektedir. Bununla birlikte, bazı yöntemler arka plan takibi için arka plandan elde edilen özellikleri de kullanmaktadır. Önerilen yöntemde arka plandan elde edilen belirli özelliklerin izlenmesi yaklaşımı kullanılmaktadır. Özellik olarak videodan elde edilen fotoğraflardaki köşe noktaları seçilmiştir. Bu köşe noktaları Harris köşe bulucu algoritması ile elde edilmektedir. Elde edilen köşelerin durum vektörlerini tahmin etmek ve güncellemek için doğrusal hareket modeli ve doğrusal olmayan ölçüm modeli türetilmiştir. Arka plan takibi için bu modelleri temel alan, değişken sayıdaki çoklu hedef takibinde ideale yakın bir çözüm üreten genişletilmiş Kalman olasılıksal hipotez yoğunluk süzgeci kullanılmaktadır. Önerilen metod, rastgele küme teoremi ve Kalman süzgeci yaklaşımını

kullanarak hedeflere ait yoğunlukları zaman içerisinde ilerletmektedir. Önerilen metodu gerçeklemek için MATLAB ortamı kullanılmıştır. Farklı durumlar için önerilen metod ile ilgili çeşilti deneyler yapılmış ve sonuçları açıklanmıştır. Sonuçlara göre belirli varsayımların varlığında, önerilen yöntemin klasik arka plan takip algoritmalarının yerine kullanılabildiği görülmüştür.

Anahtar Kelimeler: Arka Plan Takibi, Rastgele Sınırlı Setler, Olasılıksal Hipotezler Filtresi, Harris Köşe Bulucu, Genişletilmiş Hedef Takibi, Genişletilmiş Kalman

To my lovely fiancee and my family

ACKNOWLEDGMENTS

I wish to express my sincerest thanks to my supervisor Prof. Dr. Mübeccel Demirekler for her support, guidance and valuable contributions throughout the preparation of this thesis.

I would like to thank my colleagues in ASELSAN Inc. for their support.

I want to acknowledge the support of ASELSAN Inc. for the realization of this thesis.

I am grateful to Erdal Mehmetcik for his encouragement and support.

I would like to thank TÜBİTAK for its financial support throughout my master's studies.

I would like to forward my appreciation to my family for their continuous support throughout my life.

Finally, I have very special thanks for my lovely fiancee Büşra, for her endless love, patience, trust and understanding during this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

KF	Kalman Filter
PF	Particle Filter
RFS	Random Finite Set
NoT	Number of Targets
NoC	Number of Clutter
PM	Performance Measure
HCD	Harris Corner Detector
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
SoMG	Set of Merged Gaussians
CoS	Collection of All Subsets
PHD	Probablity Hypothesis Density
SoCC	Set of Cluster Center Candidates
PHDF	Probablity Hypothesis Density Filter
GMPHDF	Gaussian Mixture Probability Hypothesis Density Filter
EK-GMPHDF	Extended Kalman Gaussian Mixture Probablity Hypothesis Density Filter

CHAPTER 1

INTRODUCTION

The aim of this study is to track the "background" of a scene taken from a front camera of a non-maneuvering vehicle. Background tracking for this setup is a challenging problem especially when there are changes in the illumination, background geometry, motion etc. Such problems cannot be solved by simplistic, static-background models because of the non-stationary nature of the background, [1]. Various different methods exist in the literature to solve this type of problems [1], [2], [3], [4], [5]. Most of these methods are computationally very expensive. Alternatively, some methods use features or interest points of the background in order to decrease the computational load such as [6], [7], [8], [9] and [10].

In this thesis, we propose a method that utilizes the feature tracking approach to track the background of a video. The features are chosen as the corner points extracted from each frame of the video by using well known corner detection technique, known as the Harris corner detector [11]. Motion of the features is handled by a non-linear measurement and a linear motion models. Since the measurement model is non-linear, extended Kalman filter (EKF) can be utilized to handle the prediction and update stages for tracking a feature. Extended Kalman filter deals only with state estimation .In order to track multiple target, a method is needed for measurement assignment. Although the multiple target Bayesian filter is the optimal solution, it is computationally very expensive because multi target posterior density is propagated. A new method, the probability hypothesis density (PHD) filter, is proposed by Mahler [12], as an approximation of the multiple target Bayesian filter. The first order statistical moment of the state is propagated rather than the multiple target posterior density in the PHD filter (PHDF). Furthermore, data association is not needed for PHDF. Hence, the PHD filter is capable of multiple target tracking when a time varying number of targets and data association uncertainty exist. Using some assumptions, Vo and Ma [13] proposed a version of the PHD filter as Gaussian Mixture PHD filter, namely GMPHDF and given in [13]. The GMPHDF uses KF equations to propagate the co-variance matrix and mean vector of the state. Existence of the linear observation and motion models is precondition to use the GMPHDF. Since the observation model is non-linear in our problem, a specific version of the PHD filter is used. This specific filtering method is the same as the GMPHDF except that the new method uses the extended Kalman filter equations instead of Kalman filter equations. The specific filter is named as extended Kalman GMPHDF and denoted as EK-GMPHDF. The flowchart of the proposed method is given in Figure 1.1.

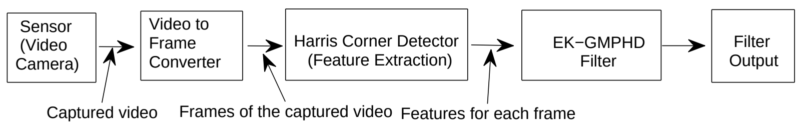

Figure 1.1: Flowchart of the proposed method

We flesh the proposed method out in the following chapters. After validation of the models and the proposed method, several experiments are done to investigate the performance of the proposed method. Lastly, the extracted features of the background are tracked by using the EK-GMPHDF to solve the defined problem.

1.1 Outline of thesis

In Chapter 2, the Bayesian filtering concept is briefly reviewed. Some commonly used filtering approaches are also given in this chapter.

In Chapter 3, the random set filtering approach, single and multiple target RFS are explained. The basics of the PHD, GMPHDF and the EK-GMPHDF that is used as the filtering approach of the proposed method in this thesis are also explained in detail.

In Chapter 4, the feature extraction method, the motion model and the measurement model are given.

The results obtained with the proposed method are elaborated in Chapter 5.

Finally, a brief summary of this study, the derived conclusions and the suggested future work are given in Chapter 6, the last chapter.

CHAPTER 2

BAYESIAN FILTERING

In this chapter a brief review of Bayesian filtering is given. As stated in [14], Bayesian signal processing is related with the estimation of the probability distribution of a random signal, to perform statistical inference. The sequential estimation of the system state at each time step according to received noisy measurement sequence is called *filtering* as stated in [15]. The system and measurement processes are modeled as probabilistic forms to apply filtering. Theoretically optimal and the commonly used filtering approach is the Bayesian approach. In Bayesian approach, system states are predicted based on process model and then updated by measurements. Bayesian approach recursively propagates the updated state estimate (also called the posterior pdf) using the Bayes' theorem given in (2.1).

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.1}$$

The posterior probability density uses all information about the system collected up to that time step. In a general filtering framework, the process model can be stated as in (2.2).

$$x_k = f_{k|k-1}(x_{k-1}, w_{k-1}) \tag{2.2}$$

where, $f_{k|k-1}$ is a known function that describes characteristics of the system, x_k is the system state and w_{k-1} is the process noise. Recursive update of the state x_k is done by using measurements $\mathbb{Z}_k = [Z_1, Z_2, ..., Z_k]^T$ at time k. Notice that, in this general form, no assumptions are made about the noise characteristics, for instance additive, multiplicative etc. The only assumption about the noise is that it is white. In other words, knowing w_{k-1} gives no information about w_k for all k. The measurements are related to the states at time k are given in (2.3).

$$Z_k = h_{k|k}(x_k, v_k) \tag{2.3}$$

where, $h_{k|k}$ is a known function that describes the relationship between the state (x_k) and the measurement (v_k) at time k. w_{k-1} and v_k are used to compensate the mismatch in the assumed and actual process and measurement models respectively. As it will be seen, the general form has no analytic solution for the functions (f, h) and noise types (w, v). To obtain a closed form analytic solution, general model is constrained to a specific case by using some approximations. For instance, the Kalman filter provides such a solution for linear process and measurement models in the presence of additive white Gaussian noise. In Bayesian approach, the posterior density, i.e $p(x_k \mid \mathbb{Z}_k)$ is estimated by using all received measurements up to time k, i.e. $\mathbb{Z}_k$. Prediction density, i.e. $p(x_k \mid \mathbb{Z}_{k-1})$, is calculated starting from the initial density $p(x_0 \mid Z_0)$ where x_0 is the initial state and Z_0 is the initial measurement set. Assuming that $p(x_{k-1} \mid \mathbb{Z}_{k-1})$ is known at time $k-1$. The prior density or prediction is obtained as given in [16] and follows;

$$p(x_k \mid \mathbb{Z}_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid \mathbb{Z}_{k-1}) dx_{k-1} \tag{2.4}$$

where, $p(x_k \mid x_{k-1})$ is an order one Markov process and known as *transition density*. This equation is sometimes called the *Chapman-Kolmogorov equation* [16]. The prediction density $p(x_k \mid \mathbb{Z}_{k-1})$ is updated after the measurement set Z_k is received at time k according to Bayes' rule as given in (2.5).

$$p(x_k \mid \mathbb{Z}_k) = p(x_k \mid Z_k, \mathbb{Z}_{k-1}) = \frac{p(Z_k \mid x_k, \mathbb{Z}_{k-1}) p(x_k \mid \mathbb{Z}_{k-1})}{p(Z_k \mid \mathbb{Z}_{k-1})} = \frac{p(Z_k \mid x_k) p(x_k \mid \mathbb{Z}_{k-1})}{p(Z_k \mid \mathbb{Z}_{k-1})} \tag{2.5}$$

where

$$p(Z_k \mid \mathbb{Z}_{k-1}) = \int p(Z_k \mid x_k)p(x_k \mid \mathbb{Z}_{k-1})dx_k \tag{2.6}$$

$p(\ Z_k \ \mid \ x_k\ )$ is known and determined by the measurement model. Although the Bayesian approach is theoretically optimal, the computational complexity of calculating the integrals may hinder the feasibility of this approach for practical implementation. Since these integrals are most of the time analytically intractable, certain numerical methods are needed. The number of the numerical calculations increases exponentially, if the dimension of the state (or measurement) vectors increases. In spite of this drawback, there are popular filtering methods that use the Baysesian approach for state estimation using some assumptions. In the following sections, two commonly used filtering methods, the Kalman filter (KF) [17] and the extended Kalman filter (EKF) [18] which use Bayesian approach, are explained. Other methods that are well studied in the literature are the unscented Kalman filter (UKF) [19] and the particle filter (PF) [20]. However in the scope of this thesis we will restrict ourselves to KF and EKF.

2.1 Kalman Filter

The Kalman filter (KF) proposed in [17] is the optimal filtering algorithm for recursive Bayesian state estimation under some restrictive assumptions [16]. The assumptions are that the posterior density is Gaussian, measurement and process noises are independent white noises with zero mean. The process and measurement models are assumed to be linear. KF propagates the covariance matrix and mean vector of the posterior density, since Gaussian density is completely characterized by its covariance matrix and mean vector. The equations of process and measurement models [21] that the Kalman filter uses are given in (2.7) and (2.8) respectively.

$$x_k = F_{k|k-1}x_{k-1} + w_{k-1} \tag{2.7}$$

$$z_k = H_{k|k}x_k + v_k \tag{2.8}$$

$F_{k|k-1}$ is an *nxn* linear transition matrix and $H_{k|k}$ is an *mxn* measurement matrix where n and m are dimensions of the state and measurement vectors respectively. w_{k-1} and v_k are additive white Gaussian noises with covariance matrices Q_{k-1} and R_k respectively, and given in the following equations.

$$w_{k-1} \sim N(x; 0, Q_{k-1}) \tag{2.9}$$

$$v_k \sim N(x; 0, R_k) \tag{2.10}$$

The notation shown in (2.11) will be used in this thesis for representation of any Gaussian density.

$$N(x; m, P) = \frac{1}{\sqrt{|\ 2\pi P\ |}} e^{-0.5(x-m)^T P^{-1}(x-m)} \tag{2.11}$$

In this representation; x is the argument, P is the covariance matrix and m is the mean vector. The prior and posterior densities are represented in (2.12) and (2.13) respectively.

$$P(x_k \mid \mathbb{Z}_{k-1}) = N(x_k; \tilde{x}_{k|k-1}, P_{k|k-1}) \tag{2.12}$$

$$P(x_k \mid \mathbb{Z}_k) = N(x_k; \tilde{x}_{k|k}, P_{k|k}) \tag{2.13}$$

The prior and posterior mean and the covariance matrices of the KF can be derived as in [16]. The prior covariance matrix and mean are given in (2.15) and (2.14) respectively.

$$x_{k|k-1} = F_{k|k-1}\ x_{k-1|k-1} \tag{2.14}$$

$$P_{k|k-1} = Q_{k|k-1} + F_{k|k-1} P_{k-1|k-1} F_{k|k-1}{}^T \tag{2.15}$$

The posterior covariance mean is given in (2.16) where the term $z_k - H_{k|k}x_{k|k-1}$ is innovation and the $K_{g,k}$ is Kalman gain.

$$x_{k|k} = x_{k|k-1} + K_{g,k}(z_k - H_{k|k}x_{k|k-1}) \tag{2.16}$$

The $K_{g,k}$ is given in (2.17).

$$K_{g,k} = P_{k|k}H_{k|k}{}^T S_k{}^{-1} \tag{2.17}$$

where the covariance matrix of the innovation is denoted by S_k and given as follows;

$$S_k = H_{k|k}P_{k|k}H_{k|k}{}^T + R_k \tag{2.18}$$

The posterior covariance matrix of the state is given in (2.19).

$$P_{k|k} = P_{k|k-1} - K_{g,k}S_k K_{g,k}^T \tag{2.19}$$

The P_k can be rewritten in terms of the prior covariance matrix according to (2.17) and (2.18) as

$$P_{k|k} = [I_{nxn} - K_{g,k}H_{k|k}]P_{k|k-1} \tag{2.20}$$

2.2 Extended Kalman Filter

In practice, process and measurement models are usually non-linear. The Kalman filter cannot be used for these cases because the linearity assumption of the Kalman Filter is no longer valid. Extended Kalman Filter (EKF) [18] is a suboptimal non-linear filter that uses linear approximations of the process and the measurement functions. The linear approximations of the non-linear functions are obtained by using the first terms of their Taylor series expansions instead of the non-linear functions. The assumptions about the process and the measurement noises that are explained in the Section 2.1 still hold. The non-linear process and measurement equations are given in (2.21) and (2.22) respectively.

$$x_k = f_{k|k-1}(x_{k-1}) + w_{k-1} \tag{2.21}$$

$$z_k = h_{k|k}(x_k) + v_k \tag{2.22}$$

$f_{k|k-1}$ and $h_{k|k}$ are the non-linear analytic functions. The linear approximations of $f_{k|k-1}$ and $h_{k|k}$ are

$$\tilde{F}_{k|k-1} = \left.\frac{\partial f_{k|k-1}(x_{k-1})}{\partial x_{k-1}}\right|_{x_{k-1}=x_{k-1|k-1}} \tag{2.23}$$

$$\tilde{H}_{k|k} = \left.\frac{\partial h_{k|k}(x_k)}{\partial x_{k-1}}\right|_{x_k=x_{k|k-1}} \tag{2.24}$$

where, $\tilde{H}_{k|k}$ and $\tilde{F}_{k|k-1}$ are Jacobians evaluated at the estimates of the state vector. Although not true, it is assumed that both the predicted and the filtered states are Gaussian. If the assumptions hold, the prior covariance matrix and mean can be written as:

$$x_{k|k-1} = f_{k|k-1}(x_{k-1|k-1}) \tag{2.25}$$

$$P_{k|k-1} = Q_{k|k-1} + \tilde{F}_{k|k-1} P_{k-1|k-1} \tilde{F}^T_{k|k-1} \tag{2.26}$$

The posterior covariance mean is given in (2.27) where the term $z_k - H_{k|k} x_{k|k-1}$ is innovation and the $K_{g,k}$ is Kalman gain.

$$x_{k|k} = x_{k|k-1} + K_{g,k}(z_k - h_{k|k}(x_{k|k-1})) \tag{2.27}$$

The $K_{g,k}$ is given as

$$K_{g,k} = P_{k|k-1}\tilde{H}_{k|k}^{T}{S_k}^{-1} \tag{2.28}$$

where the covariance matrix of the innovation is denoted by S_k. The equation for S_k is given below.

$$S_k = \tilde{H}_{k|k}P_{k|k-1}\tilde{H}_{k|k}^{T} + R_k \tag{2.29}$$

The posterior covariance matrix of the state is given in (2.19).

$$P_{k|k} = P_{k|k-1} - K_{g,k}S_kK_{g,k}^{T} \tag{2.30}$$

P_k can be rewritten in terms of the prior covariance matrix using (2.28) and (2.29) as

$$P_{k|k} = [I_{nxn} - K_{g,k}\tilde{H}_{k|k}]P_{k|k-1} \tag{2.31}$$

where I_{nxn} is an n by n identity matrix.

The extended Kalman filter works well, when the non-linear function is well characterized by a linear function around the state estimates. As explained before, the unscented Kalman filter and particle filter are the two other popular state estimation methods that can handle the non-linearities in the process and measurement models [16]. However these methods will not be investigated in the scope of this thesis.

CHAPTER 3

RANDOM SET FILTERING

Many different algorithms have been developed to track multiple targets so far. There have been an extensive study on multiple target tracking (see [22] and [23] for a detailed analysis on the subject). Recursive Bayesian nonlinear filtering is accepted as the theoretically optimal solution to track multiple targets in the presence of multiple sensors [12]. However, the computational load of this filter is challenging even for the single target tracking applications. In that case, the multiple target recursive Bayesian filter is not applicable unless the computational load of the filter can be handled in practice. The load can be decreased by using some approximatons like the constant-gain Kalman filter as explained in Section 2.1. The constant-gain KF that propagates the mean, is the fastest filtering approach for single target tracking.

Mahler [12] proposed a new method, namely Probability Hypothesis Density filter (PHDF), for tracking a time varying number of multi-target based on the propagation of the means in the presence of measurement and data association uncertainty, and false alarms. The mean of the multi-target posterior distribution called as the *PHD* which is a function whose integral in any region gives the mean value of the number of targets in the region. Peaks of the PHDs are used to estimate the states. The PHD filter models the states and the measurements as random finite sets (RFS) and applies PHD recursion to propagate the posterior intensity. PHD is used instead of the all multiple target posterior distribution because less computation. Although the computational advantages of the PHD filter, absence of a closed-form solution of the PHD filter is a drawback. In order to solve this problem, some assumptions given in Section 3.2 are used. Detailed explanation about the random finite set formulation is given below.

3.1 Multiple Target Filtering by using Random Finite Sets

In this section, theory of the random finite sets is explained to be used for multiple target tracking. According to random finite set theory, size of the measurement set and size of the state set may change for different time step k. A target that existed in the previous time step may survive or die. There are also new-born targets. If number of targets changes because of deaths or births, size of the target state changes. It is possible that sensors can generate one measurement, more than one measurement or no measurement for a target. Therefore, size of the measurement set is time-varying. Furthermore, measurements are indistinguishable and also include false alarms and noise. Only some of the measurements really belong to targets. Therefore states and measurements are represented by random finite sets and given in (3.1) and (3.2) respectively, where $CoS(X)$ is the collection of all subsets of X and $CoS(Z)$ is the collection of all subsets of Z.

$$X_k = [x_{k,1}, ..., x_{k,NoT_k}]^T \in CoS(X) \tag{3.1}$$

$$Z_k = [z_{k,1}, ..., z_{k,NoM_k}]^T \in CoS(Z) \tag{3.2}$$

As mention above, a target that existed in previous time step $k-1$ may survive or die at time step k and the process is characterized by a probability as known as survival probability, i.e. $p_{S,k}$. The survival probability is a function of the target states x_{k-1} and given as $p_{S,k}(x_{k-1})$. Since a target survives with the probability $p_{S,k}$, it dies with the probability $1 - p_{S,k}(x_{k-1})$ at time k. An RFS model, i.e. $S_{k|k-1}(x_{k-1})$, is written according to the behavior of $x_{k-1} \in X_{k-1}$ at time step k and is given following.

$$S_{k|k-1}(x_{k-1}) = \begin{cases} x_k, & \text{if target survives} \\ \emptyset, & \text{if target dies} \end{cases}$$

This is done for each target that exists at the previous time step $k-1$. Size of the target state may also change with births at time k in addition to survivals and deaths. New-born and spawned targets form the births. A target may spawn at time k from a target that existed at previous time step $k-1$. For instance, a missile is launched from a war craft while the war craft is flying results in the birth of a new target. The other birth type is spontaneous birth of a target at time k. The targets are known as *new-born targets*. Consequently, the multiple target state is a union of the all survival, spawned and new-born targets at time step k and is given in (3.3). $S_{k|k-1}(x_\ell)$ is the RFS of the survival targets, $Sp_{k|k-1}(x_\ell)$ is the RFS of spawned targets and $N_k(x_k)$ is the RFS of new-born targets. In (3.3), RFSs are independent of each other and $Sp_{k|k-1}(x_\ell)$ and $N_k(x_k)$ are problem dependent where $x_\ell \in X_{k-1}$ and L is equal to size of the target state (X_k).

$$X_k = \left[\bigcup_{\ell=1}^{L} S_{k|k-1}(x_\ell) \right] \cup \left[\bigcup_{\ell=1}^{L} Sp_{k|k-1}(x_\ell) \right] \cup N_k(x_k) \tag{3.3}$$

At each time step k, a measurement set (Z_k) is produced by the sensors for the multiple targets. The targets are detected with the probability $p_{D,k}(x_k)$ and missed with the probability $1 - p_{D,k}(x_k)$. Since the measurement set contains true target measurements and false alarms (clutter), size of the Z_k is time-varying. An RFS corresponding to target state is produced, i.e. M_k, where

$$M_k(x_k) = \begin{cases} m_k, & \text{if target is detected} \\ \emptyset, & \text{if target is not detected} \end{cases}$$

Multiple target measurement RFS (z_k) is the union of measurements and clutter as given in (3.4), where C_k is problem dependent clutter RFS.

$$Z_k = \left[\bigcup_{x \in X_k} M_k(x) \right] \cup C_k \tag{3.4}$$

Multiple target transition density $(f_{k|k-1}(X_k \mid X_{k-1}))$ can be written using (3.3) as stated in [12], [24]. The multiple target transition density $(f_{k|k-1})$ is a function or matrix that characterizes the changes of the states from time step $k-1$ to time step k. According to relationship between the previous state (X_{k-1}) and the current state (X_k), $f_{k|k-1}$ can be linear or non-linear.

Similarly, multiple target likelihood density $g_k(Z_k \mid X_k)$ can be written using (3.4). The multiple target likelihood density $g_k(Z_k \mid X_k)$ is a function or matrix that characterizes the prediction of the measurement set from states of the targets at that time step k. The $g_k(Z_k \mid X_k)$ can be linear or non-linear according to measurement model.

As a consequence, multiple target prediction density and the multiple target posterior density can be written using Bayes' recursion, and given in (3.5) and (3.6) respectively.

$$p_{k|k-1}(X_k \mid \mathbb{Z}_{k-1}) = \int f_{k|k-1}(X_k \mid X)\, p_{k-1}(X \mid \mathbb{Z}_{k-1})(dX) \tag{3.5}$$

$$p_k(X_k \mid \mathbb{Z}_k) = \frac{g_k(Z_k \mid X_k)\, p_{k|k-1}(X_k \mid \mathbb{Z}_{k-1})}{\int g_k(Z_k \mid X)\, p_{k|k-1}(X \mid \mathbb{Z}_{k-1})\,(dX)} \tag{3.6}$$

3.2 Probability Hypothesis Density Filter

It is hard to cope with the complexity of the joint multiple target likelihood when the number of the targets increases in the multiple target tracking problems [25]. The PHD filter of Mahler is based on recursive propagation of the intensities [12]. An intensity is a non-negative function and represented by γ. Integration of γ over any region that belongs to state space X gives the expected number of targets in that region. In PHD filter, all the RFSs are modeled as Poisson RFSs that are completely characterized by their intensities [13]. The parameters used in PHD filter are given following.

$p_{S,k}$	:	survival probability
$p_{D,k}$	:	detection probability
n_k	:	intensity of the new-born targets
$sp_{k\|k-1}$	:	intensity of the spawned targets
c_k	:	intensity of clutter
$\gamma_{k\|k-1}$	:	prior intensity
γ_k	:	posterior intensity

PHD filter uses some assumptions which are used by almost all tracking algorithms. The first assumption is that the clutter is Poisson and independent from targets. The second assumption is that predicted number of targets is also Poisson [13]. The third assumption is that statistics of targets and observations are independent of each other. Based on these assumptions, prior density is calculated by using the intensity of the survived targets, intensity of the spawned targets and intensity of the new-born targets. Integration of the density over the surveillance region gives the expected number of target at that region. Hence, integrations of the intensities mentioned above are calculated separately and then summed to find the prior intensity ($\gamma_{k|k-1}(x)$), i.e. predicted number of targets at that region [26], as given in (3.7). After that, the posterior intensity ($\gamma_k(x)$) is calculated by the summation of no measurement update and measurement update as given in (3.10). No measurement update is done if the target is not detected and given in (3.8). If the target is detected then the measurement update is done and as given in (3.9).

$$\gamma_{k|k-1}(x) = \left(\int p_{S,k}(\delta)\, f_{k|k-1}(x \mid \delta)\, \gamma_{k-1}(\delta)\, d\delta \right) + \left(\int sp_{k|k-1}(x \mid \delta)\, \gamma_{k-1}(\delta)\, d\alpha \right) + n_k(x) \tag{3.7}$$

$$\text{no measurement update : } [\ 1 - p_{D,k}(x)\ ]\ \gamma_{k|k-1}(x) \tag{3.8}$$

$$\text{measurement update : } \sum_{z \in Z_k} \frac{p_{D,k}(x)\, g_k(z \mid x)\, \gamma_{k|k-1}(x)}{c_k(z) + \int p_{D,k}(\beta)\, g_k(z \mid \delta)\, \gamma_{k|k-1}(\delta)\, d\delta} \tag{3.9}$$

$$\gamma_k(x) = [\ 1 - p_{D,k}(x)\ ]\ \gamma_{k|k-1}(x) + \sum_{z \in Z_k} \frac{p_{D,k}(x)\ g_k(z \mid x)\ \gamma_{k|k-1}(x)}{c_k(z) + \int p_{D,k}(\delta)\ g_k(z \mid \delta)\ \gamma_{k|k-1}(\delta)\ d\delta} \tag{3.10}$$

3.2.1 Gaussian Mixture Probability Hypothesis Density Filter

The Gaussian mixture PHD filter, GMPHDF for short, is proposed in [13], [27] and [26]. The GMPHDF is implemented by [27] and [26] for several extended multiple target tracking applications. As explained before, the PHD filter does not have a closed form solution. Based on some assumptions, Vo and Ma [13] proposed a closed form solution for the GMPHDF to track multiple targets. The first assumption is that process and measurement models are linear Gaussian models as given in (3.11) and (3.12) respectively.

$$f_{k|k-1}(x \mid \xi) = N(x; F_{k|k-1}\xi, Q_{k|k-1}) \tag{3.11}$$

$$g_k(z \mid x) = N(z; H_{k|k}x, R_k) \tag{3.12}$$

where

$F_{k\|k-1}$	:	state transition matrix
$Q_{k\|k-1}$	:	the covariance matrix of the process noise
$H_{k\|k}$	:	observation matrix
R_k	:	the covariance matrix of the measurement noise

The second assumption is that the intensity of the new-born target RFS is modeled as the summation of Gaussians and is given in (3.13). The $m_{n,k}^{(j)}$ indicates the maximum of the new-born target intensity where $J_{n,k}$ is maximum number of the new-born targets. In other words, the first show up probability of the new-born targets is maximum at $m_{n,k}^{(j)}$. For instance, harbors for ships, parking lots for cars etc. $P_{n,k}^{(j)}$ denotes the dispersion of the new-born targets' intensity around the maxima. The $w_{n,k}^{(j)}$ is the expected number of the new-born targets at the maximum, i.e. $m_{n,k}^{(j)}$.

$$n_k(x) = \sum_{j=1}^{J_{n,k}} w_{n,k}^{(j)} \, N(x; m_{n,k}^{(j)}, P_{n,k}^{(j)}) \tag{3.13}$$

The third assumption is that intensity of the spawned target RFS are modeled as summation of Gaussians and is given in (3.14) in common with the new-born target intensity. The term $F_{sp,k|k-1}^{(i)}\delta + d_{sp,k|k-1}^{(i)}$ indicates the expected state of the spawned target and is a function of the previous state δ. A target spawns at neighborhood of its parent. For instance, when a missile is launched from an aircraft it means that the missile spawned from the parent, i.e. the aircraft. The $Q_{sp,k|k-1}^{(i)}$ denotes the dispersion of the spawned targets' intensity around the parent. The $w_{sp,k}^{(i)}$ is the expected number of the spawned targets from the parent δ.

$$\gamma_{sp,k|k-1}(x \mid \delta) = \sum_{i=1}^{J_{sp,k}} w_{sp,k}^{(i)} \, N(x; F_{sp,k|k-1}^{(i)}\delta + d_{sp,k|k-1}^{(i)}, Q_{sp,k|k-1}^{(i)}) \tag{3.14}$$

The fourth assumption is that the detection and survival probabilities are time invariant.

$$p_{S,k}(x) = p_S \tag{3.15}$$

$$p_{D,k}(x) = p_D \tag{3.16}$$

Before proceeding further, we would like to mention the two following facts that are used in derivations.

Fact 1: If P and R are positive definite and symmetric matrices, multiplication of the two Gaussians can be written by using the derivations which is explained in the Section 3.8 of [16] as:

$$N(x; Hx, R) \, N(x; m, P) = \psi(z) \, N(x; \hat{m}, \hat{P}) \tag{3.17}$$

where

$$\psi(z) = N(z; Hm, R + HRH^T) \tag{3.18}$$

$$\hat{m} = m + C\,[z - Hm] \tag{3.19}$$

$$\hat{P} = [I - CH]\,P \tag{3.20}$$

$$C = PH^T [\,HPH^T + R\,]^{-1} \tag{3.21}$$

H^T indicates the transpose of the matrix H and R^{-1} indicates the inverse of the matrix R.

Fact 2: If P and Q are symmetric and positive definite matrices, integral of the multiplication of two Gaussians can be written as another Gaussian [13] using *Fact 1*, and is given in (3.22). In addition, dimensions of the given matrices, i.e. P, m, Q, d, F, must be suitable.

$$\int N(\alpha; m, P)\, N(x; F\alpha + d, Q)\, d\alpha = N(x; Fm + d, Q + FPF^T) \tag{3.22}$$

We proceed by computing the predicted intensity assuming that the posterior intensity is a Gaussian mixture as given below.

$$\gamma_{k-1}(x) = \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)}\, N(x; m_{k-1}^{(j)}, P_{k-1}^{(j)}) \tag{3.23}$$

At time step k, the predicted intensity is derived by using (3.22) and by substituting (3.11), (3.15), (3.14), (3.13), (3.23) into (3.7). The predicted intensity is given in (3.24). The predicted intensity has 3 components that correspond to survival, spawning and new-born targets.

$$\gamma_{k|k-1}(x) = \gamma_{s,k|k-1}(x) + \gamma_{sp,k|k-1}(x) + n_k(x) \tag{3.24}$$

where $\gamma_{s,k|k-1}$ is the intensity of the survival targets, $\gamma_{sp,k|k-1}$ is the intensity of the spawned targets and $n_k(x)$ is the intensity of the new-born targets. The predicted intensity of the survival targets $\gamma_{s,k|k-1}$ is given in (3.25).

$$\gamma_{s,k|k-1} = p_S \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \, N(x; m_{s,k|k-1}^{(i)}, P_{s,k|k-1}^{(i)}) \tag{3.25}$$

where

$$m_{s,k|k-1}^{(i)} = F_{k|k-1} \, m_{k-1}^{(i)} \tag{3.26}$$

$$P_{s,k|k-1}^{(i)} = Q_{k|k-1} + F_{k|k-1} P_{k-1}^{(i)} F_{k|k-1}^{T} \tag{3.27}$$

The predicted intensity of the spawned targets is given in (3.28).

$$\gamma_{sp,k|k-1} = \sum_{i=1}^{J_{k-1}} \sum_{t=1}^{J_{t-1}} w_{k-1}^{(i)} \, w_{sp,k-1}^{(t)} \, N(x; m_{sp,k|k-1}^{(i,t)}, P_{sp,k|k-1}^{(i,t)}) \tag{3.28}$$

where

$$m_{sp,k|k-1}^{(i,t)} = F_{sp,k|k-1}^{(t)} m_{k-1}^{(i)} + d_{sp,k-1}^{(t)} \tag{3.29}$$

$$P_{sp,k|k-1}^{(i,t)} = Q_{sp,k|k-1}^{(t)} + F_{sp,k|k-1}^{(t)} P_{sp,k-1}^{(i)} (F_{sp,k|k-1}^{(t)})^{T} \tag{3.30}$$

If the previously explained four assumptions hold, then the prior intensity $\gamma_{k|k-1}$ is a Gaussians mixture and is given in (3.31).

$$\gamma_{k|k-1} = \sum_{j=1}^{J_{k|k-1}} w_{k|k-1}^{(j)} \, N(x; m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) \tag{3.31}$$

The $J_{k|k-1}$ is the predicted number of Gaussians, $w_{k|k-1}^{(j)}$ is the predicted weight, $m_{k|k-1}^{(j)}$ is the mean of the prediction and $P_{k|k-1}^{(j)}$ is the covariance matrix of the predictions.

Once the prediction intensity is found, the posterior intensity can be found using the measurements and predictions. In a similar manner, the posterior intensity can be written as a sum of Gaussians and is given in (3.32).

$$\gamma_k = \gamma_{NMU} + \gamma_{MU} \tag{3.32}$$

Equation 3.32 is written as the summation of the *no measurement update intensity (γ_{NMU})* and *measurement update intensity (γ_{MU}).* The no measurement update intensity is calculated by using the prior intensity in the absence of measurement and is given in (3.33).

$$\gamma_{NMU} = (1 - p_D)\,\gamma_{k|k-1}(x) \tag{3.33}$$

The measurement update is written as the summation of the all the intensities that are calculated for each measurement $z \in Z_k$ and is given in (3.34). Z_k is the measurement set at time step k.

$$\gamma_{MU} = \sum_{z \in Z_k} \gamma_{D,k}(x \mid z) \tag{3.34}$$

The intensity of measurement update $\gamma_{D,k}(x \mid z)$ is calculated for each measurement as given in the following equations.

$$\gamma_{D,k}(x \mid z) = \sum_{i=1}^{J_{k|k-1}} w_k^{(i)}(z)\, N(x; m_{k|k}^{(i)}(z), P_{k|k}^{(i)}) \tag{3.35}$$

where

$$w_k^{(i)}(z) = \frac{p_D\, w_{k|k-1}^{(i)}\, \psi_k^{(i)}(z)}{c_k(z) + p_D\, \sum_{t=1}^{J_{k|k-1}} w_{k|k-1}^{(t)}\, \psi_k^{(t)}(z)} \tag{3.36}$$

$$m_{k|k}^{(i)}(z) = m_{k|k-1}^{(i)} + K_{g,k}^{(i)}\,(z - H_{k|k}\, m_{k|k-1}^{(i)}) \tag{3.37}$$

$$P^{(i,t)}_{k|k} = [\ I - K^{(i)}_{g,k} H_{k|k}\ ]\ P^{(i)}_{k|k-1} \tag{3.38}$$

$$K^{(i)}_{g,k} = P^{(i)}_{k|k-1}\ H^T_{k|k}\ (\ H_{k|k}\ P^{(i)}_{k|k-1}\ H^T_{k|k} + R_k\ )^{-1} \tag{3.39}$$

Note that $K^{(i)}_{g,k}$ is the Kalman gain and $c_k(z)$ is the clutter intensity at time step k.

3.2.1.1 Truncation of Gaussian Terms

The number of Gaussian terms which are used to represent the posterior intensity increases to a value given in (3.40). This makes the problem intractable. Hence a truncation algorithm is required to decrease the number of the Gaussian terms. At time k, the complexity of the posterior intensity increases without bound and given in (3.40) where NoM_k is the number of measurements Z_k.

$$O(\ J_{k-1}\ NoM_k\ ) = O(\ J_{k-1}(\ 1 + J_{b,k}\ ) + J_{n,k}\ )\ (\ 1 + NoM_k\ ) \tag{3.40}$$

The truncation algorithm explained in [13] is basically formed by *pruning* and *merging*. Elimination of the Gaussian terms is called as *pruning*. Pruning algorithm eliminates the Gaussian terms whose weights are less than a pruning threshold T_p. After this elimination, the pruned posterior intensity $\gamma_{p,k}$ is used instead of the posterior density γ_k which is given by

$$\gamma_{p,k}(x) = \mu \sum_{j=N_p}^{J_k} w^{(j)}_k\ N(x; m^{(j)}_k, P^{(j)}_k) \tag{3.41}$$

where, $j = 1, ..., N_p - 1$ are the indices for the Gaussians terms whose weights are smaller than T_p. μ is a scaling term to normalize the weights after pruning operation, to ensure that the sum of the weights add up to the sum of the weights before pruning. The equation for μ is given as follows.

$$\mu = \frac{\sum_{j=1}^{J_k} w_k^{(i)}}{\sum_{j=N_p}^{J_k} w_k^{(i)}} \tag{3.42}$$

Joining the similar Gaussian terms is called *merging*. Statistically similar Gaussians are merged by using a clustering method that chooses the largest weighted Gaussians as the cluster center. Each Gaussian with a weight greater than the threshold T_p is selected as a cluster center. Others are distributed among these centers according to their distances from these centers. In (3.43), $SoCC$ denotes the set of cluster center candidates. The i term denotes the cluster center in (3.44).

$$SoCC = \{j \mid w_k^{(j)} > T_p\} \tag{3.43}$$

$$i = argmax_{j \in SoCC}(w_k^{(j)}) \tag{3.44}$$

After the cluster center is determined, distance between the center and the other tracks are calculated. If the merging threshold, i.e. T_m, is greater than the distance, then the Gaussians are merged. The merged Gaussians form the set of merged Gaussians, namely $SoMG$ and is given below.

$$SoMG = \{j = SoCC \mid (m_k^{(j)} - m_k^{(i)})^T \; (P_k^j)^{-1} \; (m_k^{(j)} - m_k^{(i)}) \leq T_m\} \tag{3.45}$$

The new weights, means and covariance matrices are calculated as given below.

$$\bar{w}_k^{(l)} = \sum_{j \in SoMG} w_k^{(j)} \tag{3.46}$$

$$\bar{m}_k^{(l)} = \frac{1}{\bar{w}_k^{(l)}} \sum_{j \in SoMG} w_k^{(j)} \; x_k^{(j)} \tag{3.47}$$

$$\bar{P}_k^{(l)} = \frac{1}{\bar{w}_k^{(l)}} \sum_{j \in SoMG} w_k^{(j)} \left(P_k^{(j)} + (\bar{m}_k^{(l)} - m_k^{(j)}) \, (\bar{m}_k^{(l)} - m_k^{(j)})^T \right) \tag{3.48}$$

After that a new center is determined and the process is repeated until all the tracks are put into process. Finally the state estimation is done with the truncated Gaussians that have weights are greater than target threshold T_T. If the weight of a target is greater than T_T, the target is considered a real target and processed for at the state estimation block. As explained in [13], the pseudo code of the truncation algorithm is given below.

Algorithm 1 Truncation of the Gaussian Terms

procedure TRUNCATION ALGORITHM($w_k^{(j)}$, $m_k^{(j)}$, $P_k^{(j)}$, T_w, T_m)
    SoCC $\leftarrow \{j \mid w_k^{(j)} > T_P\}$
    repeat
        $\ell \leftarrow \ell + 1$
        $i \leftarrow argmax_{i \in SoCC} w_k^{(j)}$
        $SoMG \leftarrow \{i \in SoCC \mid (m_k^{(j)} - m_k^{(i)})^T (P_k^{(j)})^{-1} (m_k^{(j)} - m_k^{(i)}) \leq T_m\}$
        $\bar{w}_k^{(\ell)} = \sum_{i \in SoMG} w_k^{(i)}$
        $\bar{m}_k^{(\ell)} = \frac{1}{\bar{w}_k^{(\ell)}} \sum_{i \in SoCC} w_k^{(j)} x_k^{(j)}$
        $\bar{P}_k^{(\ell)} = \frac{1}{\bar{w}_k^{(\ell)}} \sum_{i \in SoCC} w_k^{(j)} (P_k^{(j)} + (\bar{m}_k^{(\ell)} - m_k^{(i)})(\bar{m}_k^{(\ell)} - m_k^{(i)})^T)$
        $SoCC \leftarrow SoCC/SoMG$
    until $SoCC \rightarrow \emptyset$
end procedure

3.2.2 Extended Kalman Gaussian Mixture Probability Hypothesis Density Filter

The extended Kalman GMPHDF, namely EK-GMPHDF, is proposed in [13] and [26]. The main differences between EK-GMPHDF and GMPHDF filter are the given process and measurement models. The process model and/or measurement model are non-linear in the EK-GMPHDF, whereas they are linear in the GMPHDF. The functions of the process and the measurement models are $f_{k|k-1}(x_{k-1}, \beta_{k-1})$ and $h_{k|k}(x_k, \epsilon_k)$ and are given in below.

$$x_k = f_{k|k-1}(x_{k-1}, \beta_{k-1}) \tag{3.49}$$

$$z_k = h_{k|k}(x_k, \epsilon_k) \tag{3.50}$$

The time invariance assumption of the survival probability p_S and the detection probability p_D still holds. Additionally, the new-born target intensity and the spawned target intensity are still assumed Gaussians. The EK-GMPHDF is basically a special version of the GMPHDF. The prior intensity of EK-GMPHDF is approximated as a summation of Gaussians and is given in following.

$$\gamma_{EK,k|k-1} \rightarrow \gamma_{s,k}(x) + \gamma_{sp,k}(x) + n_k(x) \tag{3.51}$$

where $\gamma_{s,k}$ is the approximated intensity of the survival targets, $\gamma_{sp,k}$ is the approximated intensity of the spawned targets and n_k is the intensity of the new-born targets. The survival intensity is approximated as

$$\gamma_{s,k|k-1} \rightarrow p_S \sum_{i}^{J_{k-1}} w_{s,k|k-1}^{(i)} \, N(x; m_{s,k|k-1}^{(i)}, P_{s,k|k-1}^{(i)}) \tag{3.52}$$

where

$$w_{s,k|k-1}^{(i)} = p_S \; w_{s,k-1|k-1}^{(i)} \tag{3.53}$$

$$m_{s,k}^{(i)} = f_{k|k-1}(m_{s,k-1|k-1}^{(i)}, 0) \tag{3.54}$$

$$P_{s,k|k-1}^{(i)} = G_{k-1}^{(i)} Q_{k-1} (G_{k-1}^{(i)})^T + F_{k|k-1}^{(i)} P_{k-1}^{(j)} (F_{k|k-1}^{(i)})^T \tag{3.55}$$

The $F_{k|k-1}$ and the G_{k-1} are the Jacobians of $f_{k|k-1}$ and are given by

$$F_{k|k-1}^{(i)} = \left. \frac{\partial f_{k|k-1}(x_{k-1}, 0)}{\partial x_{k-1}} \right|_{x_{t-1} = m_{k-1}^{(i)}} \tag{3.56}$$

$$G_{k-1}^{(i)} = \left. \frac{\partial f_{k|k-1}(m_{k-1}^{(i)}, \beta_{k-1})}{\partial \beta_{k-1}} \right|_{\beta_{t-1}=0} \tag{3.57}$$

At time k, $J_{sp,k}$ number of Gaussian components are produced by each Gaussian component that existed at time $k-1$. The spawned target intensity is given as follows.

$$\gamma_{sp,k|k-1}(x \mid m_{k-1}^{(j)}) = \sum_{t=1}^{J_{sp,k}} w_{sp,k}^{(t)} \, N(x; F_{sp,t-1}^{(t)\delta} + d_{sp,t-1}^{(t)}, Q_{sp,t-1}^{(t)}) \tag{3.58}$$

Then, the spawned target intensity approximates as

$$\gamma_{sp,k|k-1}(x) \rightarrow \sum_{i=1}^{J_{k-1}} \sum_{t=1}^{J_{sp,k}} w_{k-1}^{(i)} \, w_{sp,k}^{(t)} \, N(x; m_{sp,k|k-1}^{(i,t)}, P_{sp,k|k-1}^{(i,t)}) \tag{3.59}$$

where

$$m_{sp,k|k-1}^{(i,t)} = F_{sp,t-1}^{(t)} m_{k-1}^{(t)} + d_{sp,t-1}^{(t)} \tag{3.60}$$

$$P_{b,k|k-1}^{(i,t)} = H_{b,k-1}^{(t)} Q_{b,k-1} (H_{b,k-1}^{(t)})^T + F_{b,k-1}^{(i)} P_{b,k-1}^{(i)} (F_{b,k-1}^{(i)})^T \tag{3.61}$$

The posterior intensity of the EK-GMPHDF approximates PHD as

$$\gamma_{EK,k}(x) \rightarrow (\, 1 - p_{D,k} \,)\, \gamma_{k|k-1}(x) + p_D \, \gamma_{D,k}(x \mid z) \tag{3.62}$$

where, the parameters are

$$\gamma_{D,k}(x \mid z) = \sum_{j=1}^{J_{k|k-1}} \frac{w_{k|k-1}^{(j)} \, \phi_k^{(j)}(z)}{c_k(z) + p_D \sum_{t=1}^{J_{k|k-1}} w_{k|k-1}^{(t)} \, \phi_k^{(t)}} N(x; m_{k|k}^{(j)}, P_{(k|k)}^{(j)}) \tag{3.63}$$

$$m_{k|k}^{(j)}(z) = m_{k|k}^{(j)} + K_k^{(j)} \,(\, z - h_k \,(m_{k|k}^{(j)}\,)\,) \tag{3.64}$$

$$\phi_k^{(i)}(z) = N(z;\; \eta_{k|k-1}^{(j)},\; R_k + (H_k^{(j)})^T P_{(k|k)}^{(j)} H_k^{(j)}\,) \tag{3.65}$$

$$H_{k|k}^{(i)} = \left. \frac{\partial(\ h_{k|k}(x_k, 0)\ )}{\partial x_k} \right|_{x_k = m_{k|k-1}^{(i)}} \tag{3.66}$$

$$P_{k|k}^{(i)} = (\ I - K_k^{(i)} H_{k|k}^{(i)}\ )\ P_{k|k-1}^{(i)} \tag{3.67}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)}\ (H_{k|k}^{(i)})^T\ (S_k^{(i)})^{-1} \tag{3.68}$$

$$S_k^{(i)} = H_{k|k}^{(i)}\ P_{k|k-1}^{(i)}\ (H_{k|k}^{(i)})^T \tag{3.69}$$

CHAPTER 4

MODELING OF THE TRACKING PROBLEM

The difficulty in obtaining a model for the motion of the objects on the image plane is the absence of the range information. Due to the lack of this information it becomes difficult to relate the real position of the object to the corresponding pixel in the image. To overcome this difficulty we made an assumption that all background objects appear first around the vanishing point which is at a constant point on the image plane. Figure 4.1 shows the defined variables related with the appearance of a background object on the image plane under the above assumptions.

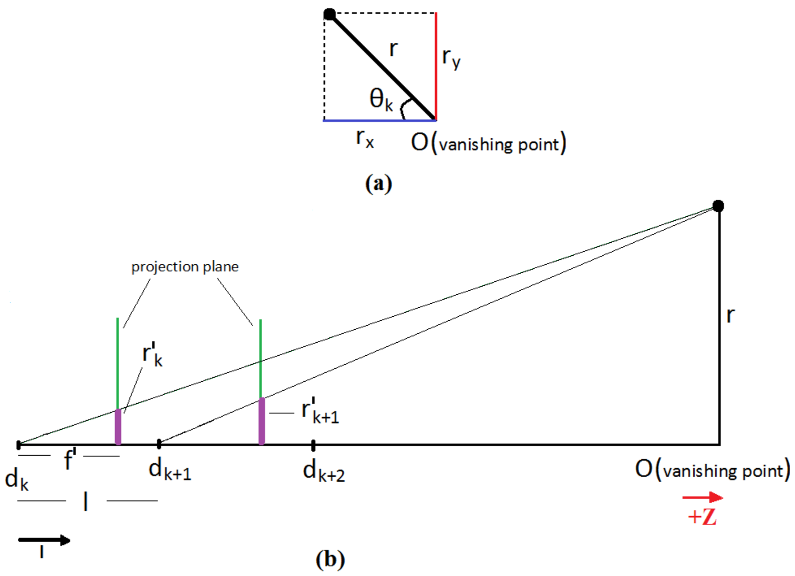

Figure 4.1: The schematic representation of the relative motion of the background object on the image plane. (a) $x - y$ plane, (b) $z - r$ plane.

The camera is assumed to be moving with almost constant velocity along the $+z$-axis which is perpendicular to the image plane. The background object (shown by a small circle in Figure 4.1) is assumed to be r meters away from the z-axis. The motion is related to the real motion of the camera that is represented by two variables, distance between the camera and the position of the background object projected onto the z axis, i.e. d_k, and velocity of the camera, i.e. l_k. When a background object is projected onto the image plane, it is r'_k pixels away from the vanishing point O. The image plane is f' meters far from the camera. Designations of the symbols which are used in the motion model are given below.

O	:	vanishing point of the image plane
ℓ	:	velocity of the vehicle or camera
f'	:	focal length of the camera
d_k	:	the actual distance between the camera and the position of the object projected onto the z axis at time k
d_0	:	the actual initial distance between the camera and the position of the object projected onto the z axis at time $k = 0$
r'_k	:	projection of the object onto the image plane at time k
r	:	the actual radial distance between the object and the z axis
θ_k	:	angle between the object and the x coordinate at time k

Note that the actual distance d_k is assumed to be a function of r'_k. This assumption corresponds to the assumption that all objects are seen for the first time at the vanishing point which is fixed for the given camera. However, the objects may not be recognized by Harris corner detector (HCD). If they are recognized at r'_k, then they have the same motion characteristics with the ones starting at O. The relationship between d_k and r'_k is given below.

$$d_k = \frac{f'\, r}{r'_k} \tag{4.1}$$

where f' and r are constants. Due to the lack of the range information it is not possible to obtain true values for $(f' r)$ multiplication. During testing, we have used a constant value, namely f for this term. Hence the relationship between d_k and r'_k becomes

$$d_k = \frac{f}{r'_k} \tag{4.2}$$

4.1 Motion Model

The motion model is the model of the motion of a stationary object as observed on the image plane while a video camera is approaching to it. The mathematical model used is based on the assumption that the camera is moving in +z direction with a constant velocity. The first parameter of the motion model is the velocity that is assumed constant and denoted as l_k at time k. The second parameter of the motion model is the actual distance between the camera and the object. The distance decreases with the velocity of the train or camera (l_k) at each time step and denoted as d_k at time k. The last parameter of the model is the angle, namely θ_k, between the object and the x-axis. θ_k is also assumed to be constant in time. Based on the given information above, the state equations of the model are given in (4.3), (4.4) and (4.5) where w^l_k, w^d_k and w^θ_k are white independent Gaussian processes.

$$l_{k+1} = l_k + w^l_k \tag{4.3}$$

$$d_{k+1} = d_k - l_k + w^d_k \tag{4.4}$$

$$\theta_{k+1} = \theta_k + w^\theta_k \tag{4.5}$$

Equations (4.3), (4.4) and (4.5) are the state equations. The measurements are the (x_k, y_k) pixel positions of the object projected onto the image plane at time step k. The measurements can be written in terms of the states as given in (4.6) and (4.7).

$$x_k = \frac{f\,cos(\theta_k)}{d_k} \tag{4.6}$$

$$y_k = \frac{f\,sin(\theta_k)}{d_k} \tag{4.7}$$

Note also that, the states can be written in terms of the measurements and are given in (4.8) and (4.9).

$$d_k = \frac{f}{\sqrt{x_k^2 + y_k^2}} \tag{4.8}$$

$$\theta_k = atan(\frac{y_k}{x_k}) \tag{4.9}$$

The model is tested by comparing the real data (b) that is extracted from [28] and the generated synthetic data (a) based on the motion model. The results are shown in Figures 4.2 and 4.3. It is clearly seen that the motion model is suitable according to the graphs shown in the following figures.

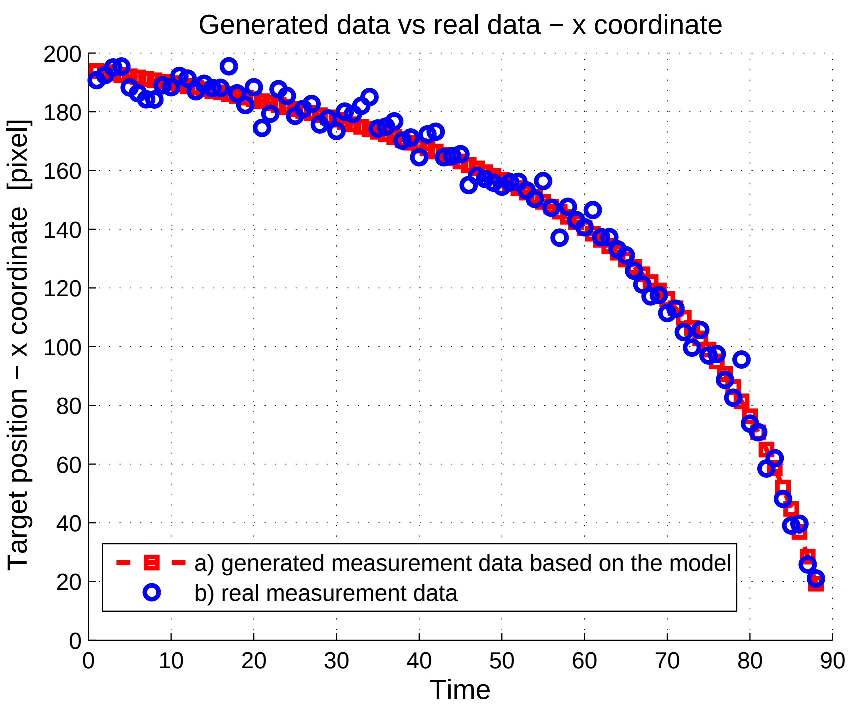

Figure 4.2: Generated data based on the model vs real data taken from the video sequence manually - x axis

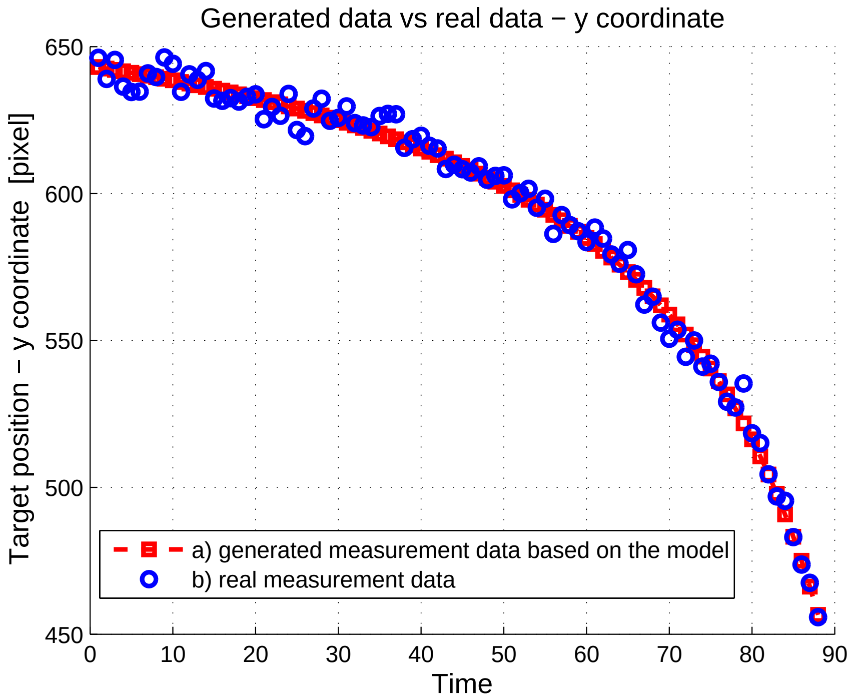

Figure 4.3: Generated data based on the model vs real data taken from the video sequence manually - y axis

4.2 Measurement Model

Measurement model is used to define the relationship between states and the measurements. In other words, measurements are predicted by using states based on the measurement model. The measurements are the corner points of the background objects. The corner points (measurements) are extracted by using the HCD, which will be explained in the next sections. Equations of the measurement model are given in (4.10) and (4.11).

$$h_k^x = \frac{f\ cos(\theta_k)}{d_k} \tag{4.10}$$

$$h_k^y = \frac{f\ sin(\theta_k)}{d_k} \tag{4.11}$$

where

h_k^x : x coordinate of the measurement at time k
h_k^y : y coordinate of the measurement at time k
v_k^x : measurement noise for x-coordinate
v_k^y : measurement noise for y-coordinate

4.3 Feature Extraction Method

In this thesis, our aim is to extract isolated points as features. The feature extraction method must be invariant to rotation, intensity shift or scaling. These requirements are satisfied by the popular interest point detector, namely the Harris corner detector (HCD) [29]. The HCD will be explained next.

4.3.1 Harris Corner Detector

Harris corner detector is an auto-correlation detector that uses a local auto correlation function, f_{lac}. The function gives measures for the local changes in a window by shifting it in any direction on an image. In other words, the local auto-correlation function gives how similar the image function, I(x, y), is at point (x,y) to itself when shifted by (Δx, Δy). A point is defined as a *corner* at the point where the f_{lac} has distinct peaks. The f_{lac} is defined in [11], [30] as

$$f_{lac}(x, y) = \sum_{(u,v)\in W} w(u, v)\ [\ I(u, v) - I(u + \Delta x, v + \Delta x)\ ]^2 \tag{4.12}$$

where w(x,y) is a window centered at the point (x,y) and $I(u, v)$ is the image function. The window function can be either constant or Gaussian. Constant window takes the value 1 if all points are in the window and 0 otherwise. Gaussian window is a better choice to overcome the noise problem [11] and is given in (4.13).

$$w(u, v) = e^{-\frac{(u-x)^2+(v-y)^2}{2\sigma^2}} \tag{4.13}$$

The shifted image function, i.e. $I(u + \Delta x, v + \Delta x)$, is approximated by its first-order Taylor series expansion as

$$\begin{aligned} I(u + \Delta x, v + \Delta x) &\approx I(u, v) + I_x(u, v)\Delta x + I_y(u, v)\Delta y \\ &= I(u, v) + [I_x(u, v) I_y(u, v)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned} \tag{4.14}$$

where I_x and I_y are the partial derivatives of I(x,y) with respect to x and y respectively [11]. The partial derivatives are given as

$$\begin{aligned} I_x &= \frac{\partial I(x,y)}{\partial x} \approx \frac{I(x+1,y)-I(x-1,y))}{2} \\ I_y &= \frac{\partial I(x,y)}{\partial y} \approx \frac{I(x,y+1)-I(x,y-1))}{2} \end{aligned} \tag{4.15}$$

$\sum_w$ is used instead of $\sum_{(u,v)\in W}$ w for simplicity. The auto-correlation function is approximated by a quadratic function, i.e. $Q(x, y)$ for small shifts and is given in (4.16).

$$\begin{aligned} f_{lac}(x, y) &= \textstyle\sum_W [I(u, v) - I(u + \Delta x, v + \Delta x)]^2 \\ &\approx \textstyle\sum_W (\begin{bmatrix} I_x(u, v) & I_y(u, v) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix})^2 = Q(x, y) \end{aligned} \tag{4.16}$$

$$Q(x, y) = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \sum_W I_x^2 & \sum_W I_x I_y \\ \sum_W I_x I_y & \sum_W I_y^2 \end{bmatrix} \tag{4.17}$$

Let λ_1 and λ_2 be the eigenvalues of $Q(x, y)$. The eigenvalues form a rotationally invariant description of $Q(x, y)$ [11] and [30]. There are three cases to be considered

1. If both λ_1 and λ_2 are low, the local autocorrelation function $f_{lac}(x, y)$ is flat. That means $f_{lac}(x, y)$ has small changes in any direction and the windowed image region has approximately constant intensity.

2. If one eigenvalue is high and the other is low, the local autocorrelation function $f_{lac}(x, y)$ is ridge shaped. That means $f_{lac}(x, y)$ has small changes along the ridge direction and significant changes in the orthogonal direction. This indicates an edge.

3. If both eigenvalues are high, the local auto-correlation function $f_{lac}(x, y)$ has a distinct peak. $f_{lac}(x, y)$ has significant changes in any direction. This indicates a corner.

Corner response function, R, is used to measure the quality of a corner. Isolated corner points are selected according to the magnitude of the corner response function. R is characterized by eigenvalues of $Q(x, y)$ as

$$R = (\lambda_1 + \lambda_2) - k(\lambda_1 \lambda_2)^2 \tag{4.18}$$

where k is an adjustable positive constant. k is selected empirically between 0.04 and 0.06. R is positive and large at corner regions, small at flat regions, negative and large at edge regions. Local maxima of R give the corner points. An adjustable threshold for R can be used to eliminate some less quality corner points.

4.3.2 Parameter Settings for New-Born Targets

A measurement is a two dimensional vector denoted by $z = [z_x, z_y]^T$ which is an element of Z_k at time k. Unless the intensity is less than the predefined threshold value, it means that the measurements correspond to the pixel positions comes from a new-born target. The mean values of the parameters of initial state of a new-born target, i.e. d_0, l_0 and θ_0, are calculated by using the measurement z and is given in (4.19) and (4.20) except l_0. Since, l_0 is the velocity of the vehicle and is assumed to be same for all targets. Hence, l_0 for a new-born target is given as a constant predefined value. Covariance matrix of the state of a new-born target is taken to be same as the Q given in (4.22).

$$d_0 = \frac{f}{r} \tag{4.19}$$

$$\theta_0 = atan(\frac{z_y}{z_x}) \tag{4.20}$$

Consequently, any target is born with its mean and covariance as given in following equations.

$$m_n = \begin{bmatrix} d_0 \\ l_0 \\ \theta_0 \end{bmatrix} \tag{4.21}$$

$$P_n = \begin{bmatrix} w_l^2 & 0 & 0 \\ 0 & w_d^2 & 0 \\ 0 & 0 & w_\theta^2 \end{bmatrix} \tag{4.22}$$

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 Introduction

In this chapter the test results of the proposed algorithm will be given. The algorithm is tested with both synthetic and real data. The synthetic data is used to validate the algorithm and the real data is used to observe the performance of the proposed method for real measurements. Synthetic data contains generated noisy measurements and clutter. Generation process of the measurements is based on the motion and measurement models. "True" state sequence is generated by using the motion model. Related measurements are generated according to the measurement model by considering the probability of detection, p_D. Consequently, some measurements are discarded according to p_D. Clutter is generated by assuming uniform distribution in the surveillance area. Predefined number of clutter points are generated and added to the measurement set at each time step as well.

Firstly, single target tracking is done to validate of the proposed method. Since the measurement equations are nonlinear and the noises are additive Gaussian white noises, extended Kalman filter is a good solution for single target tracking. However, a different solution is needed for multiple target tracking because the EKF filter can track only a single target. Therefore, EK-GMPHDF is chosen because of its advantages that are explained in Chapter 3.2. Before using the EK-GMPHDF for multiple target tracking, the filter is tested for tracking a single synthetic target with different clutter densities.

For multiple target tracking using EK-GMPHDF, birth process is critical and should be defined appropriately. In this thesis, it is assumed that a birth process occurs at a predefined region around the vanishing point and the number of births is modeled as a Poisson process. In our specific problem, targets cannot spawn. Therefore, if a new target appears it is a new-born target. Obviously a target may die at any time. A target dies when its weight is so small relative to other targets. If the measurements are not close to its predicted position, weight of a target becomes small. The birth and death processes are the same both for synthetic and real data experiments.

After the all experiments are done for synthetic data, the real data is used to investigate the performance of the proposed EK-GMPHDF. Real data is the output of the HCD. The first step of the proposed method is to extract the features from the movie. The movie is taken from [28].

The video used in this study is 60 seconds long. The frame rate is 29 frames per second and each frame has 1280x720 pixels. In the selected part, velocity of the train is almost constant and the railway is linear. A frame captured from the video is shown in Figure (5.1).

Figure 5.1: A frame captured form the movie

Each frame of the movie (5.1) is an input of the HCD. The HCD finds the corners as interest points or features with their x and y coordinate values (pixels). A sample output of the HCD is given in Figure (5.2) when its input is the image given in Figure (5.1). The positions of the extracted features are the outputs of the HCD and named as the "*real data*".

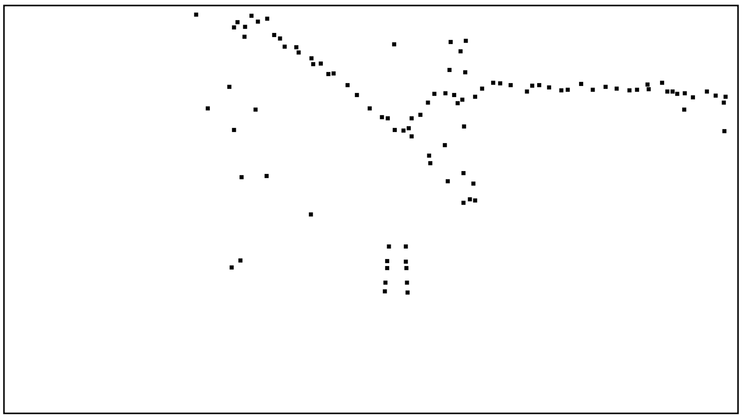

Figure 5.2: Output of the HCD for the real data frame. Black dots show the corners

Figure 5.3 shows the union of all corners detected by the HCD for 100 frames.

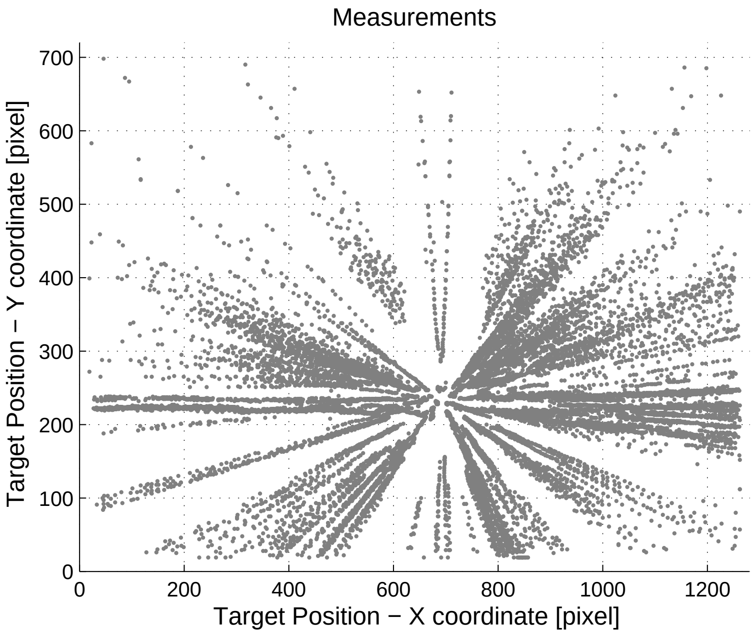

Figure 5.3: Multiple targets measurements taken from real video of time length 100 frames. All extracted features on 100 frames are superimposed into one picture

For each of these 100 frames, vanishing point is computed manually. The vanishing point can be seen in Figure 5.3 according to perspective and the point is calculated by finding the crossing point of the lines in a frame. Here, the vanishing point is calculated manually using the railway lines. It is observed that the vanishing point is approximately same for each frame. As a result, the vanishing point is set to $(693, 231)$ pixel location. The center $(0, 0)$ is moved to the vanishing point.

5.2 Single Target Tracking

We have done several experiments for single target tracking using the proposed model and the EKF to investigate the performance of EKF and validate the model. Synthetic measurement data is used in the experiments before real data. We flesh the experiments out in the following sections.

5.2.1 Synthetic Measurement Generation for a Single Target

First step of the single target tracking is the generation of the synthetic measurement data based on the model. In our case, a target can be born in a predefined specific region. The position of the new-born target gives initial r_0 and θ_0 values. The r_0 is selected from uniform distribution on the interval $[55, 65]$. As mentioned before, the real data is the output of the HCD for the input video [28]. θ_0 value is selected from a uniform distribution between $-\pi$ and π. Then initial measurement $z_0 = [x_0, y_0]^T$ is calculated as given in (5.1) and (5.2).

$$x_0 = r_0 \, cos(\theta_0) \tag{5.1}$$

$$y_0 = r_0 \, sin(\theta_0) \tag{5.2}$$

After calculation of initial measurement vector, initial state $x_0 = [d_0, l_0, \theta_0]^T$ is calculated. The third component of the state vector, θ_0, is already known. d_0 is selected according to the value of r_0 as

$$d_0 = \frac{f}{r_0} \tag{5.3}$$

and l_0 is set to a fixed value of 0.8. The constant f (f = 1000 in our experiments) and l_0 is determined from the real data experiments. A target dies if generated measurement is outside of the surveillance area. The surveillance area is taken as $([-640, 640]x[-360, 360])$ that is similar with the real data.

After initiation a target is propagated according to (4.3), (4.4) and (4.5). After all the states are calculated, measurements are generated according to (4.6) and (4.7) for each time step. Hence the synthetic measurement set for a single target, i.e. $Z = [z_1, ..., z_k]^T$, is generated.

5.2.2 Single Target Tracking with EKF

The potential of EKF is investigated as a tracker for our problem. For this purpose, the synthetic measurement obtained as explained in Section 5.2.1 is used. In the simulations, the covariance matrix of the process noise, i.e. Q, and the covariance matrix of measurement noise, i.e. R are selected as given in (5.4) and (5.5).

$$Q = \begin{bmatrix} w_l^2 & 0 & 0 \\ 0 & w_d^2 & 0 \\ 0 & 0 & w_\theta^2 \end{bmatrix} \tag{5.4}$$

$$R = \begin{bmatrix} r_x^2 & 0 \\ 0 & r_y^2 \end{bmatrix} \tag{5.5}$$

where $w_l = 0.04$, $w_d = 4$, $w_\theta = 0.0001$ and $r = 4$. In figures 5.4, 5.5 and 5.6, the true trajectory of the target (a), measurements (b) and the tracked trajectory (c) are shown respectively. According to these results, it is decided that the EKF filter is capable of tracking the non-linear movement of the background objects.

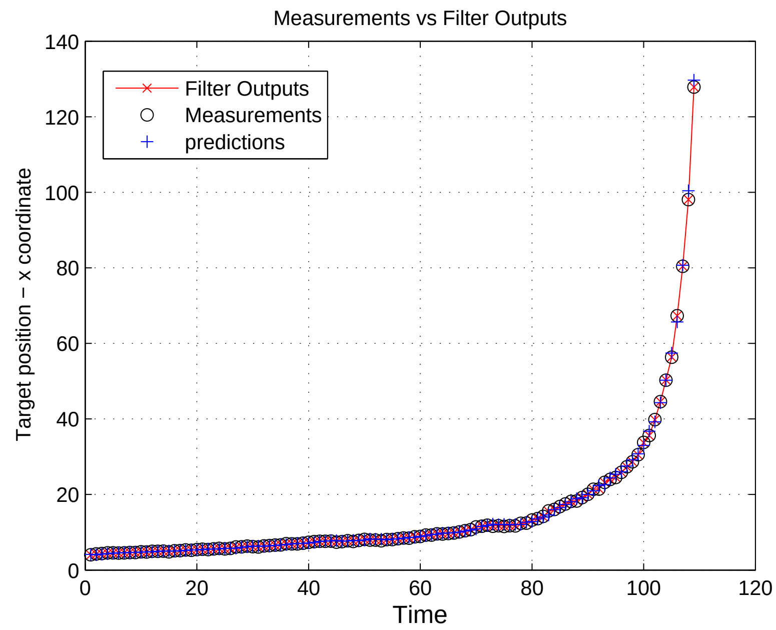

Figure 5.4: Tracking results of EKF in x-t coordinate with synthetic data

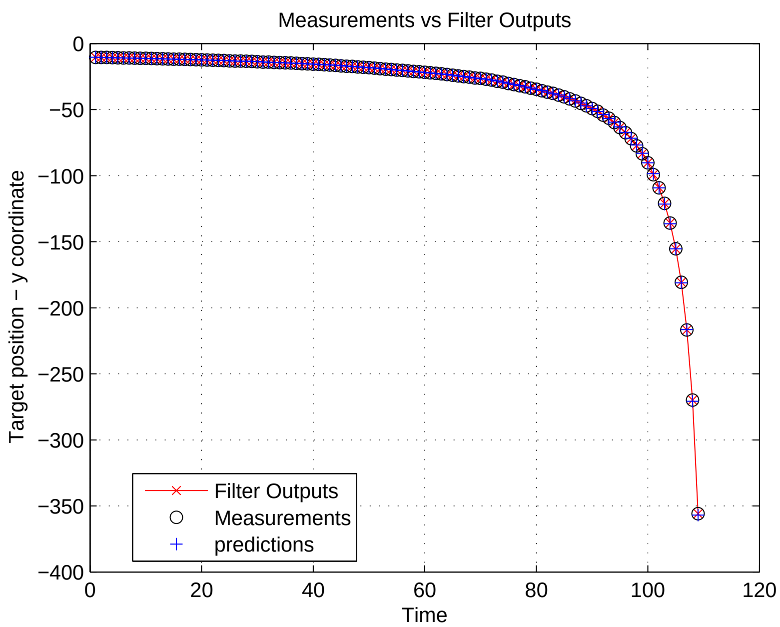

Figure 5.5: Tracking results of EKF in y-t coordinate with synthetic data

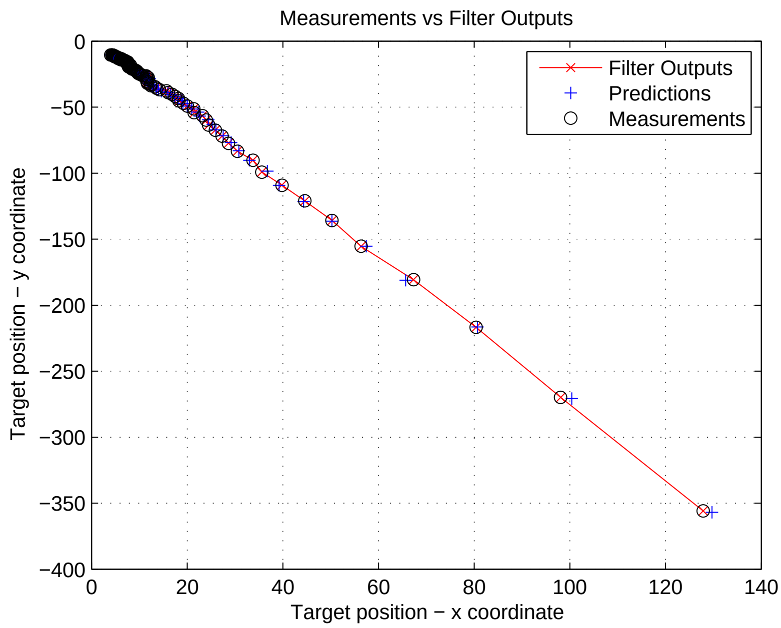

Figure 5.6: Tracking results of EKF in x-y coordinate with synthetic data

After the validation of the EKF as the tracker, we used the EKF with the real data. The model is the same as the synthetic data simulation. The only difference is the measurements that are taken manually from the input video. The EKF can track the target successfully according to results that are given in (5.7), (5.8) and (5.9).

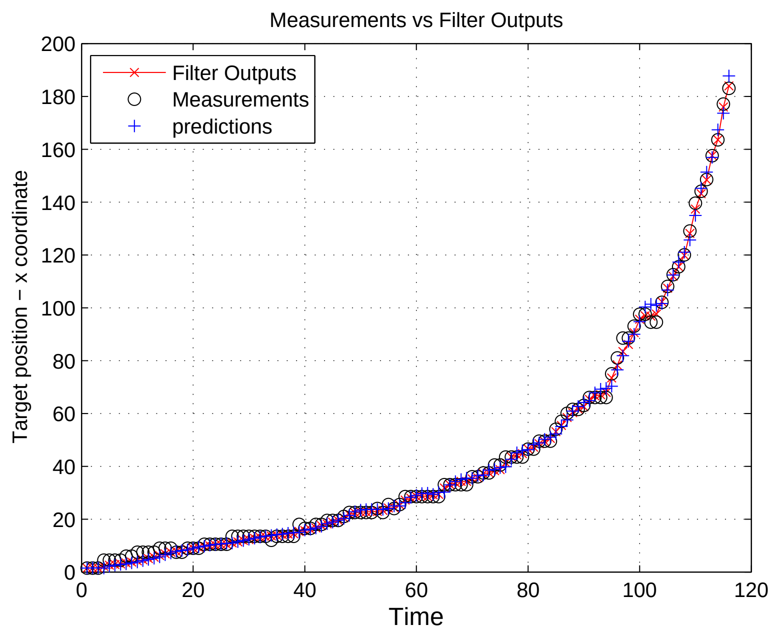

Figure 5.7: Tracking results of EKF in x direction with real data

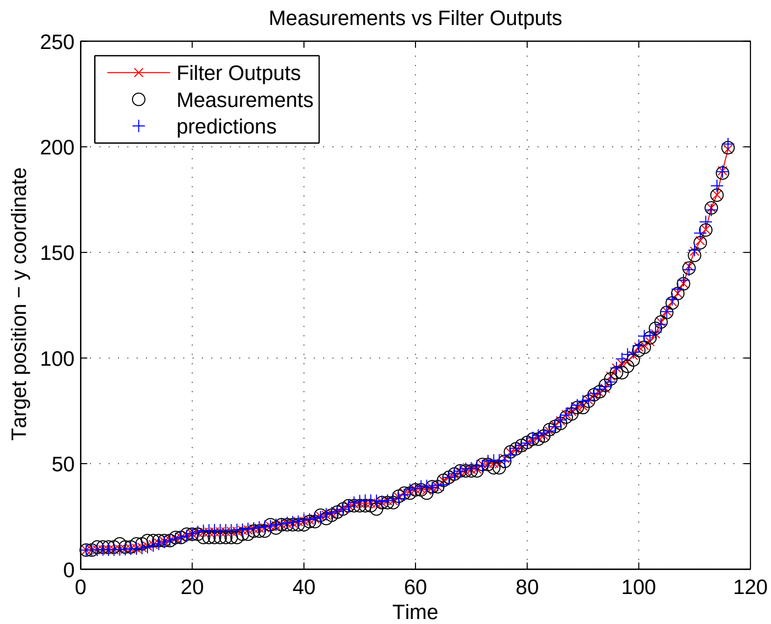

Figure 5.8: Tracking results of EKF in y direction with real data

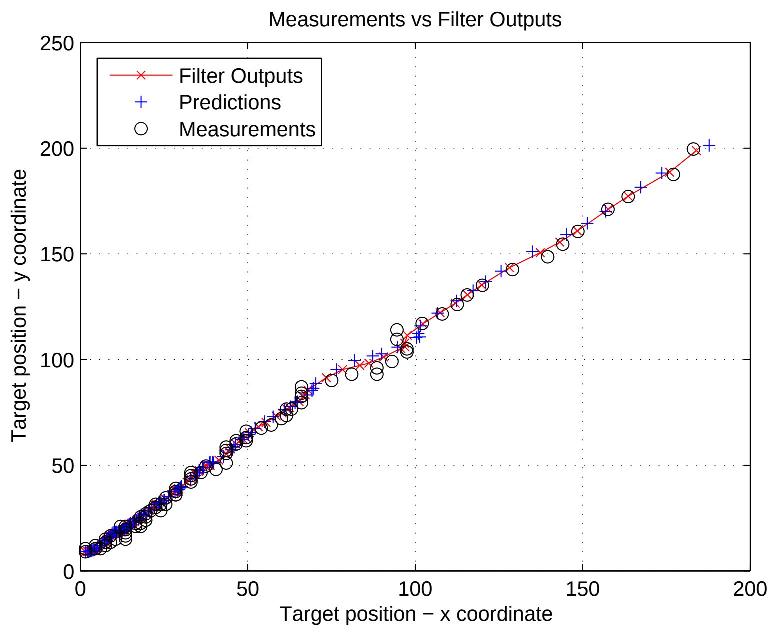

Figure 5.9: Tracking results of EKF in xy direction with real data

Figures show that the EKF using the given model that gives satisfactory performance and also validates the model. Although the tracking performance of EKF is satisfactory for tracking a single target, for multiple targets case some additional tools are required. PHD filter is selected as the tracker of the multiple targets in this thesis because of the absence of data association. When the Gaussian assumptions for noises hold like in our case, GMPHDF can be used. However, GMPHDF uses linear Kalman filter equations that are not suitable with the models. Therefore, EKF equations are used instead of the Kalman filter equations in GMPHDF and the filter became EK-GMPHDF can track any number of targets. Tracking of a single target with EK-GMPHDF is given in the following section.

5.2.3 Single Target Tracking with EK-GMPHDF

Single target tracking is performed to investigate the performance of the algorithm before multiple target tracking. Similar to the single target EKF tracking given in the previous section, the real data is used with detection probability $p_D = 1$ and number of clutter per unit volume nClutter = 10. In these experiments, another target measurement is added that does not fit the model to see the performance of the filter. In Figures (5.10) and (5.11), filter results are given as $x - time$, $y - time$ and $x - y$. Figure (5.12), extracted number of targets is given. As it can be seen in the following figures, the target can be tracked with EK-GMPHDF successfully. In Section 5.3, tracking different number of targets with EK-GMPHDF is given in detail.

Single target tracking is performed to see the algorithm works well before multiple target tracking. Similar with the single target EKF tracking given previous section, the real measurements were used with detection probability $p_D = 1$ and number of clutter per unit volume nClutter = 10. In these experiments, there is added another target measurement that does not fit the model to see the performance of the filter. In Figures (5.10) and (5.11), filter results are given for both x and y coordinates separately and together respectively. Figure (5.12), it is seen that how many target could be extracted. As it can be seen in the following figures, the target can be tracked with EK-GMPHDF successfully. In section 5.3, tracking different number of targets with EK-GMPHDF will be given in detail. Single target tracking is performed to see the algorithm works well before multiple target tracking. Similar with the single target EKF tracking given previous section, the real measurements were used with detection probability $(p_D) =$ 1 and number of clutter per unit volume (nClutter) = 10. In these experiments, there is added another target measurement that does not fit the model to see the performance of the filter. In Figures (5.10) and (5.11), filter results are given for both x and y coordinates separately and together respectively. Figure (5.12), it is seen that how many target could be extracted. As it can be seen in the following figures, the target can be tracked with EK-GMPHDF successfully. In section 5.3, tracking different number of targets with EK-GMPHDF will be given in detail.

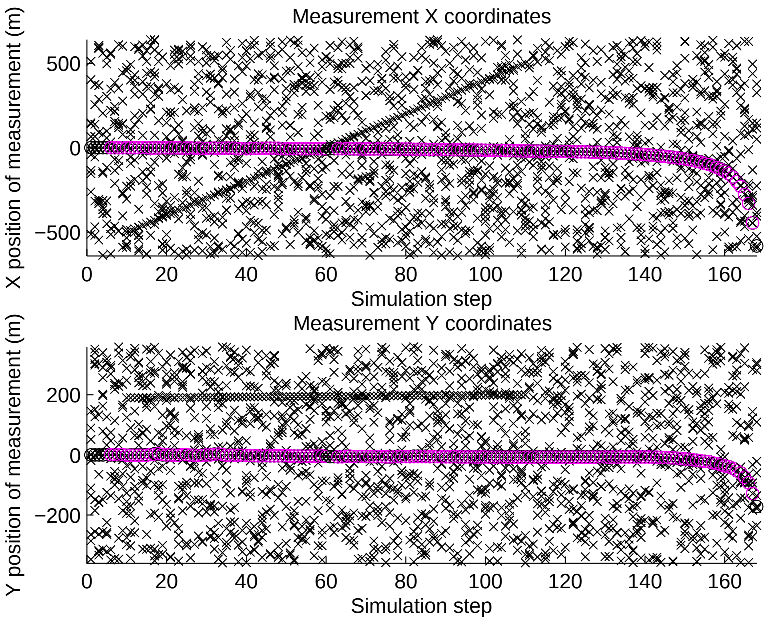

Figure 5.10: Tracking results of EK-GMPHDF in x and y coordinates with real data

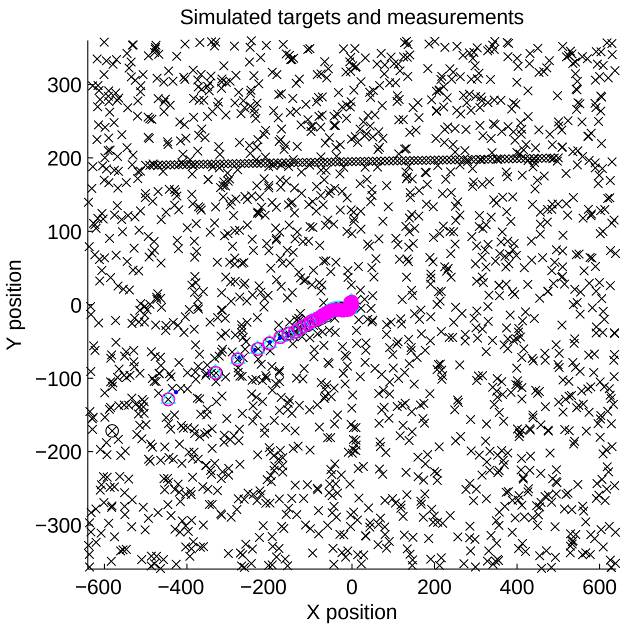

Figure 5.11: Tracking results of EK-GMPHDF in xy coordinate with real data

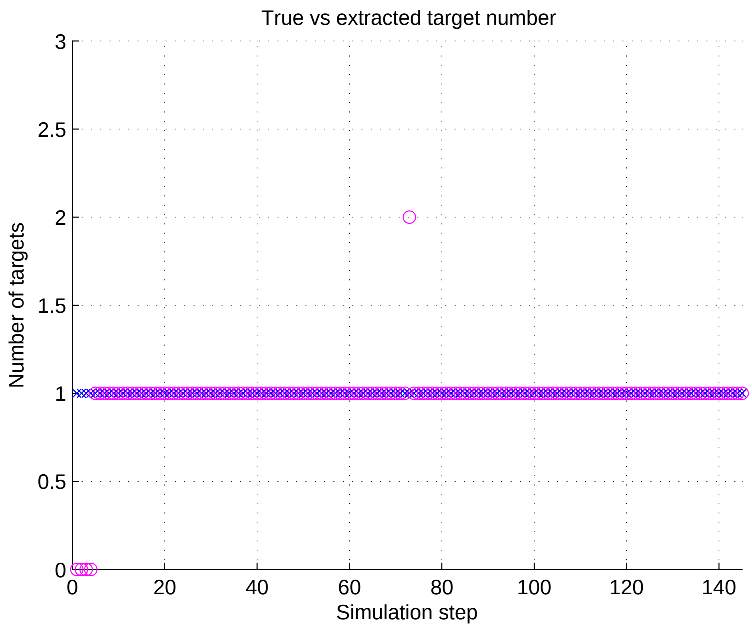

Figure 5.12: Real target number vs Extracted target number(EK-GMPHDF - 1 Target)

5.3 Multiple Target Tracking with EK-GMPHDF

Multiple target tracking experiment are also done using both the real and the synthetic data. The synthetic data experiments are done for different values of probability of detection p_D and number of clutter per unit volume *nClutter*. Output of HCD has natural false alarms with a unknown p_D. In the real data experiments, different values of probability of detection p_D are used in the EK-GMPHDF algorithm. In EK-GMPHDF algorithm, data association between measurements and the targets is not done. In addition, number of targets, birth times of the targets, death times of the targets, and birth places of targets are not known. In order to detect and then track a target, a well defined birth process must be used. The birth process is explained in the following section.

5.3.1 Birth Process

According to the inspection of the video, births occur at positions that are approximately 60 pixels far from the center of the image. Intensity of a new-born target is modeled as an exponential distribution which is zero for $r < r_0$ and given in (5.6) and

$$n(r) = \lambda^r e^{r-r_0} u[r - r_0] \tag{5.6}$$

$$r = \sqrt{z_x^2 + z_y^2} \tag{5.7}$$

In (5.7), $z = [z_x, z_y]^T$ is an element of Z_k at time k and is considered as a measurement that belongs to a new-born target if the intensity of this pixel is greater than the predefined threshold value. Then initial mean of the new-born target is calculated and the target joins the survival targets. States of the initial mean, i.e. d_0, l_0 and θ_0, are calculated by using the measurement z and given in (5.8) and (4.20) except l_0. Because, l_0 is the velocity of the vehicle and it is assumed constant for all targets. Hence, l_0 for the new-born target is given as a constant predefined value, i.e. $l_0 = 0.8$. Covariance of the new-born target is taken same as the Q given in (5.4).

$$d_0 = \frac{f}{r} \tag{5.8}$$

$$\theta = atan(\frac{z_y}{z_x}) \tag{5.9}$$

Consequently, any target births with its mean and covariance as given in following equations.

$$m_n = \begin{bmatrix} d_0 \\ l_0 \\ \theta_0 \end{bmatrix} \tag{5.10}$$

$$P_n = \begin{bmatrix} w_l^2 & 0 & 0 \\ 0 & w_d^2 & 0 \\ 0 & 0 & w_\theta^2 \end{bmatrix} \tag{5.11}$$

5.3.2 Truncation Process

Merging and pruning processes are done as explained in Section 3.2.1.1. Furthermore, the merging and pruning parameters are chosen similar to the parameter given in [13] and [26]. Basically, there are three parameters, i.e. weight threshold (T_w), merge threshold (T_m) and extracted weight threshold ($T_e w$). At first, all the weights less than T_w are eliminated. T_w is chosen as 10^{-5}. After the elimination of the small Gaussians, the remaining ones are clustered and each cluster is merged to single Gaussian.

After that, maximum intensity is found and then he distances between the maximum intensity and the others are calculated. If any distance is less than the merge threshold, i.e. T_m, the targets are combined. T_m is chosen as 2. Then, scaling of intensities is done and finally the intensities that are greater than $T_e w$ are accepted as real extracted targets. Here, $T_e w$ is chosen as 0.5.

5.3.3 Synthetic Measurement Experiments

The detection probability p_D, clutter intensity and number of targets NoT are the parameters that effect the performance of the proposed method. In order to investigate the performance of the proposed method, a performance measure is used. The measure is the average of the differences between the number of true tracks and number of the extracted tracks. Furthermore a single measure that shows the performance for correct number of targets is defined as follows.

$$PM = 1 - EoF \tag{5.12}$$

where

$$EoF = \frac{1}{\sum_{k=1}^{TH}} \sum_{k=1}^{TH} \mid NoT_k^t - NoT_k^e \mid \tag{5.13}$$

TH denotes the time horizon; NoT_k^t is the true number of targets at time k and NoT_k^e is the output of the system as number of targets.

The performance measure is calculated as the average of results of 15 runs. The measurement data corresponding to a target is generated as given in Section 5.2.1. All the measurements corresponding to targets are generated explicitly and then combined according to birth times of the targets. Birth times of the targets are determined according to Poisson distribution with mean $\lambda = 15$.

In the first set of experiments to observe the effects of number of targets, detection probability p_D is set to 1 and the number of clutter per time step NoC is set to 10. The typical results obtained from a single run when $NoT = 10$ are given in Figures 5.13, 5.14, 5.15 and 5.16. In this experiment, the survival probability p_S is set to 0.99, number of clutters per time step NoC is set to 10. In Figure 5.13, generated synthetic measurement data for 10 targets is shown. The extracted and predicted targets are shown in the Figure 5.14. The blue dots show the predicted target positions and the magenta circles show the filter output and the black crosses show the measurements. In Figure 5.15, filtered target positions are shown in x and y axes separately with respect to time horizon. The extracted number of targets and the real number of targets are shown in Figure 5.16.

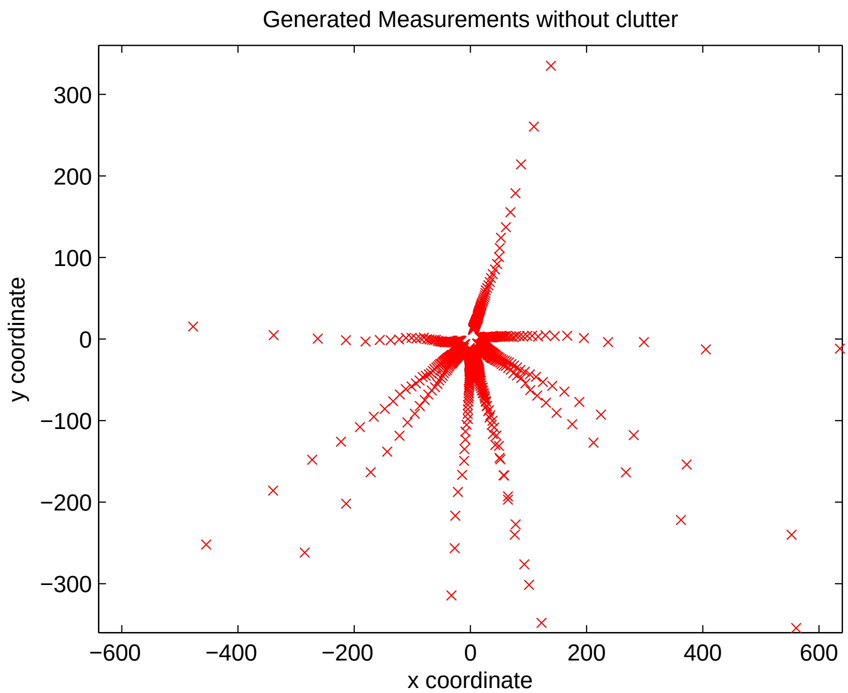

Figure 5.13: Generated measurement data for 10 targets

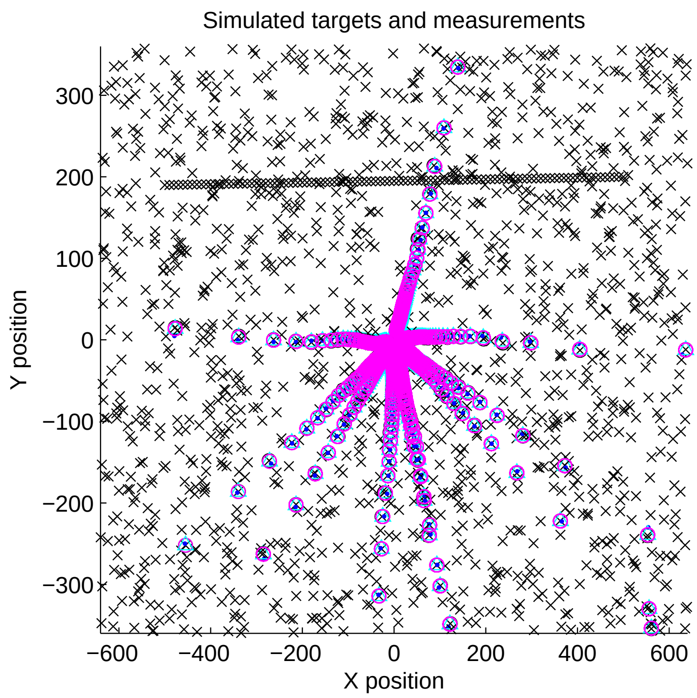

Figure 5.14: Tracking result of EK-GMPHDF with synthetic data

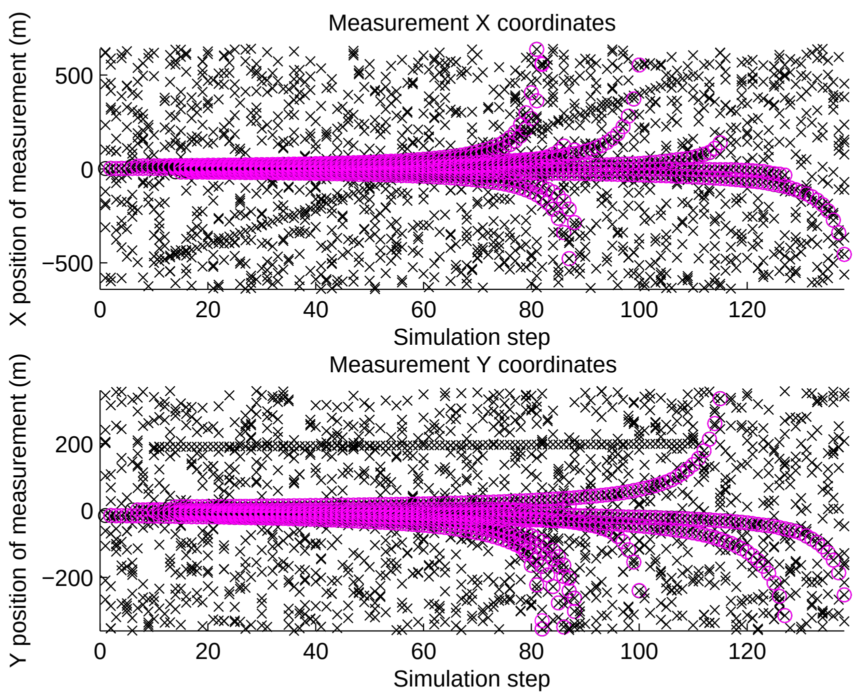

Figure 5.15: Tracking results of EK-GMPHDF in x and y coordinates with synthetic data

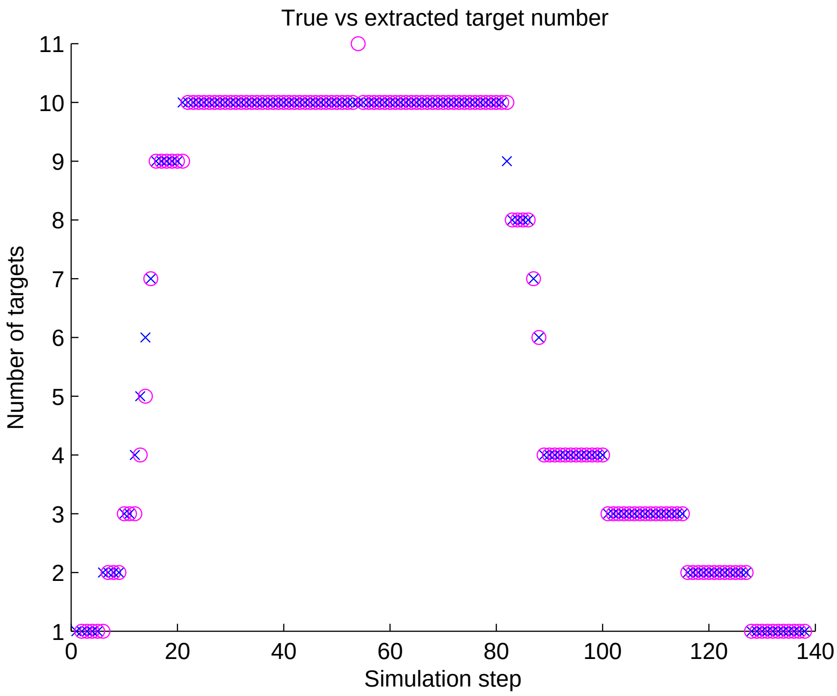

Figure 5.16: Extracted number targets vs real number of targets: o represents the extracted and x represents the number of targets

Table 5.1 gives this performance measure (PM) for different number of targets (NoT). As it can be seen in Table (5.1), performance of the proposed method decreases when the number of targets increases.

NoT	10	20	30	40	50
EoF	2.36%	4.95%	7.45%	8.86%	10.13%
PM	97.64%	95.05%	92.55%	91.14%	89.87%

Table5.1: Performance of the EK-GMPHDF for different number of targets (NoT)

In the second set of experiments to observe the effect number of clutters per time step (NoC), the detection probability and number of targets are kept constant as $p_D = 1$ and $NoT = 10$. The performance of the filter slightly decreases when the NoC increases and is given in Table 5.2.

NoC	10	20	30	40	50
EoF	2.36%	2.83%	3.76%	3.97%	4.13%
PM	97.64%	97.17%	96.24%	96.03%	95.87%

Table5.2: Performance of the EK-GMPHDF for different clutter intensity (NoC)

In the third set of experiments to observe the effect of detection probability p_D to the performance of the filter, number of clutter per time step over the surveillance area is kept constant as $NoC = 10$. The number of targets is also kept constant as $NoT = 10$. Performance of the EK-GMPHDF for different detection probability (p_D) is given Table 5.3. If the detection probability decreases, then the performance of the filter decreases according to Table 5.3.

p_D	1.00	0.98	0.95	0.9
EoF	2.36%	5.77%	7.80%	13.02%
PM	97.64%	94.23%	92.20%	86.98%

Table5.3: Performance of the EK-GMPHDF for different probability of detection (p_D)

To sum up the calculated errors and the performances of the proposed method for artificial data is given in the following figures. In the Figure 5.17, output errors of the filter are shown for different values of the parameters. The performance of the EK-GMPHDF is given in Figure 5.18.

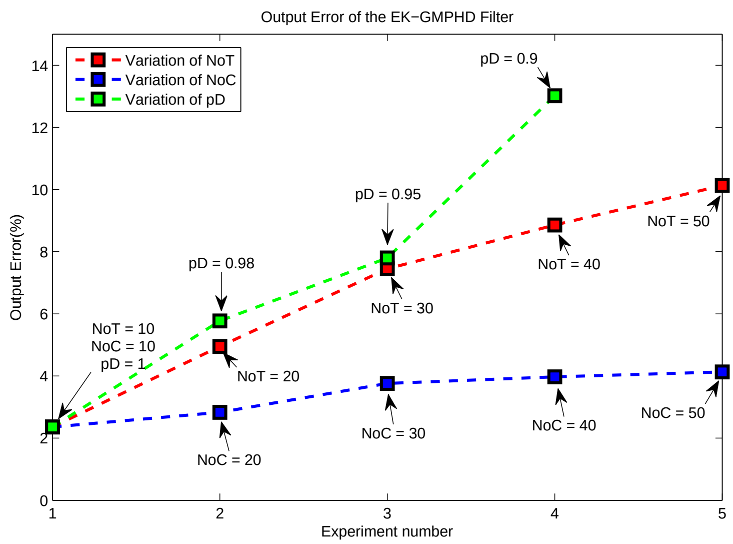

Figure 5.17: Output Errors of the EK-GMPHDF (x axis denotes the experiment number corresponding to the tables 5.2, 5.3 and 5.1

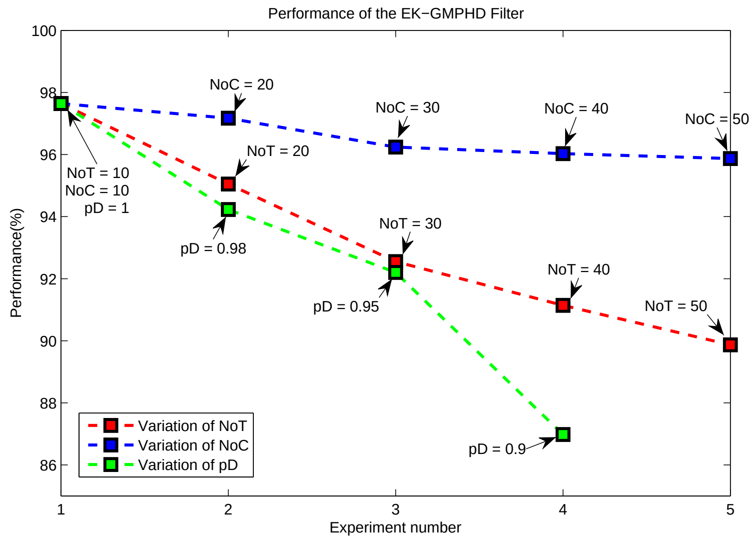

Figure 5.18: The performance of the EK-GMPHDF (x axis denotes the experiment number corresponding to the tables 5.2, 5.3 and 5.1)

5.3.4 Real Measurement Experiments

The video given in [28] is and used as the input of the proposed method to track the background objects. The video is processed image by image by the HCD that formed a measurement set, i.e. z_k, at time k. It is not possible to know which measurements are real which ones are clutters. Furthermore, number of targets, exact birth positions, birth times, death times, detection, and survival probabilities of the targets are other unknown parameters. Although the detection and survival probabilities are unknown, we have used they are predefined constant values in these experiments. The probabilities are set empirically according to the video. Birth position and birth time of a new-born target are determined according to *birth process* as explained in Section 5.3.1. Additionally, a target dies when it is in the outside of the image plane. Several experiments are done for different values of the parameters that are explained above. As a result, the parameters are approximately found to be suitable for the input and the proposed method.

In this experiment, the same parameters are used that are given in Section 5.3. In Figure 5.19, all the predicted and the extracted targets are shown. The blue dots are the predicted positions of the targets and the magenta circles are the extracted positions of the targets. Measurements are shown as the black crosses and the cyan triangles show the last extracted state. It helps to follow the movements of the tracks.

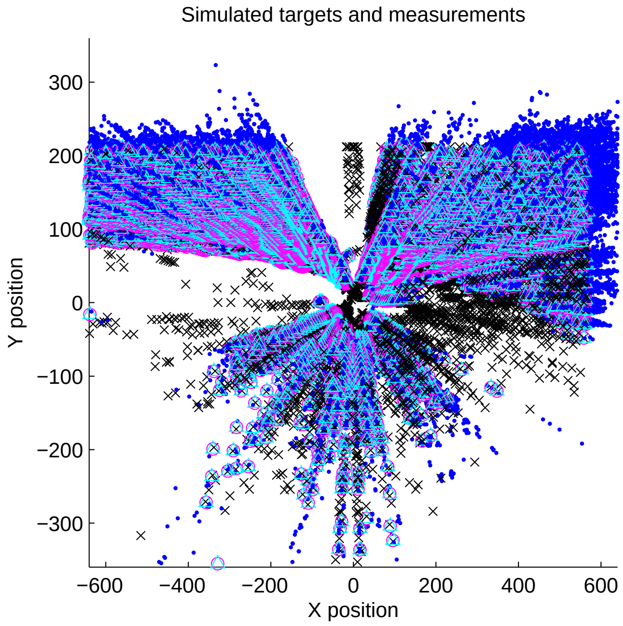

Figure 5.19: Tracking results of EK-GMPHDF in x and y coordinates with real data taken from the video camera

The number of extracted targets is given in Figure 5.20.

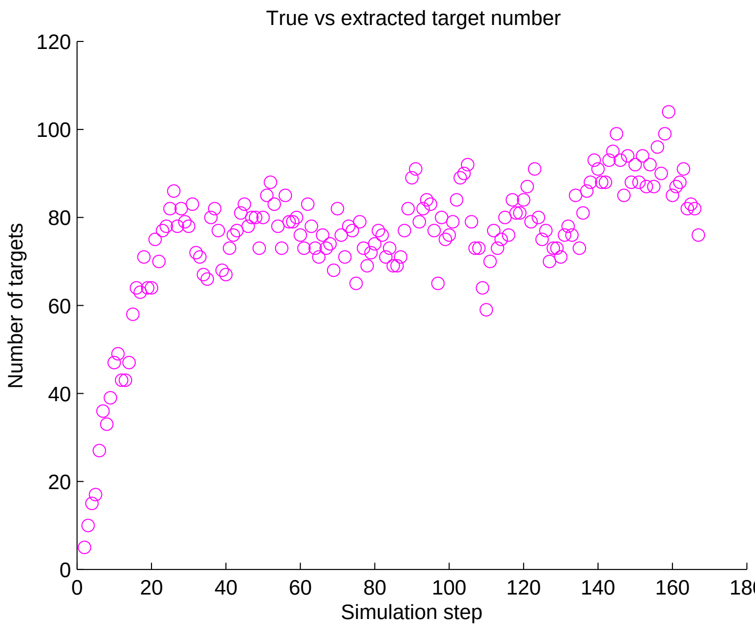

Figure 5.20: Number of the extracted background objects by using EK-GMPHDF and output of the video camera

In order to investigate that the proposed method tracks only the background objects which behave according to given models, the following experiment is done. Measurements for a target which are stationary at fixed points are added to HCD output and the union of these measurements is used as the input of the proposed EK-GMPHDF. Since the measurements of the target are incompatible with the motion and the observation model, the EK-GMPHDF does not detect and track the object. Result of the experiment is shown in Figure 5.21 where the black crosses in the red circle are the added measurements belong to the irrelevant target.

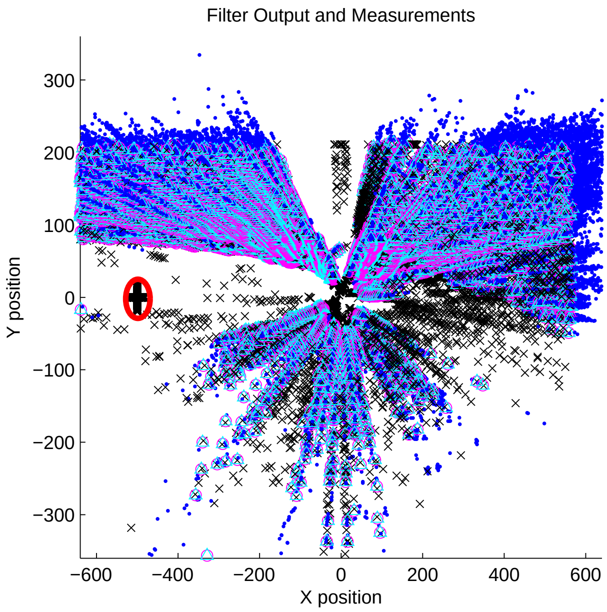

Figure 5.21: Tracking results of EK-GMPHDF in x and y coordinates with added irrelevant data to the real data

The measurements associated to the related targets are found to investigate convenience of the true measurements and the extracted targets. The algorithm takes a matrix as an input and matches the targets and measurements according to a given criteria. In our case, the criteria is a function of (r, θ). r is the radial distance from the vanishing point and the θ is the angle between the position in x-y coordinate and x axis. The criteria is named as the cost function, i.e. f_{auc}, and given in (5.14).

$$f_{auc} = \frac{1}{c_{norm}} (m_r \, (r_t - r_m)^2 + M_\theta \, (\theta_t - \theta_m)^2) \tag{5.14}$$

The m_r and M_θ are the constant multipliers. Since θ is more reliable parameter than r in our problem, M_θ is selected as 100 and m_r is selected as 1. The normalization parameter, i.e. $c_n orm$, is selected as 1000. The r_t, r_m and θ_m are calculated as given in followings. θ_t is known since it is a state parameter.

$$r_t = \sqrt{(\frac{f\ cos(\ \theta_m\ )}{d})^2 + (\frac{f\ sin(\ \theta_m\ )}{d})^2} \tag{5.15}$$

$$r_m = \sqrt{(x_m^2 + y_m^2} \tag{5.16}$$

$$\theta_m = atan(\frac{y_m}{x_m}) \tag{5.17}$$

The states of the targets and the measurements are in the form of $[d\ l\ \theta]^T$ and $[x_m\ y_m]^T$ respectively. After the auction algorithm is run with this set up, the measurements are associated to the related targets and are given in Figure 5.22. The black crosses are the measurements and the magenta circles are the positions of the extracted tracks.

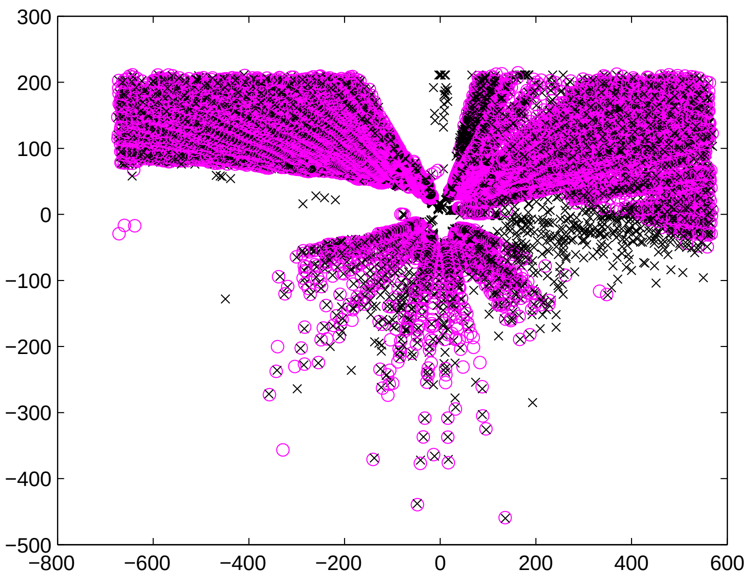

Figure 5.22: Tracking results of EK-GMPHDF in x and y coordinates after auction algorithm is run

CHAPTER 6

SUMMARY AND CONCLUSIONS

In this thesis study, we have proposed a method that utilizes the feature tracking approach to track the background of a video. Firstly, the features are chosen as the corner points of the background objects and are extracted from each frame of the video by using well known corner detection technique, namely the Harris corner detector. Then a linear process model and a non-linear measurement model are utilized to track these features. The performance is tested using both the real and the synthetic data. The real data in these tests are manually extracted from the test video. The results show that models are suitable for the problem under consideration. To handle the non-linear measurement model EKF type approach is utilized in the GM-PHD framework. Then the proposed method, namely EK-GMPHDF, is tested with synthetic data generated for a time varying number of targets. Performance of EK-GMPHDF is investigated by altering the number of targets, detection probability and clutter density. It is observed in these tests that the performance of the filter increases with higher detection probability and lower number of targets. Effect of the clutter density is relatively small. The parameter configuration of the proposed method is determined with the test results obtained from the synthetic data. As a result, all the features of the background which fit the models are successfully tracked by the proposed method. Lastly, it is observed from the test results that the proposed method can be used for background tracking of a video instead of classical background tracking methods under some assumptions.

6.1 Future Work

The performance of the proposed background tracking algorithm depends on the motion and the observation models. Better models can be designed to improve the performance of the proposed method depending on the configuration of the camera on the vehicle and the a more accurate knowledge about the motion dynamics. Moreover, a better feature extraction method can be utilized, again depending on the measurements hardware. The proposed method contains many adjustable parameters that influence the performance. Different combinations of these parameters can be studied to increase the overall performance. In addition, the proposed method can be improved to be used in the case of maneuvering motion of the vehicle. Finally, a post processing method can be utilized which may provide improvement in the performance of the tracker.

REFERENCES

[1] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. In *PROCEEDINGS OF THE IEEE*, pages 1151–1163, 2002.

[2] E. Nicholls and M. Strens. Robust background tracking for infrared image sequences. In *Target Tracking: Algorithms and Applications, 2006. The IEE Seminar on (Ref. No. 2006/11359)*, pages 123–128, 2006.

[3] V. Ablavsky. Background models for tracking objects in water. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–125–8 vol.2, 2003.

[4] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and Practice of Background Maintenance. *Seventh International Conference on Computer Vision*, 1:255+, 1999.

[5] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4, 2004.

[6] Simon Baker, Daniel Scharstein, J.P. Lewis, Stefan Roth, MichaelJ. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[7] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision (IJCV)*, 94(3):335–360, 2011.

[8] H. Aanæs, A.L. Dahl, and K. Steenstrup Pedersen. Interesting interest points – a survey of interest point performance on a unique data set. *International Journal of Computer Vision*, pages 1–18, 2011.

[9] *Robust feature-based object tracking*, volume 6568, 2007.

[10] *Real time object tracking based on dynamic feature grouping with background subtraction*, 2008.

[11] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[12] R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

[13] Ba-Ngu Vo and Wing-Kin Ma. The gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54(11):4091–4104, 2006.

[14] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

[15] Venkatarama Krishnan. *Nonlinear filtering and smoothing : an introduction to martingales, stochastic integrals, and estimation*. Wiley, New York, 1984. A Wiley-Interscience publication.

[16] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[17] Kalman, Rudolph, and Emil. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[18] Andrew H. Jazwinski. *Stochastic processes and filtering theory*. Number 64 in Mathematics in science and engineering. Acad. Press, New York, NY [u.a.], 1970.

[19] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[20] N.J. Gordon, D.J. Salmond, and A. F M Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

[21] Y. Ho and R. Lee. A bayesian approach to problems in stochastic estimation and control. *Automatic Control, IEEE Transactions on*, 9(4):333–339, oct 1964.

[22] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking : Principles and techniques*. Yaakov Bar-Shalom, Storrs (US), 1995. Index.

[23] S. S. Blackman. *Multiple-target tracking with radar applications*. Artech House, 1986.

[24] R. Mahler I. Goodman and H. Nyugen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Norwel MA, USA, 2010.

[25] David L. Hall and James Llinas. *Handbook of Multisensor Data Fusion*. CRC Press, June 2001.

[26] Daniel Edward Clark. *Multiple Target Tracking with The Probability Hypothesis Density Filter*. PhD thesis, Heriot-Watt University, Edinburgh, Scotland, 2006.

[27] K. Granstrom, C. Lundquist, and O. Orguner. Extended target tracking using a gaussian-mixture phd filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 48(4):3268–3286, 2012.

[28] Daniel Rubino. Train driver's view: Halmstad to göteborg.

[29] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2):151–172, June 2000.

[30] Konstantinos G Derpanis. The harris corner detector. *York University*, 2004.