USE OF TERRAIN INFORMATION TO IMPROVE THE PERFORMANCE OF A TARGET TRACKER

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

BY

MUSTAFA CANAY

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

JULY 2009

Approval of the thesis:

USE OF TERRAIN INFORMATION TO IMPROVE THE PERFORMANCE OF A TARGET TRACKER

submitted by **MUSTAFA CANAY** 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. İsmet Erkmen ___________
Head of Department, **Electrical and Electronics Engineering**

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

Examining Committee Members:

Prof. Dr. Mete Severcan _____________________
Electrical and Electronics Engineering Dept., METU

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

Prof. Dr. Mustafa Kuzuoğlu _____________________
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Çağatay Candan _____________________
Electrical and Electronics Engineering Dept., METU

Elif Yavuztürk, M.Sc. _____________________
ASELSAN Inc.

Date: **10.07.2009**

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 : MUSTAFA CANAY

Signature :

ABSTRACT

USE OF TERRAIN INFORMATION TO IMPROVE THE PERFORMANCE OF A TARGET TRACKER

Canay, Mustafa

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

July 2009, 92 pages

Radar target tracking problem has been a popular topic for several decades. Recent works have shown that the performance of tracking algorithms increases as more prior information is used by the system; such as maximum velocity and maximum acceleration of the target, altitude of the target, or the elevation structure of the terrain. In this thesis we will focus on increasing the performance of tracking algorithms making use of benefit from the elevation model of the environment where the target tracker is searching. For a constant target altitude and a certain radar location, we generate a "visibility map" using the elevation model of the terrain and use this information to estimate the location and the time that the target will reappear.

The second aim of this work is to use the visibility map information for improving the performance of track initiation. For that purpose, a special map has been formed, that we call as the "track initiation probability map", which shows the target first time appearance density. This information has been used at the initialization part of the track initiation algorithm in order to increase the performance.

Keywords: Target tracker, visibility map, track initiation, track initiation probability map.

ÖZ

HEDEF İZLEYİCİSİNİN PERFORMANSINI ARTIRMADA ARAZİ BİLGİSİNİN KULLANIMI

Canay, Mustafa

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

Temmuz 2009, 92 sayfa

Radarla hedef takibi problemi yıllar boyunca çalışılan önemli bir konu olmuştur. Bu konuda son yıllarda yapılan çalışmalar göstermektedir ki, kullanılan izleme algoritmasının performansı izlenen hedefle ilgili sahip olunan ön bilgilerin kullanılmasıyla artmaktadır. Bu ön bilgiler hedefin sahip olabileceği maksimum hız, maksimum ivme, hedefin yerden yüksekliği ya da hedefin uçtuğu arazinin yükselti yapısı gibi birçok şey olabilmektedir. Bu tezde radarın bulunduğu yerdeki arazinin yükseklik modelini kullanarak izleme algoritmasının performansını arttırma konusunda çalışılmıştır. Bu çalışmada, yükseklik haritasını kullanarak sabit yükseklikte uçan bir hedef ve belirlenmiş bir radar konumu için "görünürlük haritası" çıkartıp hedefin görülmediği durumlarda bu bilgiyi hedefin tekrar görüneceği yeri ve zamanı kestirmekte kullanmak öngörülmüştür.

Bu çalışmadaki ikinci hedefimiz ise, iz başlatma performansını, çıkartılan görünürlük haritasını kullanarak arttırmaktır. Bunun için hedefin ilk kez görünme yoğunluğunu gösteren "takip başlatma ihtimali haritası" adını verdiğimiz harita oluşturulmuş ve bu bilgi iz başlatma algoritmasının ilklendirmesinde kullanılıp algoritma performansı yükseltilmeye çalışılmıştır.

Anahtar Kelimeler: Hedef izleyicisi, görünürlük haritası, takip başlatma, takip başlatma olasılık haritası.

To My Family and Future Love

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my supervisor Prof. Dr. Mübeccel Demirekler for her belief, encouragements, complete guidance, advice and criticism throughout this study.

I would like to thank Aselsan Inc. for facilities provided for the completion of this thesis.

I would like to express my thanks to my friends for their support and fellowship.

I would also like to thank Tübitak for its support on scientific and technological researches.

I would like to express my special appreciation to my family for their continuous support and encouragements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND SCOPE OF THE THESIS

Sometimes tracking of an airborne target becomes cumbersome due to the geographical structure of the terrain. In some cases target disappears from the line of sight of the radar which causes track losses. One of the aims of this study is to preserve track continuity even if some obstructions of the target trajectory occur due to terrain structure. Our approach to the problem is to use the elevation structure of the environment as prior information in tracking algorithms.

As far as we know there are no works on the problem defined above in literature. There are some studies about using the map data in tracking algorithms but those works are for only ground targets which are detected by Ground Moving Target Indicator (GMTI) radar [8, 9, 10, 11]. In these studies generally a road map is used to improve the performance of the tracker by constraining the state vector of the system. In literature there are some studies about constrained state estimation problem or constrained Kalman filter. Here we will give general information about these works which is partially related to our problem. These studies can be divided into three groups.

The first group of works is the projection-type of methods [1, 2, 3, 4]. In these methods the unconstrained state estimates are projected onto the constrained surface. Standard Kalman filtering is used for both the state and the covariance matrix computations. The constraints are used only at the step of projecting the estimated state onto the constrained region. The covariance matrices are not affected from the constraints. The constraints may be in the form of equality [1], or there may be some inequality constraints about the states of the dynamic system [2, 3, 4].

The second group is to use Moving Horizon Estimation (MHE) technique for linear state estimation with constraints [5]. MHE solves the problem by using dynamic programming technique with a fixed sized window. We have mentioned about it here for the completeness however this technique is not in the scope of this thesis.

The last group of works uses Unscented Kalman filter (UKF) in constrained state estimation [6, 7, 20, 21]. With UKF probability distribution functions are represented by a certain number of points, called sigma points, that each of them has a weight W^i such that $\sum_{i=0}^{p} W^i = 1$. ($p+1$ = number of sigma points). At each iteration, sigma points are checked whether they are outside the constrained region or not. The sigma points which are outside the constrained region are projected to the boundary [7]. Both the state vector and the covariance matrix are updated by the constraints.

The studies given in the literature does not exactly match to our problem since for an airborne target there is no road map. Here the information that is used is the elevation model of the terrain. For a target flying with a constant altitude, a special map called “visibility map” which consists of three regions; “visible region”, “invisible region” and “impassable region”; can be generated. At visible region the target and the radar are in line of sight (los) while at invisible region they are not. The impassable region consists of the points whose altitudes are higher than the constant altitude of the target, so the probability of the target being at this region is zero. If the coordinates of the target are taken as the states of the system, then it is possible to put some constraints on the states by using the visibility map generated. Such a constraint is that the estimated state should not be at impassable region. These constraints are expected to improve the performance of the tracking algorithm.

The second part of this thesis is about using map information in track initiation algorithms. For that purpose, another special map that we call as “track initiation probability map” (TIPM) is generated. Track initiation probability map gives a measure which is directly related with track initiation probability for each point on the map. The map does not show the exact probability values but a value on the map informs the track initiation algorithm about the possibility of track initiation for this point relative to the other points on the map. This map is used at the initialization part of track initiation algorithms in order to decrease the number of false track initiations and to increase the true track confirmation speed.

1.2 OUTLINE OF THE THESIS

In this thesis, we are interested in a tracking problem for a single target that is not visible by radar due to the elevation structure of the environment. The developed method uses a special map called “visibility map” which consists of three regions called, visible region, invisible region and impassable region.

In Chapter 2, a brief summary of Kalman filtering is given. Constrained state estimation problem and methods that exist in literature to solve this problem are presented. Unscented Kalman filter (UKF) and the projection-type algorithm that uses it is discussed in this part.

Chapter 3 elaborates the problem. Our assumptions are given in this part and the proposed methods are also explained in details.

In Chapter 4, track initiation problem is explained. The algorithm which uses track initiation probability map in track initiation algorithms is explained in this part.

Chapter 5 contains the implementations of the proposed algorithms used for tracking and track initiation that uses the terrain information described in Chapter 3 and Chapter 4. The simulation results and the performance comparisons are also covered.

Chapter 6 is the conclusion part of this thesis. A summary of this study and the future work are given in this part.

In Appendix A, an elevation model of U.S. National Aeronautics and Space Administration (NASA) named as shuttle radar topography mission (SRTM) data, is explained and its properties are introduced.

In Appendix B, the construction of the visibility map used in the simulations, is explained.

CHAPTER 2

STATE ESTIMATION THEORY AND CONSTRAINED STATE ESTIMATION

In Figure 2-1 the flow chart of a basic control system is given. In this figure it is assumed that the observations are corrupted by additive noise. Estimation of the state vector in any control application is very desirable. State can be estimated using the available information, i.e. the measurements.

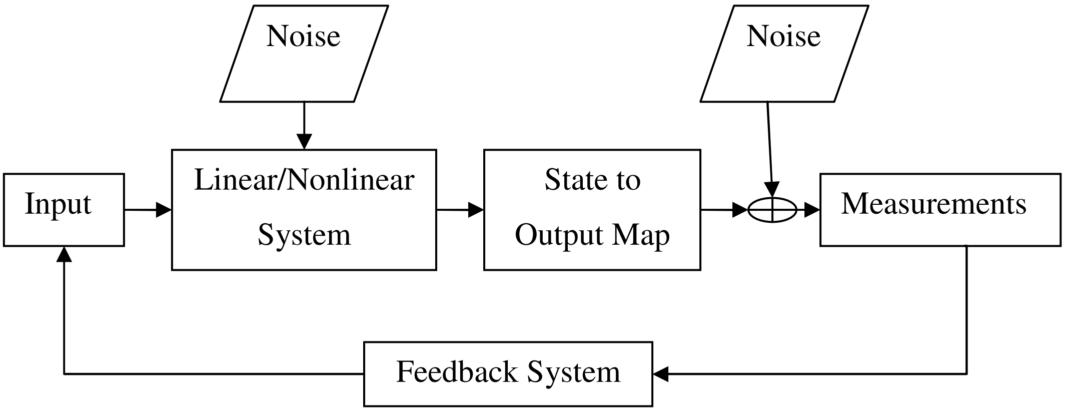

Figure 2-1: Flow chart of a basic control system

Although tracking problem is not a control problem, its main aim is to estimate the position of the target so it can be considered as a state estimation problem. For linear or nonlinear dynamic systems, the behavior of the system can be predicted by estimating the states of this system. For linear systems with Gaussian noise, Kalman filter is preferred estimator because of its optimality [13, 14, 23]. So in this section we will first describe the Kalman filter then give the algorithm of Unscented Kalman filter. We will include a somewhat detailed explanation of the constrained Kalman applications that exist in the literature into this part.

2.1 LINEAR STATE ESTIMATION

A discrete time linear stochastic dynamic system can be described by the state equation (2-1) [13, 14].

$$x_{k+1} = A_k x_k + B_k u_k + G_k w_k \tag{2-1}$$

$$y_k = C_k x_k + H_k v_k \tag{2-2}$$

Here x_k is the state vector, u_k is the known input and w_k is the zero mean white Gaussian process noise with covariance matrix Q. Equation 2-2 is the measurement equation where y_k is the measurement vector and v_k is the measurement noise which is also zero mean white Gaussian and have a covariance matrix R [14]. All the matrices $(A_k, B_k, G_k, C_k, H_k)$ are possibly time varying, and known. The initial state x_0 is a Gaussian random variable with mean 0 and covariance matrix Σ_0. $\{x_0, w_0, w_1, ...w_k, v_0, v_1, ..., v_k\}$ are all independent random variables.

For no input case $(u = 0)$, the system equation becomes:

$$x_{k+1} = A_k x_k + G_k w_k \tag{2-3}$$

$$y_k = C_k x_k + H_k v_k \tag{2-4}$$

2.1.1 Kalman Filter

For the system given in (2-3) and (2-4), when all the requirements given above are satisfied, Kalman filter is the optimal MMSE (minimum mean square estimate) state estimator [12, 13, 14], that is the reason why Kalman filter is widely used in state estimation problems.

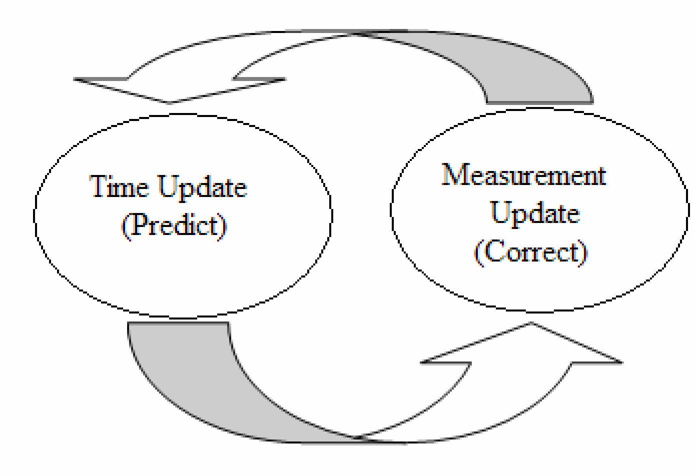

Figure 2-2: Discrete Kalman filter cycle

In Kalman filter, state is estimated recursively. The estimation process has two main steps for each iteration. These are "time update" (state and measurement prediction) and "measurement update" (state update) [12, 13, 14]. The measurement is predicted in the time update phase, which is then used in the measurement update phase by comparing with the exact measurement. Here Y^k denotes the set of all measurements up to time k i.e.

$$Y^k = \{y_0,\dots, y_k\} \tag{2-5}$$

Kalman filter equations can be derived in four steps:

a) Prediction of the state: The conditional probability density function of the state is a Gaussian process which is represented by its mean vector and covariance matrix. The mean vector and the covariance matrix are updated according to the equations (2-6) and (2-7):

$$E\left(x_{k+1} \middle| Y^k\right) = x_{k+1|k} = A_k x_{k|k} \tag{2-6}$$

$$\operatorname{cov}\left(x_{k+1}\middle|Y^k\right)=\Sigma_{k+1|k}=A_k\Sigma_{k|k}A_k^T+G_kQG_k^T \tag{2-7}$$

b) **Prediction of the measurement:** The measurement is predicted using the available data up to time k, i.e. $\{y_i\}_{i=0}^k$. Conditional mean vector is also Gaussian. The equations of the predicted measurement vector and predicted covariance matrix of the measurement vector are given in (2-8) and (2-9).

$$E\left(y_{k+1}\middle|Y^k\right)=y_{k+1|k}=C_{k+1}x_{k+1|k} \tag{2-8}$$

$$\operatorname{cov}\left(y_{k+1}\middle|Y^k\right)=\Sigma_{k+1|k}^y=C_{k+1}\Sigma_{k+1|k}C_{k+1}^T+H_{k+1}RH_{k+1}^T \tag{2-9}$$

c) **Kalman gain equation:** Kalman gain depends on the covariance matrix of the predicted measurement and the correlation between the measurement vector and the state vector. This gain decreases when the measurements are unreliable.

$$L_{k+1}=\Sigma_{k+1|k}C_{k+1}^T\left[C_{k+1}\Sigma_{k+1|k}C_{k+1}^T+H_{k+1}RH_{k+1}^T\right]^{-1} \tag{2-10}$$

d) **Filtered state equations (Estimated state and estimated covariance matrix):** The last step in Kalman filter is the estimation of the state vector and the covariance matrix of the state vector by using the predicted state and the predicted measurements.

$$E\left(x_{k+1}\middle|Y^{k+1}\right)=x_{k+1|k+1}=x_{k+1|k}+L_{k+1}\left(y_{k+1}-y_{k+1|k}\right) \tag{2-11}$$

$$\operatorname{cov}\left(x_{k+1}\middle|Y^{k+1}\right)=\Sigma_{k+1|k+1}=\left(I-L_{k+1}C_{k+1}\right)\Sigma_{k+1|k} \tag{2-12}$$

The first step is the prediction of the state and the measurement vector. After this prediction, the estimated state is calculated as the sum of predicted state and the difference between the exact and predicted measurements, i.e, the innovation process. Here L is the Kalman gain which shows us whether our measurements are trustable or not. When the amount of noise at the measurements is high, this gain is relatively low so the estimated state vector and covariance matrix are closer to the predicted state vector and covariance matrix.

The covariance matrix update can be done offline as it doesn't depend on the measurements. The iteration starts with $k = 0$. Initially we need $x_{0|-1}$ which is equal to the initial state x_0. We explained that the initial state is also a random variable so (2-13) can be used for initialization [14].

$$x_{0|-1} = x_0 = E(x_0) \tag{2-13}$$

Similarly the initial covariance matrix is:

$$\Sigma_{0|-1} = \Sigma_0 \tag{2-14}$$

By using the initial conditions at (2-13) and (2-14), the state and the covariance matrix can be estimated at each iteration.

2.2 UNSCENTED KALMAN FILTER

Kalman filtering is the optimal procedure when the system satisfies linearity and Gaussianity assumptions. However in case of nonlinearity either in state or measurement equation, it can not be used. One attempt to use Kalman filter is to linearize the nonlinear equations at each step. Such an approach gives us Extended Kalman filter (EKF). In the last decade, a new method called Unscented Kalman filter (UKF) is introduced and become popular because of its good performance.

In our study although we assume that the state and measurement equations are linear, we modify them because of the constraints. This modification introduces "nonlinearity" to the system and one way of modeling the modification is to use the idea of UKF.

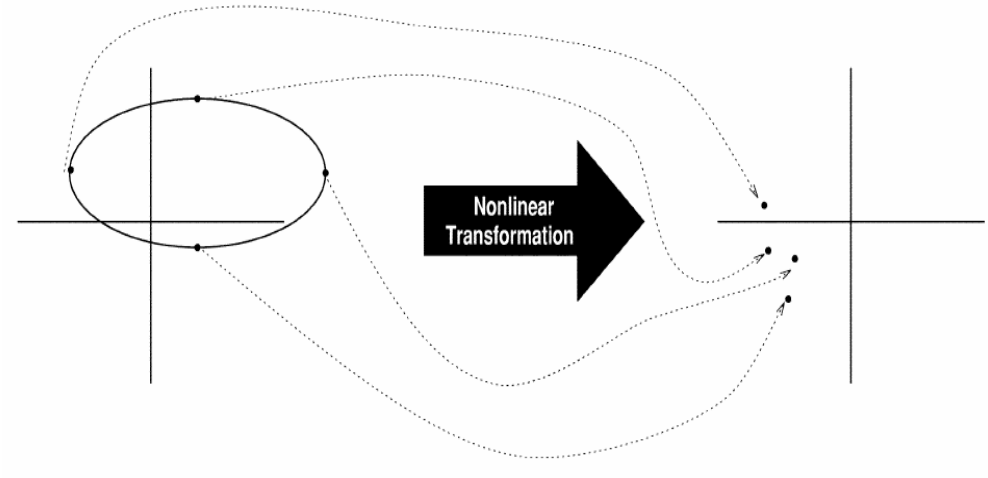

Figure 2-3: The principle of the Unscented Transformation [6].

UKF is based on approximating the probability distribution function by a few points. EKF assumes that the conditional density function of the state and the measurement are Gaussian. UKF makes the same assumption, i.e. the conditional probability density functions of the state and the measurement are Gaussian. The difference is that UKF applies no linearization but it transforms the density $P_{k|k}(x_k)$ by a fixed number of points called sigma points. These sigma points are represented as $\{x_k^{(i)}\}_{i=1}^{M}$ and are propagated in time using the nonlinear state function [6, 7, 23]. In computations, instead of using the whole density function only the sigma points are used. So the nonlinear transformation is applied on a few points.

In Figure 2-3, on the plot at the left hand side we see the density function and the chosen sigma points which will be transformed by the nonlinear function of the system. We choose a set of sigma points S, according to the rule that their calculated mean vector and calculated covariance matrix are equal to the mean vector and the covariance matrix of the state respectively. The set S consists of $(p+1)$ vectors $x^{(i)}$, and their associated weights $W^{(i)}$ such that $S=\{x^{(i)},W^{(i)}\}_{i=0}^{p}$. p

is usually chosen as $2N_x$ where N_x is the order of the system. The weights $W^{(i)}$ can be positive or negative but they must satisfy (2-15) [6, 7].

$$\sum_{i=0}^{p} W^{(i)} = 1 \tag{2-15}$$

An example of a symmetric set of sigma points for a probability distribution of the state with mean vector $\bar{x}$ and covariance matrix Σ_x is given below [6]. The notation $(A)_i$ is the i^{th} row or column of the matrix A. $W^{(0)}$ is a free parameter which represents the weight on the mean.

$$x^{(0)} = \bar{x} \tag{2-16}$$

$$x^{(i)} = \bar{x} + \left(\sqrt{\frac{N_x}{1-W^{(0)}} \Sigma_x} \right)_i , \qquad i = 1,\dots,N_x \tag{2-17}$$

$$x^{(i+N_x)} = \bar{x} - \left(\sqrt{\frac{N_x}{1-W^{(0)}} \Sigma_x} \right)_i , \qquad i = 1,\dots,N_x \tag{2-18}$$

$$W^{(i)} = W^{(i+N_x)} = \frac{1-W^{(0)}}{2N_x} , \qquad i = 1,\dots,N_x \tag{2-19}$$

We give below the UKF for a special system where the state equation is in the form of additive noise i.e.:

$$x_{k+1} = f_k(x_k) + w_k \tag{2-20}$$

$$y_k = h_k(x_k) + v_k \tag{2-21}$$

When the sigma points are determined, they are transformed by the nonlinear function $f(.)$ so that the predicted state will be $\hat{x}_n^{(i)} = f\left(x_n^{(i)}, u_n\right)$. The predicted mean and covariance can be computed as below [6].

$$\hat{\mu}_n = \sum_{i=0}^{p} W^{(i)} \hat{x}_n^{(i)} \tag{2-22}$$

$$\hat{K}_n = \sum_{i=0}^{p} W^{(i)} \left(\hat{x}_n^{(i)} - \hat{\mu}_n\right)\left(\hat{x}_n^{(i)} - \hat{\mu}_n\right)^T \tag{2-23}$$

The sigma points are found again using the predicted mean vector and covariance matrix of the state. Each of the sigma points is transformed by the measurement model $h(.)$ as in (2-24). The predicted mean vector and predicted covariance (innovation covariance) matrix are calculated as in (2-25) and (2-26). (2-27) is the cross covariance matrix of the state and the measurement.

$$\hat{y}_n^{(i)} = h\left(\hat{x}_n^{(i)}, u_n\right) \tag{2-24}$$

$$\hat{y}_n = \sum_{i=0}^{p} W^{(i)} \hat{y}_n^{(i)} \tag{2-25}$$

$$\hat{S}_n = \sum_{i=0}^{p} W^{(i)} \left(\hat{y}_n^{(i)} - \hat{y}_n\right)\left(\hat{y}_n^{(i)} - \hat{y}_n\right)^T \tag{2-26}$$

$$\hat{K}_n^{xy} = \sum_{i=0}^{p} W^{(i)} \left(\hat{x}_n^{(i)} - \hat{\mu}_n\right)\left(\hat{y}_n^{(i)} - \hat{y}_n\right)^T \tag{2-27}$$

The measurement update is performed using normal Kalman filter equations.

One step of the algorithm is shown below.

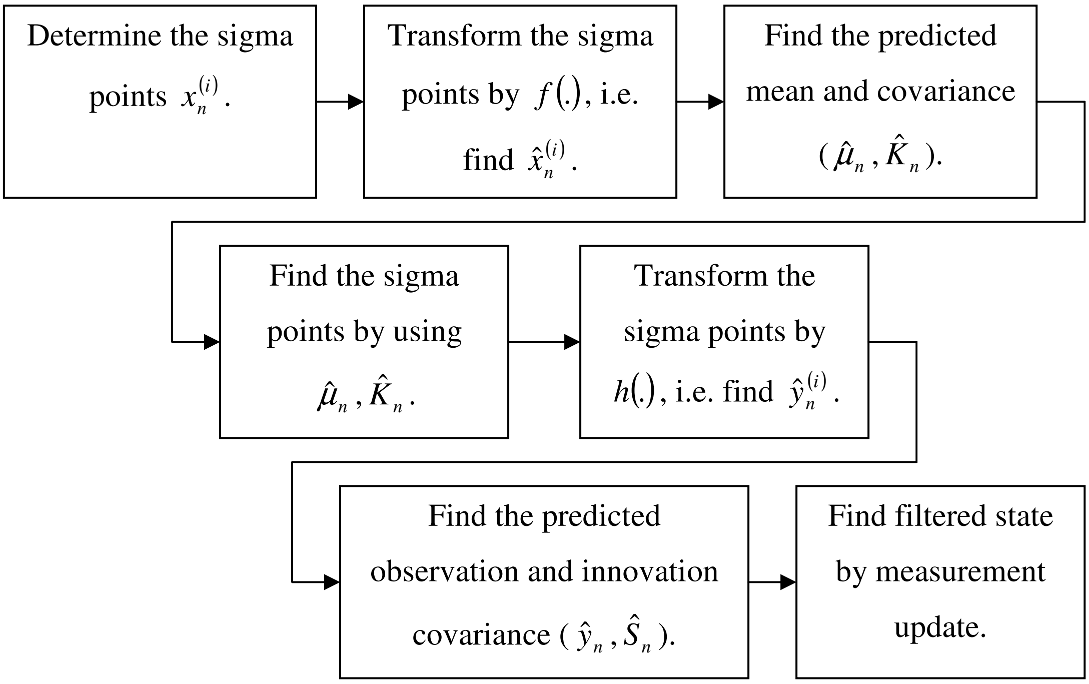
Determine the sigma points $x_n^{(i)}$.
Transform the sigma points by $f(.)$, i.e. find $\hat{x}_n^{(i)}$.
Find the predicted mean and covariance $(\hat{\mu}_n, \hat{K}_n)$.
Find the sigma points by using $\hat{\mu}_n, \hat{K}_n$.
Transform the sigma points by $h(.)$, i.e. find $\hat{y}_n^{(i)}$.
Find the predicted observation and innovation covariance $(\hat{y}_n, \hat{S}_n)$.
Find filtered state by measurement update.

Figure 2-4: One step of UKF.

2.3 CONSTRAINED LINEAR STATE ESTIMATION

Some of the KF applications require constraining the state and/or measurement to a set of the state space or a subspace of the state space. One example is the 2-dimensional tracking of an object moving on a road. In this example the 2-d movement of the target is clearly restricted to a line that can be considered as a one-dimensional space. Such restrictions are not well suited to KF theory. In this section we will review the methods that exist in literature that are concentrated on solving constraint state KF problems.

We clarify the existing literature on constrained Kalman filter into two categories. They are mainly projection-type algorithms [1, 2, 3] and UKF methods [7, 20, 21]. These methods can be applied to the equality constraints as well as the inequality constraints. Below we give a brief explanation of them.

2.3.1 Projection-Type Algorithms

The methods under this name are the ones given in references [1, 2, 3]. They are Maximum Probability method, Mean Square method and Projection method. All of these three methods rely on the principle of projection so that they are all grouped under the name of "projection-type algorithms".

2.3.1.1 The Maximum Probability Method

Generally a state equality constraint is given as in (2-28) [1, 3].

$$Dx_k = d_k \tag{2-28}$$

Here D is a known sxn constant matrix, d_k is a known $sx1$ vector, s is the number of constraints and n is the number of states and generally $s \le n$ [1, 3].

It is known that the Kalman filter estimate is that value of x that maximizes the conditional density function $p(x \mid Y)$. The constrained Kalman filter with Maximum Probability method can be found by finding an estimate $\tilde{x}$ such that the conditional probability density $p(\tilde{x} \mid Y)$ is maximized and $\tilde{x}$ satisfies the constraint (2-28) [1, 2, 3]. So the problem is:

$$\max\left(\ln\{p(\tilde{x}|y)\}\right) \rightarrow \min\left((\tilde{x}-\bar{x})^T \Sigma^{-1} (\tilde{x}-\bar{x})\right) \tag{2-29}$$

such that $D\tilde{x} = d$. Here $\bar{x}$ represents the conditional mean of the state.

Constraint minimization problem defined above can be solved by Lagrangian relaxation as written below.

$$L = (\tilde{x}-\bar{x})^T \Sigma^{-1} (\tilde{x}-\bar{x}) + 2\lambda(D\tilde{x}-d), \text{ when } \lambda \ge 0 \tag{2-30}$$

Note that λ is a scalar value. The solution is obtained as:

$$\frac{\partial L}{\partial \tilde{x}} = 0 \rightarrow \Sigma^{-1}(\tilde{x}-\bar{x}) + D^T \lambda = 0 \tag{2-31}$$

$$\frac{\partial L}{\partial \lambda} = 0 \rightarrow d = D\tilde{x} \tag{2-32}$$

The solution is:

$$\lambda = \left(D\Sigma D^{T}\right)^{-1}\left(D\bar{x} - d\right) \tag{2-33}$$

$$\tilde{x} = \bar{x} - \Sigma D^{T}\left(D\Sigma D^{T}\right)^{-1}\left(D\bar{x} - d\right) \tag{2-34}$$

As the conditional mean of x, $\bar{x}$, is the Kalman filter estimate, the constrained Kalman estimate $\tilde{x}$ can be derived from unconstrained estimate $\hat{x}$ as,

$$\tilde{x} = \hat{x} - \Sigma D^{T}\left(D\Sigma D^{T}\right)^{-1}\left(D\hat{x} - d\right) \tag{2-35}$$

2.3.1.2 The Mean Square Method

In this method the aim is to minimize the mean square error by satisfying the equality constraint (2-28) [1, 2, 3].

$$\min_{\tilde{x}}\left(E\left(\left\|x - \tilde{x}\right\|^{2} \middle| Y\right)\right) \tag{2-36}$$

such that $D\tilde{x} = d$, where $\|.\|$ denotes the L_2 norm. When x and Y are jointly distributed,

$$\begin{aligned} E\left(\left\|x - \tilde{x}\right\|^{2} \middle| Y\right) &= \int (x - \tilde{x})^{T}(x - \tilde{x})p(x|Y)dx \\ &= \int x^{T}xp(x|Y)dx - 2\tilde{x}^{T}\int xp(x|Y)dx + \tilde{x}^{T}\tilde{x} \end{aligned} \tag{2-37}$$

Now the Lagrangian is,

$$\begin{aligned} L &= E\left(\left\|x - \tilde{x}\right\|^{2} \middle| Y\right) + 2\lambda(D\tilde{x} - d) \\ &= \int x^{T}xp(x|Y)dx - 2\tilde{x}^{T}\int xp(x|Y)dx + \tilde{x}^{T}\tilde{x} + 2\lambda(D\tilde{x} - d) \end{aligned} \tag{2-38}$$

The first order conditions can be formulated as

$$\frac{\partial L}{\partial \tilde{x}} = 0 \rightarrow -2\hat{x} + 2\tilde{x} + 2D^{T}\lambda = 0 \tag{2-39}$$

$$\frac{\partial L}{\partial \lambda} = 0 \rightarrow D\tilde{x} - d = 0 \tag{2-40}$$

That gives the solution as follows.

$$\lambda = \left(DD^T\right)^{-1}\left(D\hat{x} - d\right) \tag{2-41}$$

$$\tilde{x} = \hat{x} - D^T\left(DD^T\right)^{-1}\left(D\hat{x} - d\right) \tag{2-42}$$

2.3.1.3 The Projection Method

In this method the unconstrained state estimate of Kalman filter is directly projected on the constrained surface [1, 2, 3]. That means solving (2-43)

$$\min_{\tilde{x}}(\tilde{x} - \hat{x})^T W(\tilde{x} - \hat{x}) \tag{2-43}$$

such that $D\tilde{x} = d$ where W is a symmetric positive definite matrix. The solution of (2-43) is:

$$\tilde{x} = \hat{x} - W^{-1}D^T\left(DW^{-1}D^T\right)^{-1}\left(D\hat{x} - d\right) \tag{2-44}$$

As stated in [1, 2], Maximum Probability method and Mean Square method are the special cases of the Projection method. In (2-44) when W is chosen as $W = \Sigma^{-1}$, we have the solution of Maximum Probability method (2-35). For $W = I$ we have the solution for Mean Square method, (2-42) [1, 2, 3].

2.3.2 Unscented Kalman Filter

UKF is a state estimation method for nonlinear systems which can also be used in constrained linear state estimation. The algorithm given in [7] is explained below:

For each iteration:

a) Calculate the sigma points using the covariance matrix of the state. Then project the points which are outside the feasible region (constrained region) to the boundary between the constrained region and the unconstrained region in order to obtain the constrained sigma points.

b) Transform the sigma points using the state update function (time update). Check whether the transformed sigma points are in the constrained region or not. If they are outside the constrained region project them to obtain the constrained transformed sigma points. By using these sigma points calculate the mean vector and covariance matrix of the predicted state.

c) Find the sigma points from the distribution of the predicted state. Transform the sigma points using the measurement update function (measurement update). If the estimate violates the constraints, apply the projection process again.

In Figure 2-4 we see the schematic representation of the UKF algorithm for constrained state estimation [7]. In projection-type algorithms only the state estimate is updated by the use of constraints, while in UKF both the state vector and covariance matrix estimate is updated using constraints. Updating the covariance matrix in UKF provides smaller covariance matrices than the covariance matrices used in the projection-type algorithms.

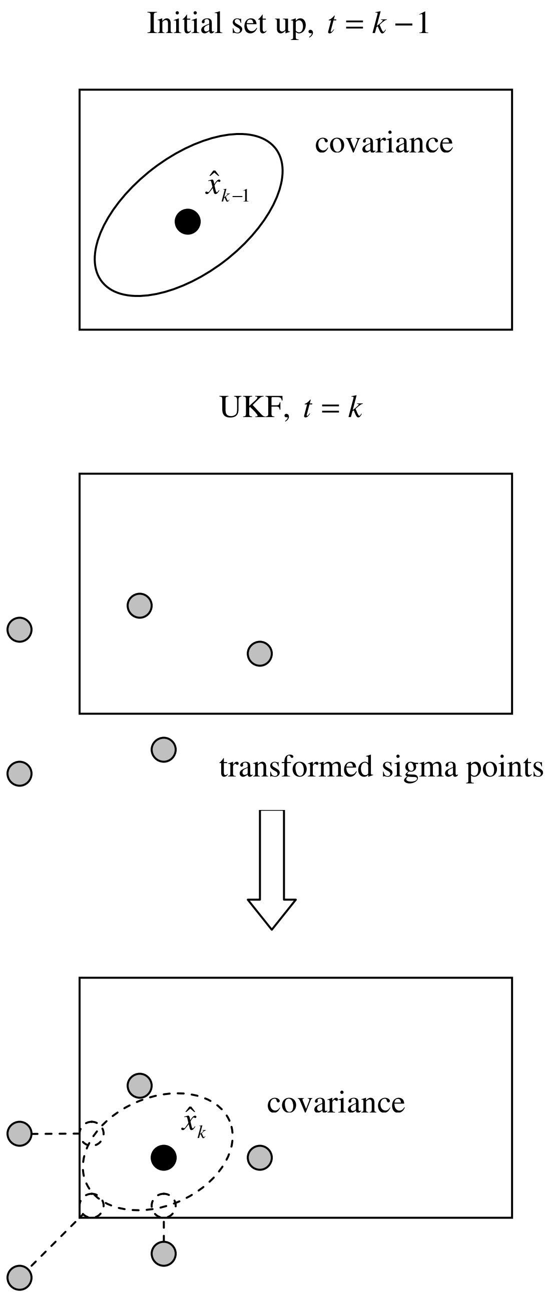

Figure 2-5: Constrained State Estimation with UKF [7].

CHAPTER 3

RADAR TARGET TRACKING USING ELEVATION INFORMATION OF THE TERRAIN

3.1 PROBLEM DEFINITON

In tracking problems, target tracker and the target may not be in line of sight which causes tracking losses. For example, when the target is obstructed by a mountain, tracking may end up. In such a case, knowing the elevation structure of the environment and using this information in tracking algorithms may overcome the problem. In this work, the elevation structure of the terrain is used to obtain a visibility map of the tracker under some assumptions. First these assumptions are defined then the methods are introduced.

3.1.1 Assumptions

- The target follows a path on a plane with a constant height h. A visibility map, obtained using this constant height, will be used.
- In order to simplify the problem, a model is built with a single target and a single target tracker.
- Radar location is fixed and known, (x_r, y_r).
- Curvature of the earth is not taken into consideration.

It is obvious that the assumptions are not realistic. However they can be relaxed by some effort and furthermore our aim is to demonstrate that map information is useful.

3.1.2 Generation of the Visibility Map Information

To obtain the visibility map of a radar tracker the elevation model of the terrain should be known. The elevation data called as Shuttle Radar Topography Mission (SRTM) used in this work is taken from the web site of National Aeronautics and Space Administration (NASA) [24]. SRTM data, shown in Figure 3-1 and Figure 3-2, is a 301x301 matrix, whose elements are the elevation of the terrain. Originally the sample spacing of individual data points is 90 meters. Because the SRTM data is down sampled by a factor of 4 in order to make the computations faster, the sample spacing becomes 360 meters. Detailed information about SRTM data can be found in Appendix-A.

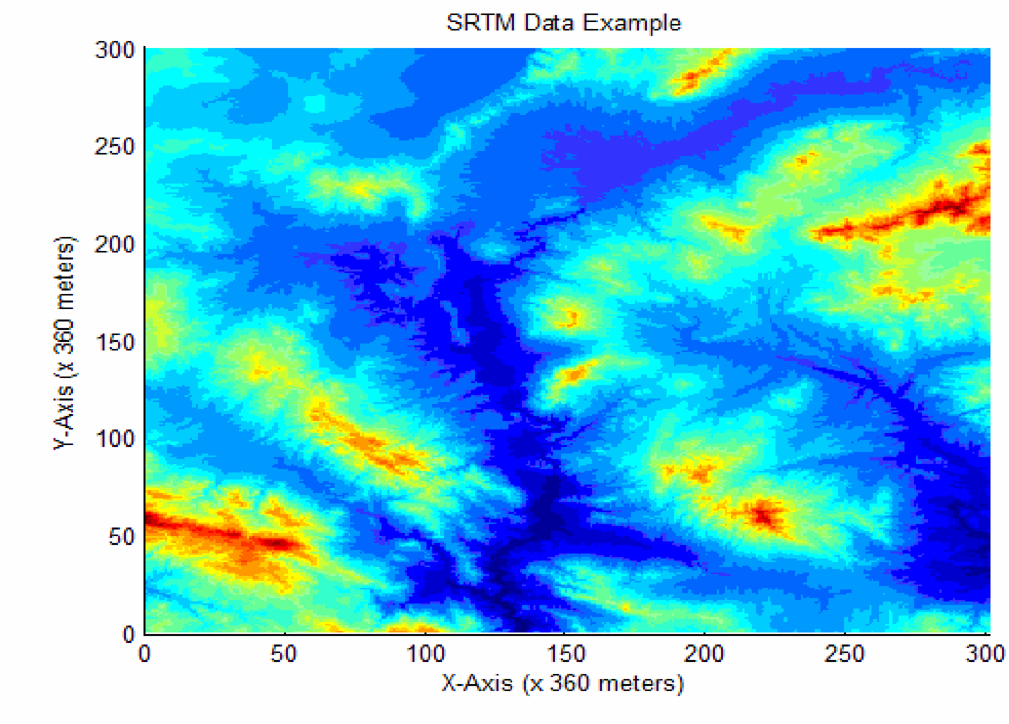

Figure 3-1: Elevation map data (SRTM data) top view.

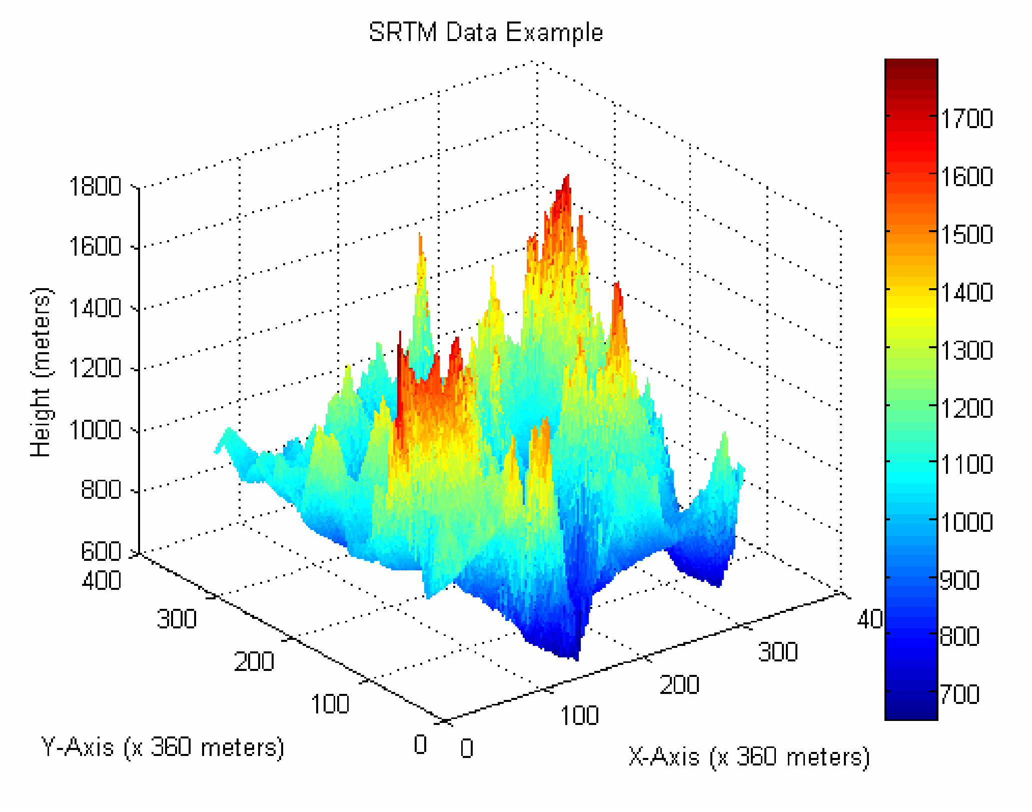

Figure 3-2: Elevation map data (SRTM data) 3-D view.

For a radar located at (x_r, y_r) and for a constant target height h, the visibility map of this radar can be obtained [Appendix-B]. In visibility map there are three regions; visible region, invisible region, impassable region. The points where the target is in line of sight of radar composes the visible region while the invisible region is formed by the points where the target is not in line of sight. The points which are located at a higher altitude than the constant height h of the target, forms the impassable region. An example of this map is shown in Figure 3-3. The target height is $h = 1440$ meters and the radar location is $(x_r, y_r) = (153,163)$ for this example.

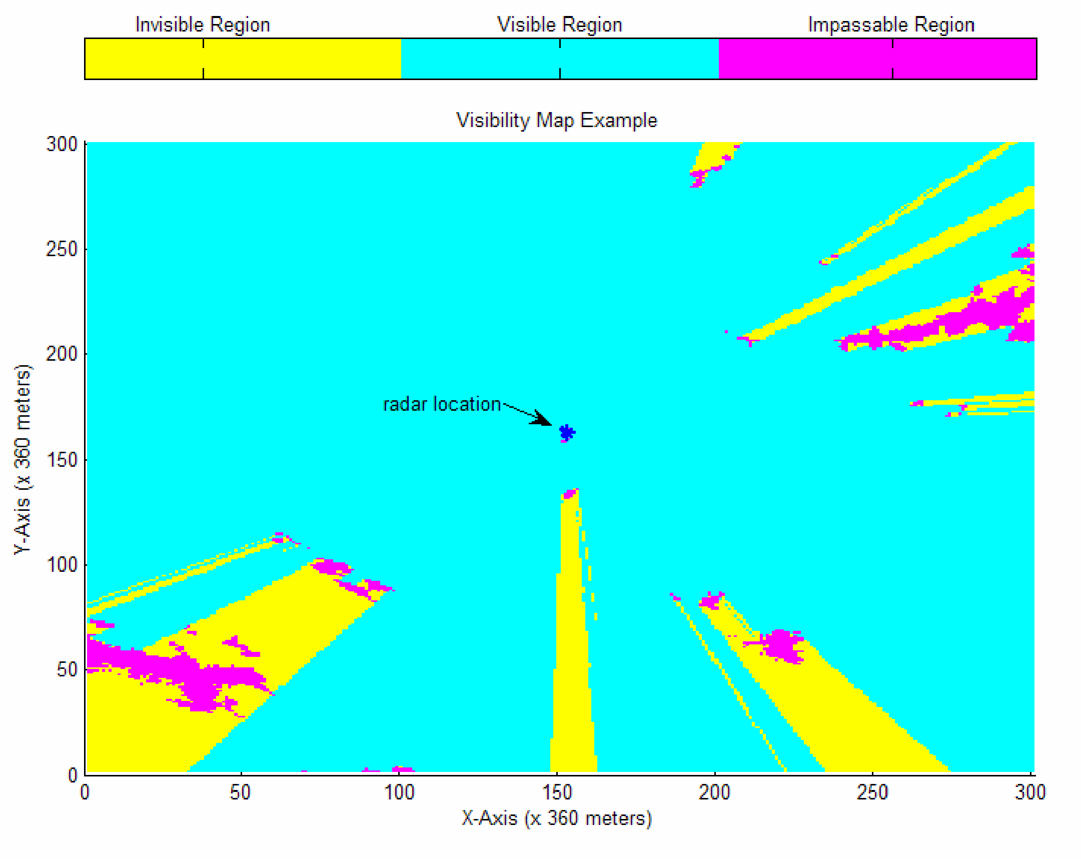

Figure 3-3: Visibility map example.

3.2 METHODS USED FOR CONSTRAINED STATE ESTIMATION

No measurements can be taken when the target enters into an invisible region. The aim of this section is to use the information that the target can not be at impassable region and modify its trajectory with respect to this information. We will demonstrate that it will help to recover the track with correct identification as it reappears at the boundary of the invisible region. This approach requires putting some restrictions, i.e., it can not pass through the impassable region, onto the trajectory hence the state of the target motion. The basic tool of state estimation done for tracking is Kalman filtering. So an analysis is given here to obtain the estimation of the state under linear state constraints.

3.2.1 Constrained State Kalman Filter

In this work it is assumed that the target is moving in a constant known altitude. Its motion model is a "constant velocity" model and will be tracked by Kalman filter. Under these assumptions, measurement space is a 2-D plane where each point is denoted by (x, y). Kalman filter gives measurement predictions which are Gaussian, under no constraints on the measurements. Below, we will give an analysis on inserting some constraints to the measurement prediction densities. To do that first we will transform the given Gaussian into the standard one then calculate the conditional probability of it to be in the region constrained by a line and finally define a new Gaussian restricted to the given region with high probability.

3.2.1.1 Obtaining the Standard Gaussian Distribution

The Gaussian distribution with zero mean vector and unit covariance matrix is defined as the standard Gaussian distribution. For a given Gaussian distribution with mean $[a \quad b]^T$ and covariance matrix Σ, the probability density function is defined as:

$$f(x, y) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}}\exp(-\frac{1}{2}[x-a \quad y-b]\Sigma^{-1}[x-a \quad y-b]^T)$$
$$\Sigma = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} \quad -1 < \rho < 1 \tag{3-1}$$

Covariance matrix can be written as $\Sigma = T^T \Lambda T$, where $T^{-1} = T^T$. $\Lambda = diagonal(\lambda_i)$ such that $\{\lambda_i\}_{i=1}^n$ are eigenvalues of matrix Σ.

Using these formulations, a new random vector with zero mean and unity covariance can be obtained as:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = F \begin{bmatrix} x-a \\ y-b \end{bmatrix}; \qquad F = \Lambda^{-1/2} T \tag{3-2}$$

With (3-2), the problem is transformed into a new coordinate system. In our formulation, constraint is represented by a straight line of the form $y = \alpha x + \beta$. Under the above defined transformation, line will be transformed as follows:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = F \begin{bmatrix} x-a \\ y-b \end{bmatrix} = F \begin{bmatrix} x-a \\ \alpha x + \beta - b \end{bmatrix} = F \begin{bmatrix} 1 \\ \alpha \end{bmatrix} x + F \begin{bmatrix} -a \\ \beta - b \end{bmatrix} = \begin{bmatrix} m_1 x + n_1 \\ m_2 x + n_2 \end{bmatrix} = \begin{bmatrix} \bar{x} \\ \mu \bar{x} + \eta \end{bmatrix} \tag{3-3}$$

So the equation of the line is $\bar{y} = \mu \bar{x} + \eta$. The condition $m_1 \neq 0$ should be satisfied in order to calculate variables μ and η.

μ and η can be calculated using Equation 3-5 as given below:

$$\bar{x} = m_1 x + n_1 \Rightarrow x = \frac{\bar{x}}{m_1} - \frac{n_1}{m_1} \tag{3-4}$$

$$\begin{aligned} \bar{y} &= m_2 \left(\frac{\bar{x}}{m_1} - \frac{n_1}{m_1} \right) + n_2 \Rightarrow \bar{y} = \frac{m_2}{m_1} \bar{x} + \left(n_2 - n_1 \frac{m_2}{m_1} \right) \\ &\Rightarrow \mu = \frac{m_2}{m_1}, \eta = \left(n_2 - n_1 \frac{m_2}{m_1} \right) \end{aligned} \tag{3-5}$$

Note that m_1, n_1, m_2, n_2 are obtained using the matrix F and (a, b).

3.2.1.2 Regional Probability Calculation

In this section, the probability of a standard Gaussian distributed 2-D random variable to be at one side of a line will be calculated. Figure 3-4 demonstrates the situation. Here the aim is to find the probability of the vector $[x' \quad y']^T$ to be at the blue shaded region. Required probability can be calculated by integrating the probability density function over the shaded region. In our problem, this probability calculation will be used in order to find predicted mean vector and covariance matrix of the measurements under the constraint of not being at impassable region.

$$\Pr\left\{\begin{bmatrix}x'\\y'\end{bmatrix}\in B\right\}=\int_{-\infty}^{d}\int_{-\infty}^{\infty}f(x',y')dy'dx'=\int_{-\infty}^{d}\int_{-\infty}^{\infty}f(x')f(y')dy'dx'=\frac{1}{2}+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2} \quad (3\text{-}6)$$

In order to simplify the calculations the transformation given in Equation 3-2 can be applied. The transformed coordinate axes are shown by red lines on Figure 3-4. Note that this transformation doesn't affect the probability value calculated.

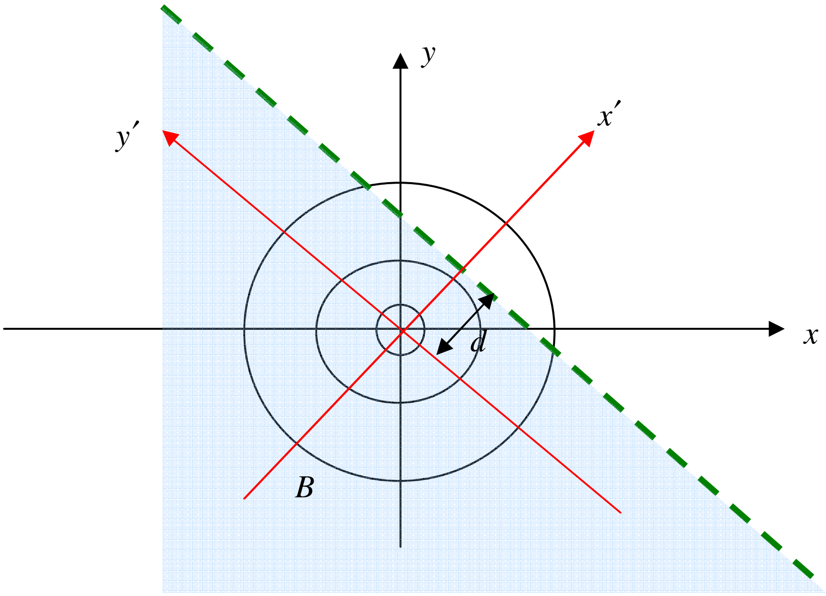

Figure 3-4: Regional probability calculation for the region B .

3.2.1.3 Mean Value Calculation over a Region

In this part, we will calculate the conditional mean of a random variable whose probability density function is constrained to be at one side of a line. The coordinate transformation used in the previous section is applied again. As can be seen from the calculations, the mean value of y' is unchanged while the mean value of x' takes a negative value depending on d .

$$E\left\{\begin{bmatrix}x'\\y'\end{bmatrix} \mid \begin{bmatrix}x'\\y'\end{bmatrix}\in B\right\}=\int_{-\infty}^{d}\int_{-\infty}^{\infty}\begin{bmatrix}x'\\y'\end{bmatrix}\frac{f(x',y')}{\Pr(B)}dy'\,dx'=\int_{-\infty}^{d}\int_{-\infty}^{\infty}\begin{bmatrix}x'\\y'\end{bmatrix}\frac{f(x')f(y')}{\Pr(B)}dy'\,dx'$$

$$=\frac{1}{\Pr(B)}\begin{bmatrix}\int_{-\infty}^{d}x'f(x')dx'\\0\end{bmatrix}=\begin{bmatrix}\frac{-1}{\sqrt{2\pi}}e^{-d^2/2}\frac{1}{\left(0.5+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2}\right)}\\0\end{bmatrix} \qquad (3\text{-}7)$$

3.2.1.4 Covariance Matrix Calculation over a Region

In this section the process to find the regional mean value is repeated in order to find the covariance matrix. For that purpose first the correlation matrix is found. The correlation matrix can be found with the given condition $\begin{bmatrix}x'\\y'\end{bmatrix}\in B$ as in Equation 3-8.

$$E\left\{\begin{bmatrix}x'\\y'\end{bmatrix}\begin{bmatrix}x' & y'\end{bmatrix} \mid \begin{bmatrix}x'\\y'\end{bmatrix}\in B\right\}=\int_{-\infty}^{d}\int_{-\infty}^{\infty}\begin{bmatrix}(x')^2 & x'y'\\x'y' & (y')^2\end{bmatrix}\frac{f(x',y')}{\Pr(B)}dy'\,dx'$$

$$=\frac{1}{0.5+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2}}\int_{-\infty}^{d}\int_{-\infty}^{\infty}\begin{bmatrix}(x')^2 & x'y'\\x'y' & (y')^2\end{bmatrix}f(x')f(y')dy'\,dx'$$

$$=\frac{1}{0.5+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2}}\begin{bmatrix}\int_{-\infty}^{d}(x')^2f(x')dx' & 0\\0 & 0.5+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2}\end{bmatrix}$$

$$=\begin{bmatrix}\frac{-de^{-d^2/2}}{\left(0.5+\frac{erf\left(\frac{d}{\sqrt{2}}\right)}{2}\right)\sqrt{2\pi}}+1 & 0\\0 & 1\end{bmatrix} \qquad (3\text{-}8)$$

The required covariance matrix can be calculated as the difference between the matrix found above and the outer product of the mean values.

$$\operatorname{var}\left(\begin{bmatrix} x' \\ y' \end{bmatrix} \mid \begin{bmatrix} x' \\ y' \end{bmatrix} \in B\right) = \begin{bmatrix} G & 0 \\ 0 & 1 \end{bmatrix}; \quad G = -\frac{e^{-d^2/2}}{\sqrt{2\pi}\,\Pr(B)}\left(d + \frac{e^{-d^2/2}}{\sqrt{2\pi}\,\Pr(B)}\right) + 1 \quad (3\text{-}9)$$

Note that the transformations by rotation F and translator $\begin{pmatrix} a \\ b \end{pmatrix}$ done at the beginning of this section and the final transformation of rotating the coordinates are reversible. We have obtained the mean and the covariance matrix conditioned to a region defined by a line on the transformed space. Now we will make the assumption that the conditional density restricted to the given area is Gaussian and represent it by the computed mean and the covariance.

The Algorithm:

According to the assumption that the target is in an invisible region at time $(k-1)$, the algorithm for providing surviving tracks after the reappearance of the targets is developed as follows:

1- Find the line equation that separates the invisible and impassable regions at time $(k-1)$ as in Figure 3-5.

2- Calculate the parameters (F matrix and a, b) which transforms the given distribution to a standard distribution. Use these parameters to convert the given distribution to the standard form.

3- Apply the same transformation to the line equation which gives the invisible region.

4- Using the formulations obtained in Equation 3-7 and Equation 3-9, calculate the constrained mean vector and the constrained covariance matrix.

5- Apply the inverse transformation to return to the original coordinates from transformed coordinates.

6- Predict the mean vector and the covariance matrix for time k.

7- Find the line that determines the boundary of the invisible region at time k.

8- If the target is visible at time k continue with the KF, otherwise go to Step 1.

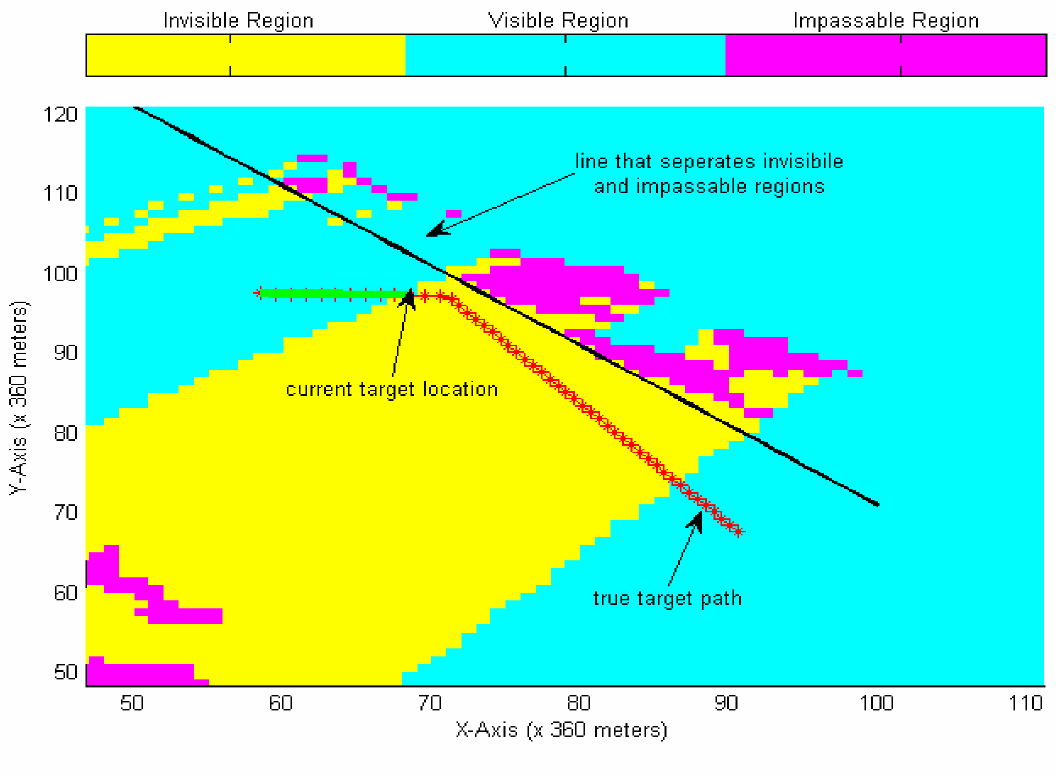

Figure 3-5: Separation of invisible and impassable regions.

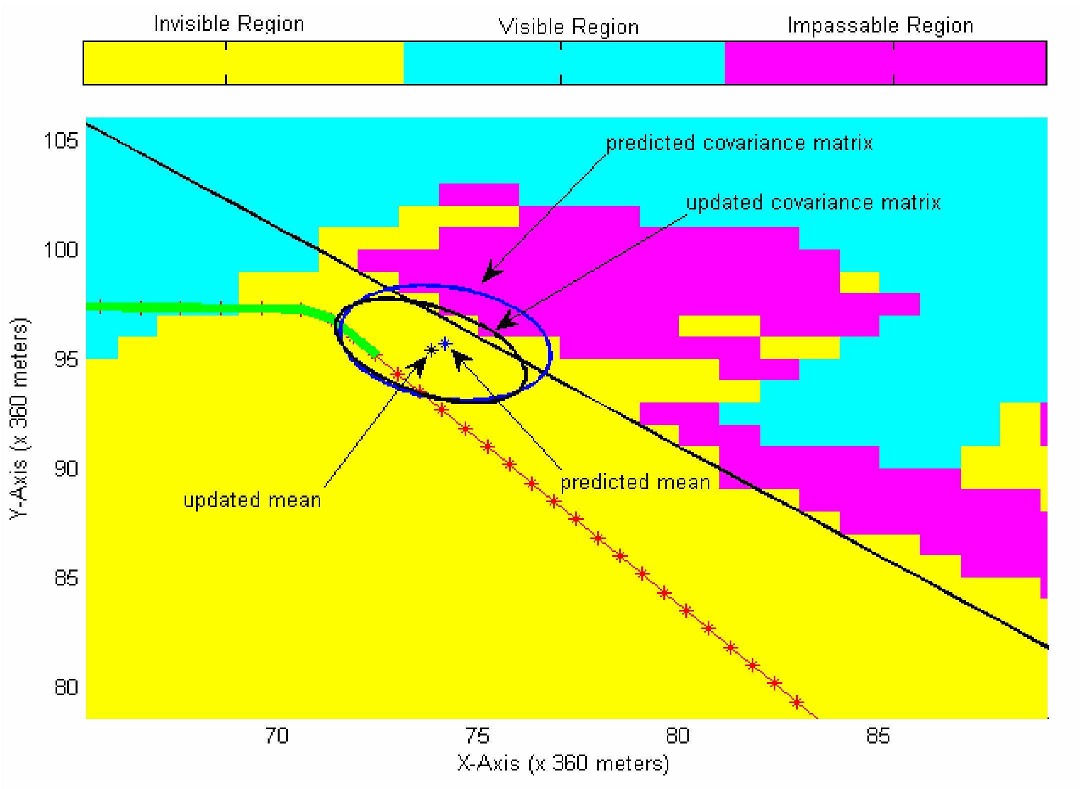

Figure 3-6: Predicted and updated covariance matrix.

In Figure 3-6, the schematic representations of the predicted and updated covariance matrices are shown. It is evident that the updated covariance matrix is smaller than the predicted covariance matrix as the visibility map information is used by the algorithm. If this information is not used, then the predicted and updated covariance matrices would be identical (no measurement update).

3.2.2 Unscented Kalman Filtering with Constraining the State

The flow diagram of the algorithm is similar to the algorithm given in Section 3.2.1. The truncation is done only with the sigma points instead of the whole density function. But here, there is no need to make coordinate transformation. Algorithm is explained for the case that the target is in the invisible region at time $(k-1)$.

1- Find the line equation that separates the invisible and impassable regions at time $(k-1)$. An example line is shown in Figure 3-5.

2- Using Kalman filter equations (2-8) and (2-9), find the predicted mean vector and covariance matrix of the measurement vector $\left(y_{k|k-1}, \Sigma^{y}_{k|k-1}\right)$.

3- For the density function $N\left(y_{k|k-1}, \Sigma^{y}_{k|k-1}\right)$, find the sigma points and their associated weights using the equations (2-16) to (2-19).

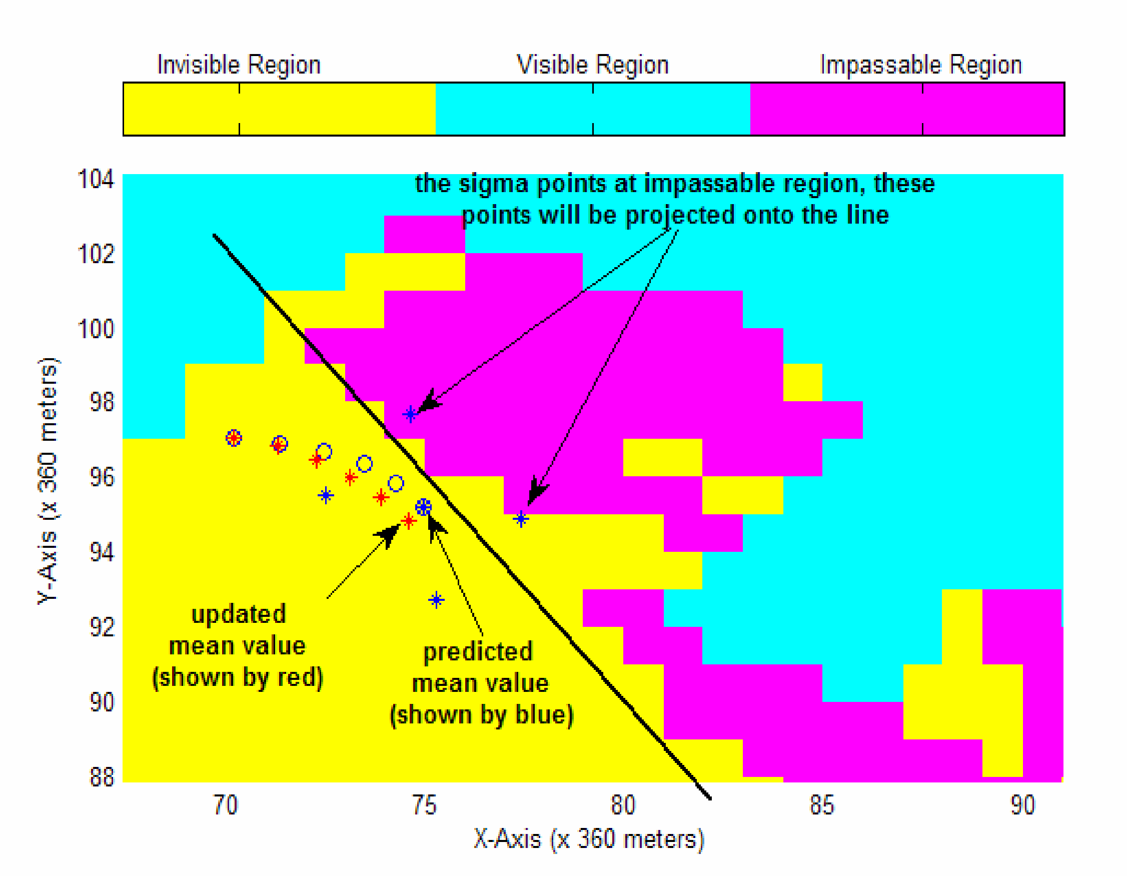

Figure 3-7: Projection of sigma points.

4- Check whether the sigma points are at impassable region or not. Project the sigma points which are at impassable region onto the line found in (1). The points that will be projected onto the line are shown in Figure 3-7.

5- Find the modified mean and covariance matrix of the measurements using the equations (2-25) and (2-26), i.e. compute $y_{k|k-1}^{\mathrm{mod}\,ified}, \Sigma_{k|k-1}^{y,\mathrm{mod}\,ified}$.

6- Take the modified mean as a pseudo measurement and the modified covariance as the covariance of this pseudo measurement. Update the Kalman filter equations (2-11) and (2-12) with the pseudo measurement.

For the next iteration if the target is still at invisible region (no measurement taken from target) repeat the algorithm. If it is visible, use normal Unscented Kalman filter equations. Note that in our application since the system is linear Gaussian, if the target is in the visible region, UKF and KF give identical results.

CHAPTER 4

TRACK INITIATION USING VISIBILITY MAP INFORMATION

Track initiation (track formation) is the process of starting a new track. Target tracker operation can be described with track initiation, tracking and track termination steps, respectively. Even if the tracking algorithm works perfectly, a false track initiation or a late track confirmation decreases the performance of the target tracker. For that reason track initiation step carries almost the same importance as the tracking step.

For a single target, due to clutter, thermal noise, or any other reasons there may be a number of measurements with uncertain origin which makes the track initiation process problematic. Data association algorithms are needed for this situation. Probabilistic data association (PDA) is a well known and widely used method for tracking and data association [15]. But this algorithm works under the assumption that a track exists [15]. This assumption is not reasonable for an uncertain environment as most of the detections are originated from clutter or thermal noise which causes unreliable measurements. Integrated probabilistic data association (IPDA) is used to overcome this limitation [16].

IPDA calculates the probability of track existence at each time scan which is an event with an associated probability [16, 17]. The method is taken from references [16, 17]. The authors of these papers model the track existence either with a two state or a three state Markov chain. In the former case (Markov chain 1) the states denote the existence or non-existence of the track. For the three states Markov chain (Markov chain 2), the states correspond to the cases in which a track exist and observable, exist and unobservable (occluded) and does not exist. In this study Markov chain 1 is used for track initiation.

In the work of [17], in heavy clutter regions, another IPDA based algorithm called IPDA-MAP is explained. This algorithm uses clutter measurement density to evaluate the probability of track existence. In [17], the performance of IPDA-MAP is compared with other track initiation algorithms (PDA, IMM-PDA and IPDA) and it is shown that IPDA-MAP algorithm superior to the other algorithms. Here it will be shown that the performance of IPDA-MAP algorithm can be further increased by the use of map information. To do that, we will create track initiation probability map which shows the possibility of first time appearance of the target, and by using this map; we will get the target probability density information which is used at the initialization part of IPDA-MAP algorithm.

4.1 USE OF TRACK INITIATION PROBABILITY MAP (TIPM) IN TRACK INITIATION

In Section 3 of this work, by using the visibility map, the target position is estimated when the target is in an invisible region. In this section we will use the visibility map in track initiation problem. The track initiation algorithm of [17] uses a parameter named “target probability density” that is generally taken uniform as there is no a prior information about the target probability density. But actually the visibility map information can be used to find this density. As an example, target probability density at impassable regions should be zero as the probability of target existence in this region is zero. So, a new track can not be initiated from this region. On the other hand, track initiation probability at the boundary between the visible region and the invisible region may have high value. This is because of the targets that go into the visible region from the invisible region. The mathematical representation of the above explanations is the “conditional target probability density” which consists of the probability of the existence of a target in a particular cell given that there is one target, which are obtained by using the visibility map. We call this density as the “Track Initiation Probability Map (TIPM)”. The values on this map are conditional probability values of a target to appear at the corresponding locations given that it exists. Below, the process of building the track initiation probability map using the visibility map is explained in detail.

4.1.1 Generation of the Track Initiation Probability Map

A track initiation probability map may be derived from training data. Training data can be obtained by observing several tracks for a given area and for a given radar position. However this way of obtaining track initiation probability map is very expensive and almost impossible. The algorithm that we propose in this work is based on some reasonable thinking.

In this part the derivation of the track initiation probability map from the visibility map is explained. It was mentioned before that the visibility map consists of three regions. In the impassable region, the probability that a target exists is zero. For that reason the probability density can be taken as zero at the points in this region. The probability density at the invisible region can also be taken as zero, since even if a target exists in this region, the target tracker will not be able to take measurements from this target. So, only the visible region is left. The probability density increases as we approach from visible region to the boundary between the visible and the invisible regions. The reasoning behind this approach is the possibility of a target to appear after passing through the invisible region. This probability density can be obtained automatically by the use of a simple algorithm from the visibility map. Here the aim is not to obtain a track initiation probability map consisting of exact values, but only to reveal that this information can be extracted from the visibility map. Also the second aim here is to show that this probability map information can be used at the track initiation algorithm.

For each visible cell investigate 1, 2, 3 and 4 distance neighboring cells. In Figure 4-1 we see an example. The observed cell is marked by “X”, the visible cells are shown by white, the invisible cells are shown by gray and the impassable cells are shown by black. In our example, there are four visible cells, three invisible cells and one impassable cell at the distance 1 neighborhood of the processed cell. Same weight is assigned to the cells at the same distance.

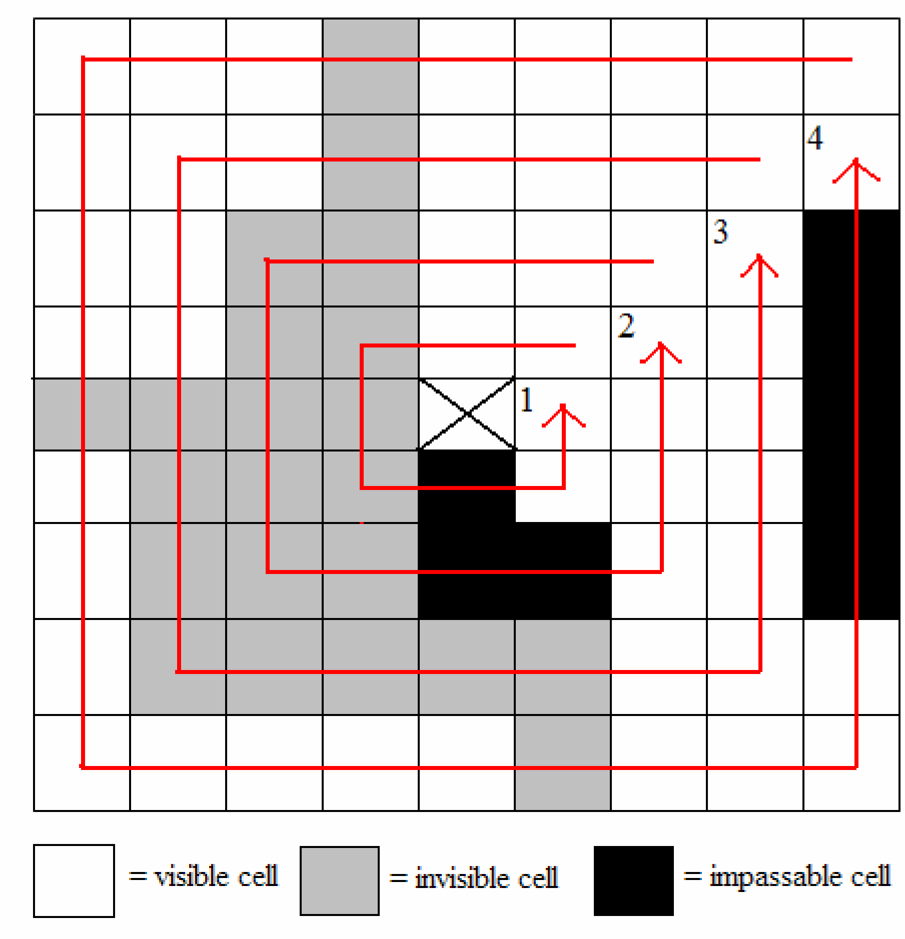

Figure 4-1: Generation of Probability Map Data

During the process we will assign a weight to each cell which is visible. Later these weights will be normalized to obtain a probability mass function. The computation of the weights will be explained on the example given in Figure 4-1 for the cell marked by “X”. A cell will have high weight if its neighbors are invisible otherwise it will have a low weight. The contribution of each neighbor will also depend on its proximity to the Cell X. The contributions of invisible cells are denoted by p_i where i is the neighbouring distance with $p_4 < p_3 < p_2 < p_1$ for a 4 neighbor system.

The contributions of individual cells will be $\frac{p_i}{M_i}$ where M_i is the number of neighbour cells at the i^{th} level, so that $[M_1 \quad M_2 \quad M_3 \quad M_4] = [8 \quad 24 \quad 48 \quad 80]$. To find the total contribution of invisible cells, we multiply the $\frac{p_i}{M_i}$ value by number of invisible cells at the i^{th} level, i.e.

$$P = \sum_{i=1}^{4} \frac{(N_i p_i)}{M_i} \tag{4-1}$$

where N_i is the number of invisible cells at the i^{th} neighbour.

We repeat the same process for impassable cells with different weights of $\{q_i\}_{i=1}^{4}$ where $q_4 < q_3 < q_2 < q_1$ should be satisfied. The total contribution is denoted by Q.

$$Q = \sum_{i=1}^{4} \frac{(R_i q_i)}{M_i} \tag{4-2}$$

where R_i is the number of impassable cells at the i^{th} neighbour. Q is used to reduce the P value to a minimum number. The reasoning behind it is that no track can start from a neighboring cell of an impassable region. However some cells may have invisible neighbors as well as impassable neighbors. So whenever Q value is above a threshold, the weight of the tested cell is set to a minimum value whatever P is.

Example: For the example in Figure 4-1, let $p_1 = 0.4$, $p_2 = 0.3$, $p_3 = 0.2$, $p_4 = 0.1$. Let $q_1 = 0.5$, $q_2 = 0.25$, $q_3 = 0.15$, $q_4 = 0.1$.

$$P = \frac{(3 \times 0.4)}{8} + \frac{(7 \times 0.3)}{24} + \frac{(9 \times 0.2)}{48} + \frac{(3 \times 0.1)}{80} = 0.2788$$

$$Q = \frac{(1 \times 0.5)}{8} + \frac{(2 \times 0.25)}{24} + \frac{(0 \times 0.15)}{48} + \frac{(5 \times 0.1)}{80} = 0.0896$$

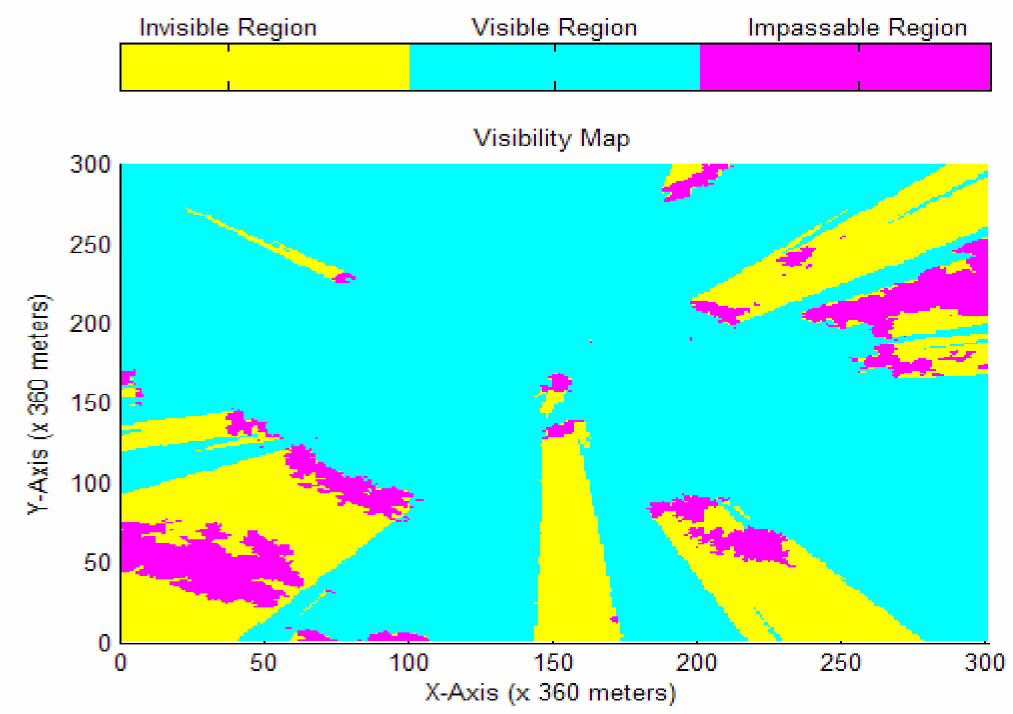

Figure 4-2: Visibility map.

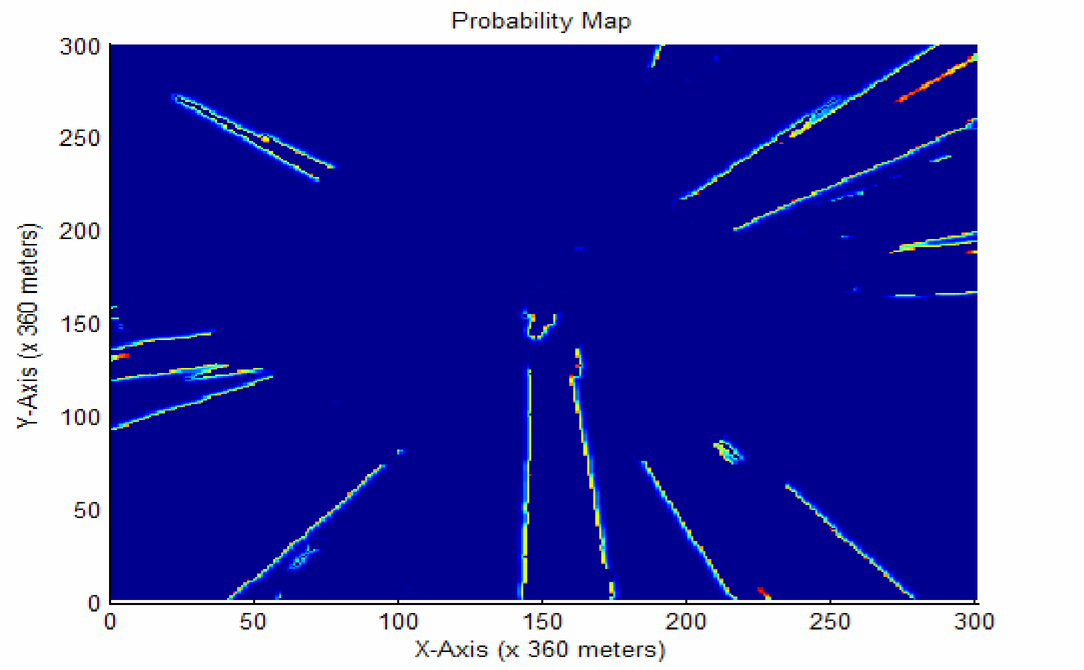

Figure 4-3: Track initiation probability map.

4.2 TRACK INITIATION ALGORITHMS

In this thesis we have used IPDA-MAP algorithm with probability map data for track initiation. First IPDA algorithm will be explained. IPDA-MAP will be discussed later.

4.2.1 Integrated Probabilistic Data Association (IPDA)

Probabilistic data association algorithm (PDA) is a well known algorithm used in tracking applications. PDA works under the assumption that a track exists. For a track initiation problem this assumption is not valid because at this point, we are not sure whether there is a track in the environment or not. In integrated probabilistic data association (IPDA) there is no assumption about the existence of a track. The track existence is modeled as a Markov chain with two states. The states represent track existence and non-existence, and the probability of track existence is evaluated at each iteration. The detailed explanation about the algorithm is given below:

Notation:

m_k = Number of measurements in the track window at time k.

z_k = Set of m_k measurements in the track window at scan k.

Z^k = Set of sets of measurements up to and including scan k.

$\tilde{s}_k \in [0,1]$ is a random variable such that $\tilde{s}_k = 0 \Rightarrow$ there is no target and $\tilde{s}_k = 1 \Rightarrow$ target exists.

$\tilde{t}_k \in [0,1,...,m_k]$ is a random variable such that $\tilde{t}_k = 0 \Rightarrow$ None of the m_k measurements is target originated, $\tilde{t}_k = i \Rightarrow i^{th}$ measurement is target originated.

Track existence is modeled as a Markov process with two states. The state diagram is shown in Figure 4-4.

Note that the a priori probability that no measurement is originated from target at scan k is:

$$P\{t_k = 0 | Z^{k-1}\} = 1 - P\{s_k = 1 | Z^{k-1}\} P_D P_W \tag{4-3}$$

In a similar way, the a priori probability that track exists and no measurement originated from the target is:

$$P\{s_k = 1, t_k = 0 | Z^{k-1}\} = (1 - P_D P_W) P\{s_k = 1 | Z^{k-1}\} \tag{4-4}$$

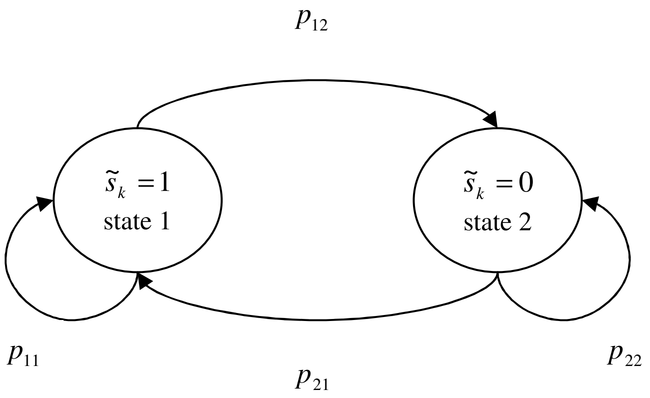

Figure 4-4: State diagram of IPDA for Markov chain 1.

The phases of the algorithm are given below:

Initialization: To initialize the filter the measurements of two consecutive scans are used. When a measurement is taken in the first scan a window is drawn by using the maximum velocity of the target which is the maximum possible velocity for any type of target. And for each validated measurement, i.e. a measurement in the window, in the second scan one new track is formed [16]. The initial value of the probability of track existence is generally taken as constant.

State Prediction: The prediction of the state is calculated using Kalman filter prediction equations.

The probability of track existence $P\{s_k = 1|Z^{k-1}\}$ is found using Markov chain formula as in (4-5).

$$P\{s_k = 1|Z^{k-1}\} = p_{11}P\{s_{k-1} = 1|Z^{k-1}\} + p_{21}P\{s_{k-1} = 0|Z^{k-1}\} \tag{4-5}$$

Note that p_{11} and p_{21} are the state transition probabilities shown in Figure 4-4.

Measurement Prediction: Using a priori probability density of the predicted measurement position $f\left(z|Z^{k-1}\right)$, which is a Gaussian density, the predicted measurement position $\hat{z}_{k|k-1}$ is calculated. A window is defined around $\hat{z}_{k|k-1}$ using window probability P_W.

$$P_W = \int_{V_k} f\left(z \middle| Z^{k-1}\right) dV \tag{4-6}$$

After finding the volume of the window V_k, the validated measurements can be found.

Probability Calculation of the Validated Measurements: Depending on the hypothesis about track existence, the likelihood ratio for the measurements is found. Let L_k be the likelihood ratio for the data received at scan k, then

$$L_k \overset{\Delta}{=} 1 - \delta_k \overset{\Delta}{=} \frac{P\left\{z_k \middle| s_k = 1\right\}}{P\left\{z_k \middle| s_k = 0\right\}} \tag{4-7}$$

$$\delta_k = \begin{cases} P_D P_W & , m_k = 0 \\ P_D P_W - P_D P_W \dfrac{V_k}{\hat{m}_k} \sum_{i=1}^{m_k} p\left\{z_{k,i} \middle| Z^{k-1}\right\} & , m_k > 0 \end{cases} \tag{4-8}$$

Here $\hat{m}_k$ denotes the expected number of false measurements in z_k and P_D is the probability of detection. Generally the number of false measurements has a Poisson distribution with parameter $\lambda = \dfrac{\hat{m}_k}{V_k}$. The derivation of Equation 4-8 can be found in Reference 16.

The coefficients of IPDA algorithm ($\beta_{k,i}$) i.e. the conditional probabilities that the measurement i is originated from target at scan k, provided that the track exists, is given below.

$$\beta_{k,0} = \frac{1 - P_D P_W}{1 - \delta_k} \tag{4-9}$$

$$\beta_{k,i} = \frac{P_D P_W \dfrac{V_k}{\hat{m}_k} p\left\{z_{k,i} \middle| Z^{k-1}\right\}}{1 - \delta_k} \quad , i > 0 \tag{4-10}$$

Note that $\beta_{k,0}$ is the probability that none of the measurements are target originated.

State Update: The state update equations are similar with the state update formulas of probabilistic data association (PDA). Let $\hat{x}_{k|k,i}$ and $\hat{P}_{k|k,i}$ be the state estimate and the error covariance matrix of the state at scan k, assuming that the measurement $z_{k,i}$ originated from target, the updated state and its covariance matrix can be found using (4-11) and (4-12).

$$\hat{x}_{k|k} = \sum_{i=0}^{m_k} \beta_{k,i} \hat{x}_{k|k,i} \tag{4-11}$$

$$\hat{P}_{k|k} = \sum_{i=0}^{m_k} \beta_{k,i} \left(\hat{P}_{k|k,i} + \left(\hat{x}_{k|k,i} - \hat{x}_{k|k} \right) \left(\hat{x}_{k|k,i} - \hat{x}_{k|k} \right)^T \right) \tag{4-12}$$

The last step in the calculations is finding the probability of track existence using the validated measurements at scan k. (4-13) is the update equation of probability of track existence.

$$P\left\{s_k = 1 \middle| Z^k \right\} = \frac{(1-\delta_k)}{1-\delta_k P\left\{s_k = 1 \middle| Z^{k-1}\right\}} P\left\{s_k = 1 \middle| Z^{k-1}\right\} \tag{4-13}$$

Equation 4-13 can be derived from the likelihood ratio equation (4-7) and using Bayes' Theorem several times. The derivation is given below:

$$\begin{aligned} P\left\{s_k = 1 \middle| Z^k\right\} &= P\left\{s_k = 1 \middle| z_k, Z^{k-1}\right\} = \frac{P\left\{s_k = 1, z_k \middle| Z^{k-1}\right\}}{P\left\{z_k \middle| Z^{k-1}\right\}} \\ &= \frac{P\left\{z_k \middle| s_k = 1, Z^{k-1}\right\} P\left\{s_k = 1 \middle| Z^{k-1}\right\}}{P\left\{z_k \middle| Z^{k-1}\right\}} = \frac{P\left\{z_k \middle| s_k = 1\right\} P\left\{s_k = 1 \middle| Z^{k-1}\right\}}{P\left\{z_k \middle| Z^{k-1}\right\}} \end{aligned} \tag{4-14}$$

Then using (4-7), (4-14) becomes:

$$P\left\{s_k = 1 \middle| Z^k\right\} = \frac{(1-\delta_k) P\left\{z_k \middle| s_k = 0\right\} P\left\{s_k = 1 \middle| Z^{k-1}\right\}}{P\left\{z_k \middle| Z^{k-1}\right\}} = \frac{(1-\delta_k) P\left\{s_k = 1 \middle| Z^{k-1}\right\}}{\dfrac{P\left\{z_k \middle| Z^{k-1}\right\}}{P\left\{z_k \middle| s_k = 0\right\}}} \tag{4-15}$$

For the denominator of (4-15):

$$\begin{aligned}\frac{P\{z_k|Z^{k-1}\}}{P\{z_k|s_k=0\}} &= \frac{P\{z_k, s_k=1|Z^{k-1}\}}{P\{z_k|s_k=0\}} + \frac{P\{z_k, s_k=0|Z^{k-1}\}}{P\{z_k|s_k=0\}} \\ &= \frac{P\{z_k|s_k=1, Z^{k-1}\}P\{s_k=1|Z^{k-1}\}}{P\{z_k|s_k=0\}} + \frac{P\{z_k|s_k=0, Z^{k-1}\}P\{s_k=0|Z^{k-1}\}}{P\{z_k|s_k=0\}} \\ &= (1-\delta_k)P\{s_k=1|Z^{k-1}\} + P\{s_k=0|Z^{k-1}\} = 1-\delta_k P\{s_k=1|Z^{k-1}\}\end{aligned} \tag{4-16}$$

So the combination of (4-15) and (4-16) gives us Equation 4-13.

For the case of no measurements in the track window at time k, i.e. $m_k=0$:

$$\begin{aligned}P\{s_k=1|Z^k\} &= \frac{P\{z_k|s_k=1\}P\{s_k=1|Z^{k-1}\}}{P\{z_k|Z^{k-1}\}} \\ &= \frac{P\{s_k=1, t_k=0|Z^{k-1}\}}{P\{t_k=0|Z^{k-1}\}} = \frac{(1-P_DP_W)P\{s_k=1|Z^{k-1}\}}{1-P_DP_WP\{s_k=1|Z^{k-1}\}}\end{aligned} \tag{4-17}$$

Equation 4-17 can also be obtained from (4-8) and (4-13) for the case of $m_k=0$. The detailed formulation of IPDA can be found in Reference 16.

To conclude, IPDA is a direct extension of PDA. The main difference is IPDA works without an initial assumption of track existence and it calculates the probability of track existence at each scan. This makes a great advantage for track initiation, confirmation and termination.

The flowchart of the algorithm is given in Figure 4-5.

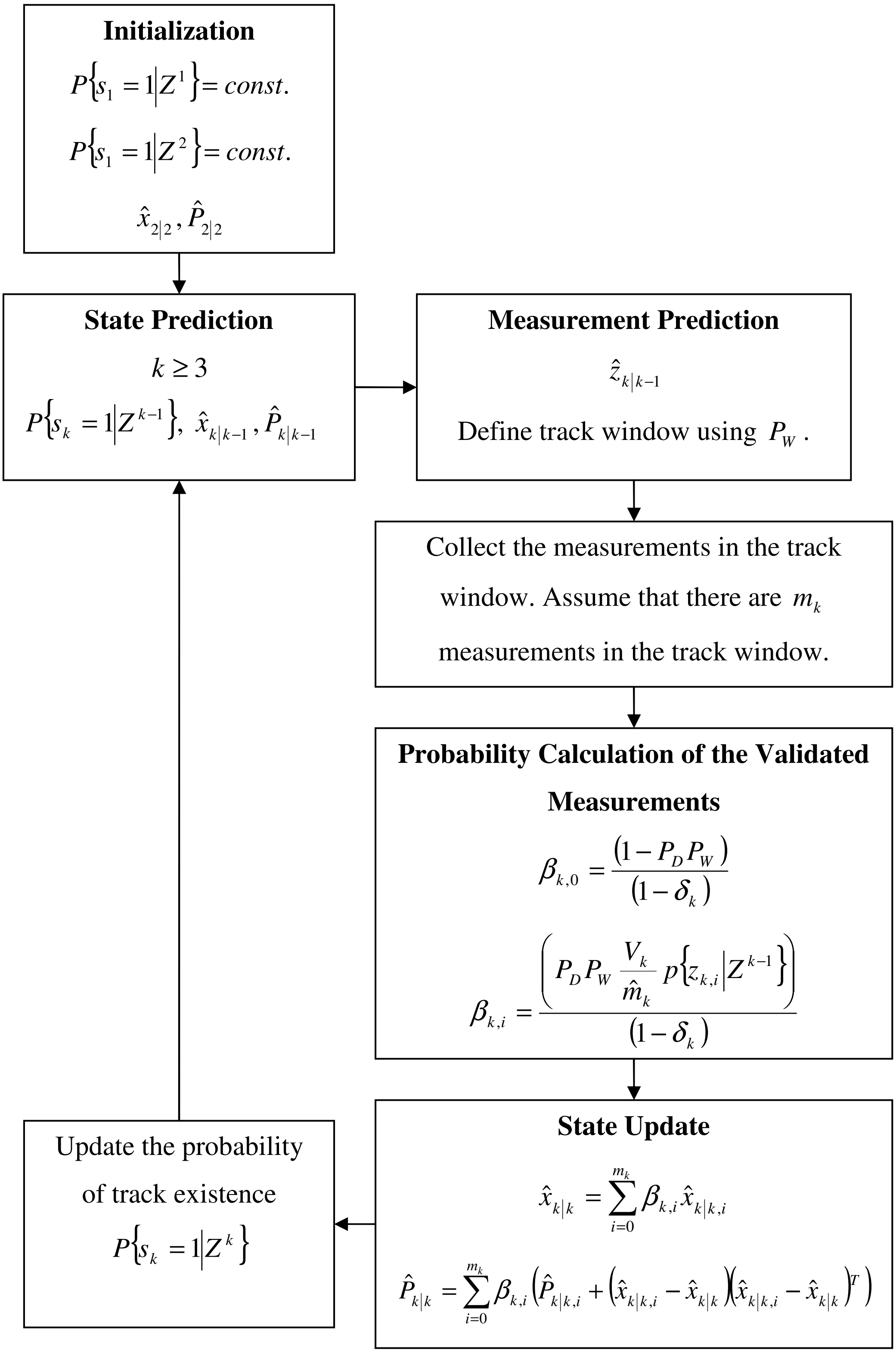
Initialization
$P\{s_1 = 1 | Z^1\} = const.$
$P\{s_1 = 1 | Z^2\} = const.$
$\hat{x}_{2|2}, \hat{P}_{2|2}$
State Prediction
$k \geq 3$
$P\{s_k = 1 | Z^{k-1}\},\ \hat{x}_{k|k-1}, \hat{P}_{k|k-1}$
Measurement Prediction
$\hat{z}_{k|k-1}$
Define track window using P_W .
Collect the measurements in the track window. Assume that there are m_k measurements in the track window.
Probability Calculation of the Validated Measurements
$\beta_{k,0} = \frac{(1 - P_D P_W)}{(1 - \delta_k)}$
$\beta_{k,i} = \frac{\left(P_D P_W \frac{V_k}{\hat{m}_k} p\{z_{k,i} | Z^{k-1}\}\right)}{(1 - \delta_k)}$
State Update
$\hat{x}_{k|k} = \sum_{i=0}^{m_k} \beta_{k,i} \hat{x}_{k|k,i}$
$\hat{P}_{k|k} = \sum_{i=0}^{m_k} \beta_{k,i} \left(\hat{P}_{k|k,i} + (\hat{x}_{k|k,i} - \hat{x}_{k|k})(\hat{x}_{k|k,i} - \hat{x}_{k|k})^T\right)$
Update the probability of track existence
$P\{s_k = 1 | Z^k\}$

Figure 4-5: IPDA algorithm flow chart.

4.2.2 IPDA-MAP Algorithm

This algorithm can be considered as an extension of the IPDA algorithm as it uses clutter map information additionally. It is known that track initiation process becomes harder when the clutter density in the environment is high. If there is prior information about the clutter density of the environment, this information can be used in order to increase the performance of the track initiation algorithm. Just like IPDA, IPDA-MAP algorithm calculates the probability of track existence and supplies probabilistic information about the target existence at each iteration. When this probability value exceeds the confirmation threshold a new track is started. Similarly, when the probability value falls below the termination threshold the track is terminated. The confirmation and termination thresholds are discussed in [18].

IPDA-MAP formulas are identical to IPDA algorithm except the calculation of the δ_k and coefficients $\beta_{k,i}$. For IPDA-MAP,

$$\delta_k = \begin{cases} P_D P_W & ,m_k = 0 \\ P_D P_W - P_D P_W \sum_{i=1}^{m_k} \dfrac{p\{z_{k,i} | Z^{k-1}\}}{\rho_{k,i}^{FA}} & ,m_k > 0 \end{cases} \tag{4-18}$$

$$\beta_{k,i} = \frac{P_D P_W \dfrac{p\{z_{k,i} | Z^{k-1}\}}{\rho_{k,i}^{FA}}}{1-\delta_k} \quad ,i > 0 \tag{4-19}$$

In equations (4-18) and (4-19) $\rho_{k,i}^{FA} = \rho_k^{FA}(z_{k,i})$ is the averaged clutter measurement density at $z_{k,i}$ on scan k. This density is the output of the clutter map.

The above equations are taken from the derivation of IPDA-MAP algorithm. The track initiation probability map will be used at the initialization part of this algorithm.

For IPDA-MAP the initial target existence probability is found using Equation 4-20.

$$P\{s_1 = 1 | Z^1\} = \frac{\rho^t}{\rho^t + \rho_{1,i}^{FA}} \tag{4-20}$$

In Equation 4-20, ρ^t is the target probability density and is assumed to be uniform if there is no information [17].

$$\rho^t = \frac{1}{V_2} \tag{4-21}$$

Here V_2 is the volume of the gate at second scan. We have already found the track initiation probability map in Section 4.1. Instead of using (4-21) we can use the output of track initiation probability map as the target probability density. In this study, we analyze how the performance is affected by using TIPM in IPDA-MAP algorithm, for which we assumed that clutter density is constant and uniformly distributed.

CHAPTER 5

ANALYSIS AND SIMULATIONS

This chapter includes the implementation of the proposed algorithms introduced in Chapter 3 and Chapter 4. The results are discussed and the performance comparison is also done in this part.

Simulations are divided into two main groups. In Section 5.1 the implementations of the algorithms given in Chapter 3, which are about tracking in invisible region, are accomplished. These are related with track continuity. In Section 5.2, the simulations of the algorithm discussed in Chapter 4, for track initiation using visibility map, are demonstrated.

5.1 TRACKING USING ELEVATION MODEL OF THE TERRAIN

This part includes the simulations of using visibility map information in tracking algorithms. The performance comparison is done between the algorithms with and without using visibility map. When a target enters into an invisible region, no measurement update can be applied to its state if the visibility map is not used. This causes an increase in the gate that the target is looked for. When the gate reaches to a certain value track loss is declared and the search is stopped. For the visibility map users, one of the two algorithms explained in Chapter 3 is used. For this case gate increase is much smaller.

Tracks are generated one at a time and tracked by the Kalman tracker. There is no clutter or false alarm during the simulations. Furthermore, tracker uses the correct parameters, i.e. the true values of the measurement and process noise covariances. For both cases, i.e. a simple Kalman and the one that uses map information, tracking is an easy problem because of the assumptions stated above, but by this way we believe that we have highlighted the improvement of using the map. The tracks are generated by using manually selected initial conditions so that the tracks are passed with high probability through the invisible region. After the generation of the tracks, the ones that pass through the impassable region or not entering the invisible region are discarded. As a result the comparison is basically done for the tracks that suffer from invisibility.

5.1.1 Target Motion Model and Measurements

Performance analysis and comparison is done by Monte Carlo simulations. Constant velocity model is used as a motion model in 2-D Cartesian coordinates. The motion is generated by using the model given in Equation 5-1.

$$X_{k+1} = AX_k + GW_k \tag{5-1}$$

Here $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $G = \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix}$, $X_k = \begin{bmatrix} x_k \\ V_{x,k} \\ y_k \\ V_{y,k} \end{bmatrix}$, $W_k = \begin{bmatrix} w_{x,k} \\ w_{y,k} \end{bmatrix}$.

The state vector X_k represents the coordinates and velocities of the target at x and y directions. $w_{x,k}$ and $w_{y,k}$ are zero mean, Gaussian process noises. Note that the true trajectory corresponds to the components (x_k, y_k) of the state vector X_k.

After the true trajectories have been formed, the paths which pass through the impassable region are excluded. The measurements are generated from the remaining paths by using Equation 5-2.

$$Y_k = CX_k + HV_k \tag{5-2}$$

Here $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $Y_k = \begin{bmatrix} y_{x,k} \\ y_{y,k} \end{bmatrix}$, $V_k = \begin{bmatrix} v_{x,k} \\ v_{y,k} \end{bmatrix}$.

The measurement vector Y_k includes the coordinates of the measurements at x and y directions. $v_{x,k}$ and $v_{y,k}$ are zero mean, Gaussian measurement noises.

In the simulations, tracking is terminated whenever the trace of the covariance matrix of the measurements exceeds the covariance matrix threshold. In the figures related with tracking performance, x-axis shows the covariance matrix threshold values while y-axis shows the corresponding number of survived tracks. Without visibility map information, when the target becomes invisible no measurement update is applied. On the other hand when the visibility map information is used, the algorithm tries to make the covariance matrices smaller by the use of the methods given in Chapter 3. For each simulation "generated trajectories", "simulation parameters table" and "performance comparison graph" are demonstrated.

Six Monte Carlo simulations are done. Two different maps are used in these six simulations; three simulations for each map. During simulations the following cases are considered. Low process noise-low measurement noise, low process noise-high measurement noise, high process noise-low measurement noise and high process noise-high measurement noise.

5.1.2 Simulation Results

The parameters that are common to all simulations about tracking performance are given in Table 5-1.

Table 5-1: Common parameters for the tracking simulations.

Parameters Related with Visibility Map Information	
Used SRTM Data	N39E033.hgt
Target Height (meter)	1450
Filter Parameter	
Gate Probability	0.989
$W^{(0)}$ (used in UKF)	0.90

The covariance matrix threshold values are normalized relative to the map. For example a threshold value of 30 means that the trace of the matrix is 30. In this case:

$$\sigma_x^2 + \sigma_y^2 = 30 \tag{5-3}$$

If the variances along x and y axes are assumed to be equal:

$$\sigma_x^2 + \sigma_y^2 = 15 \Rightarrow \sigma_x = \sigma_y = 3.87 \tag{5-4}$$

In this case a standard deviation of 3.87 corresponds to $3.87 \times 360 \cong 1390$ meters, since the resolution of the map is $360x360$ meters.

The standard deviations in meters that correspond to matrix trace values are given in Table 5-2. This table is constructed by assuming $\sigma_x = \sigma_y$.

Note that for practical applications, selection of standard deviations threshold around 1000-1500 meters is reasonable to decide on the confirmation of a track. Any standard deviation beyond 1500 meters generates a gate which covers a large area so that even if the track continuous, initiation of a new track instead of looking for the old one becomes more feasible. Selection of the threshold smaller than 1000 meters causes a track loss for very moderately behaving targets. So although the following results about the analysis of the system cover threshold values in between 600-2000 meters, conclusion is based on the performance of the tracker in the range 1000-1500 meters.

Table 5-2: Actual values of the normalized thresholds.

Normalized Covariance Threshold	**Actual Standard Deviation (meter)**		**Normalized Covariance Threshold**	**Actual Standard Deviation (meter)**
5	569.2		35	1505.9
10	804.9		40	1609.9
15	985.9		45	1707.6
20	1138.4		50	1800.0
25	1272.8		55	1887.8
30	1394.3		60	1971.8

5.1.2.1 Simulation 1 (S1)

Simulation 1 uses the first map as shown in Figure 5-1. In this scenario process and measurement noises are kept small.

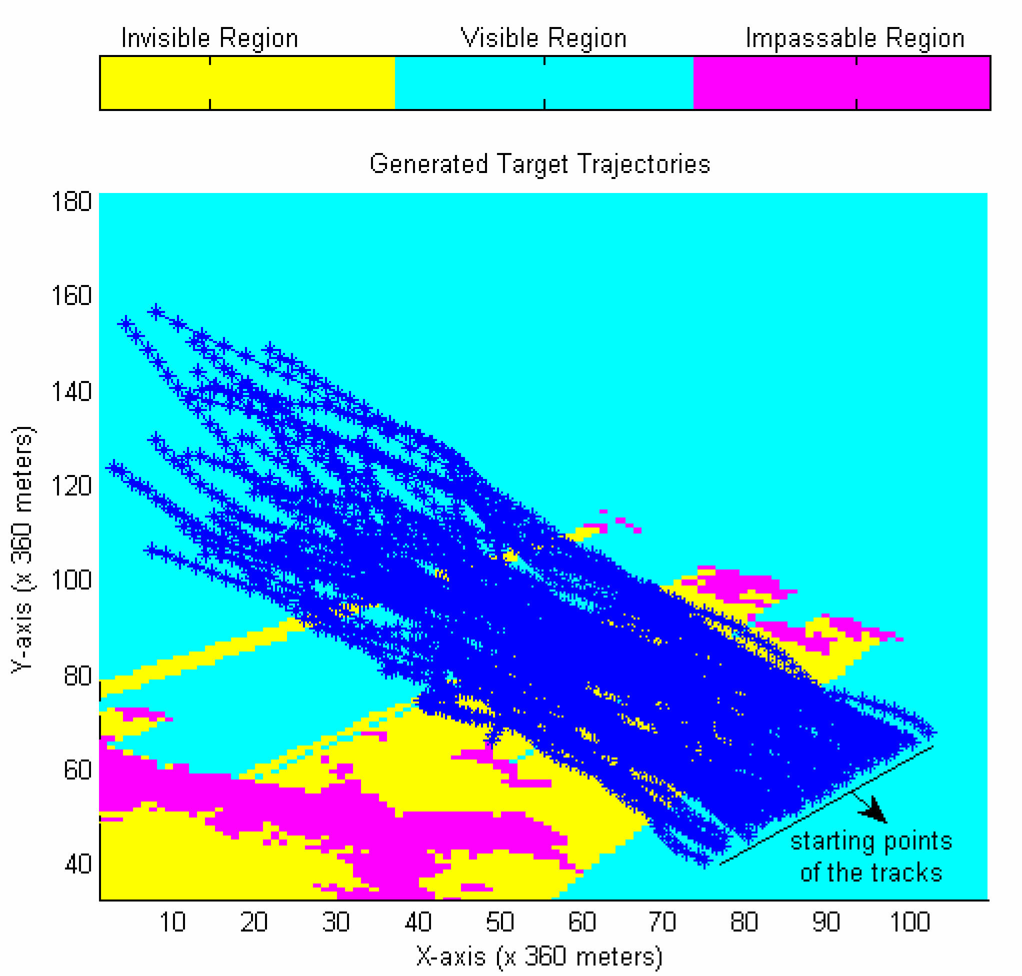

Figure 5-1: Target trajectories for Simulation 1.

Table 5-3: Parameters for Simulation 1.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 10$ Azimuth (rad): Gaussian, $\sigma = 0.0017$
Number of Tracks	62

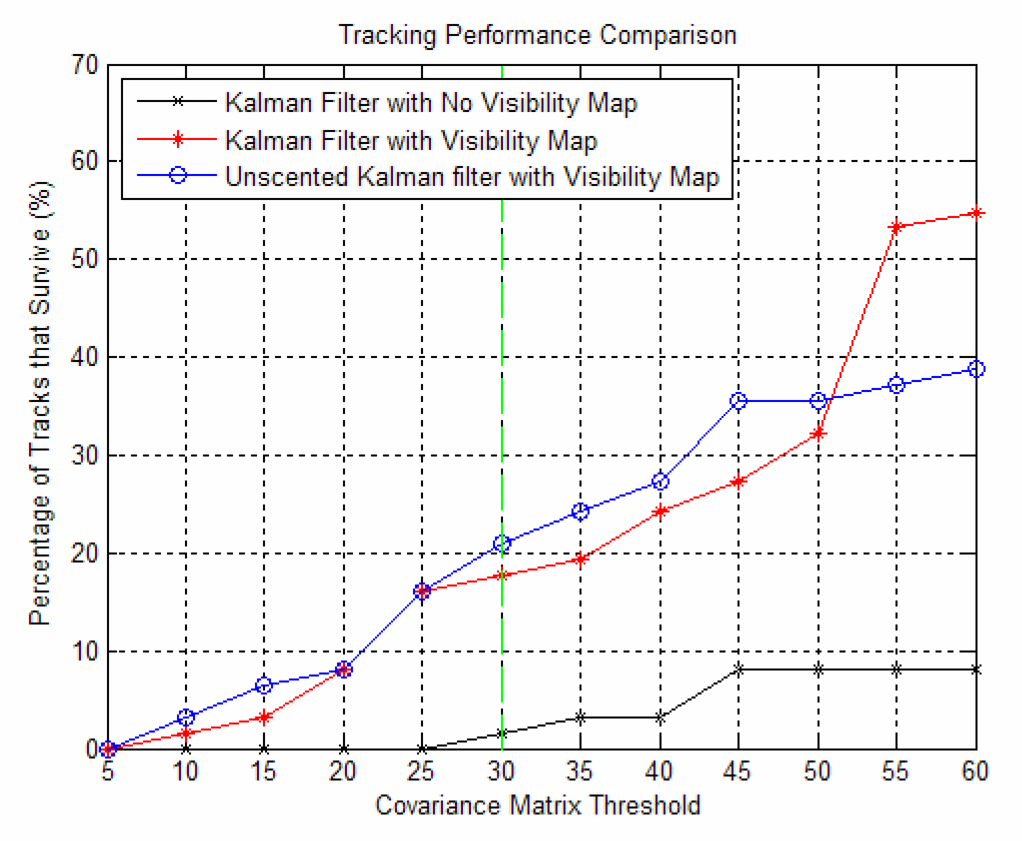

Figure 5-2: Tracking performance for Simulation 1.

5.1.2.2 Simulation 2 (S2)

The same map as S1 is used. Process noise is same as S1, however measurement noise is increased.

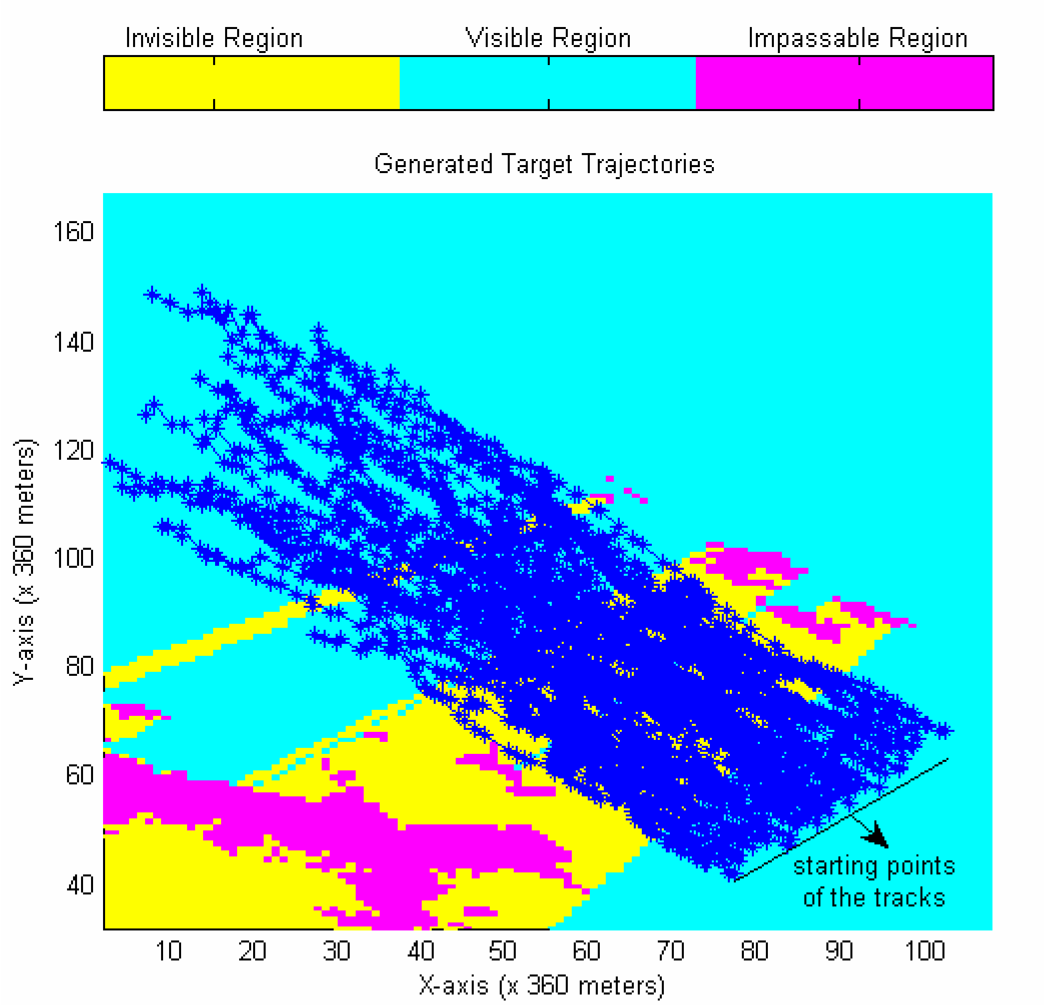

Figure 5-3: Target trajectories for Simulation 2.

Table 5-4: Parameters for Simulation 2.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0087$
Number of Tracks	60

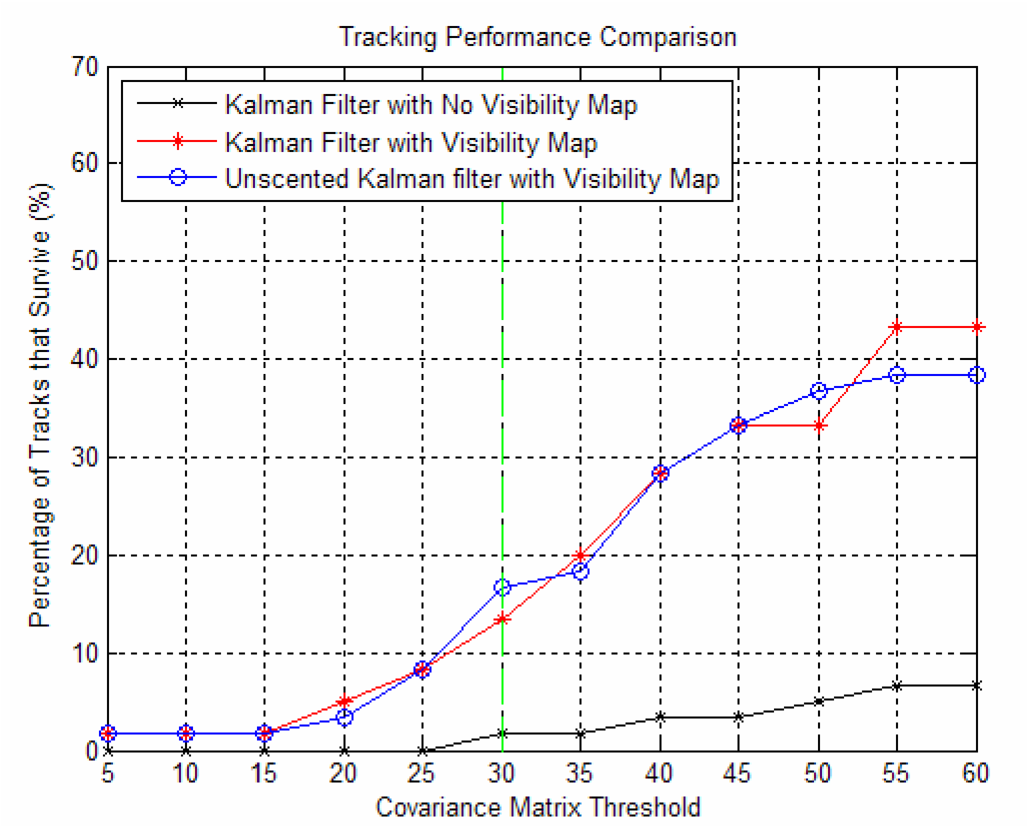

Figure 5-4: Tracking performance for Simulation 2.

5.1.2.3 Simulation 3 (S3)

Map is not changed in this simulation. Process noise is increased that gives much nonlinear trajectories compared to the first two simulations. Measurement noise is also high and is same as S2.

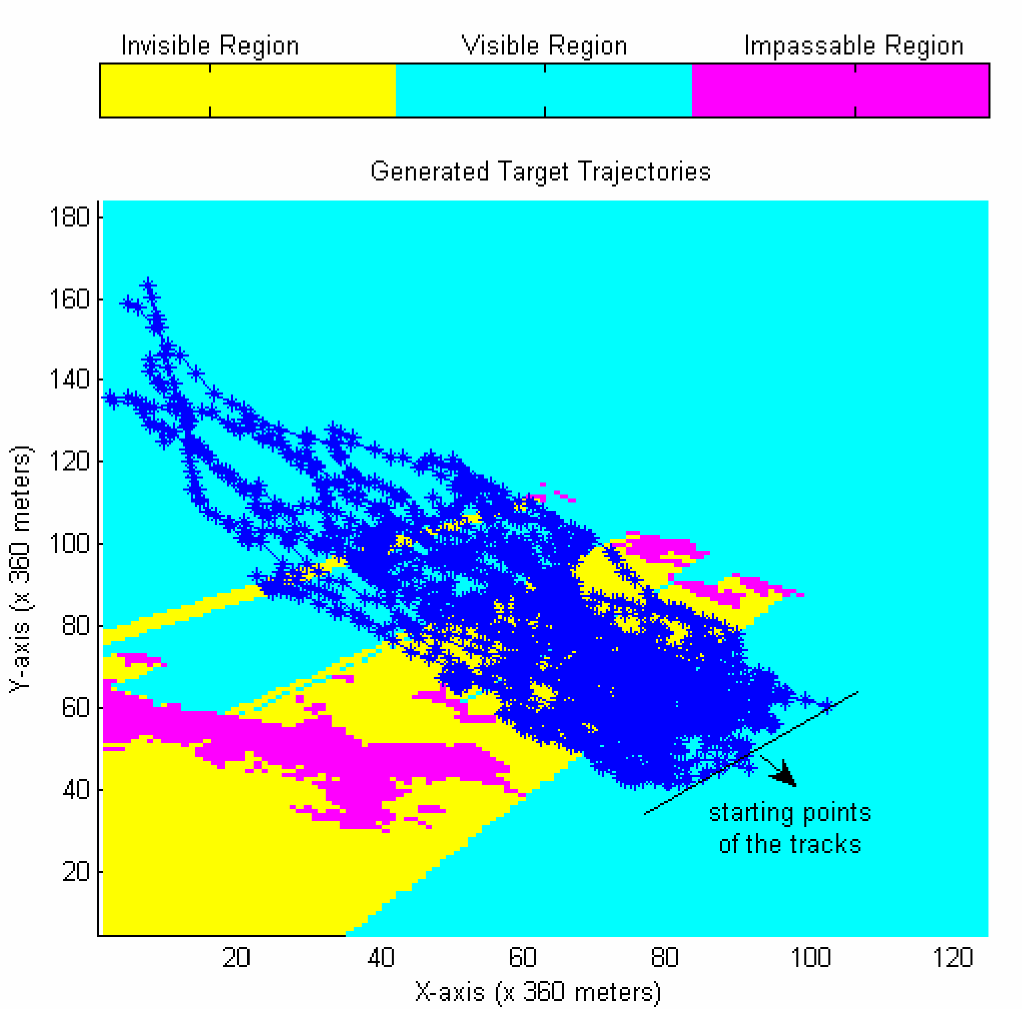

Figure 5-5: Target trajectories for Simulation 3.

Table 5-5: Parameters for Simulation 3.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 113.8, \sigma_y = 113.8$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0087$
Number of Tracks	41

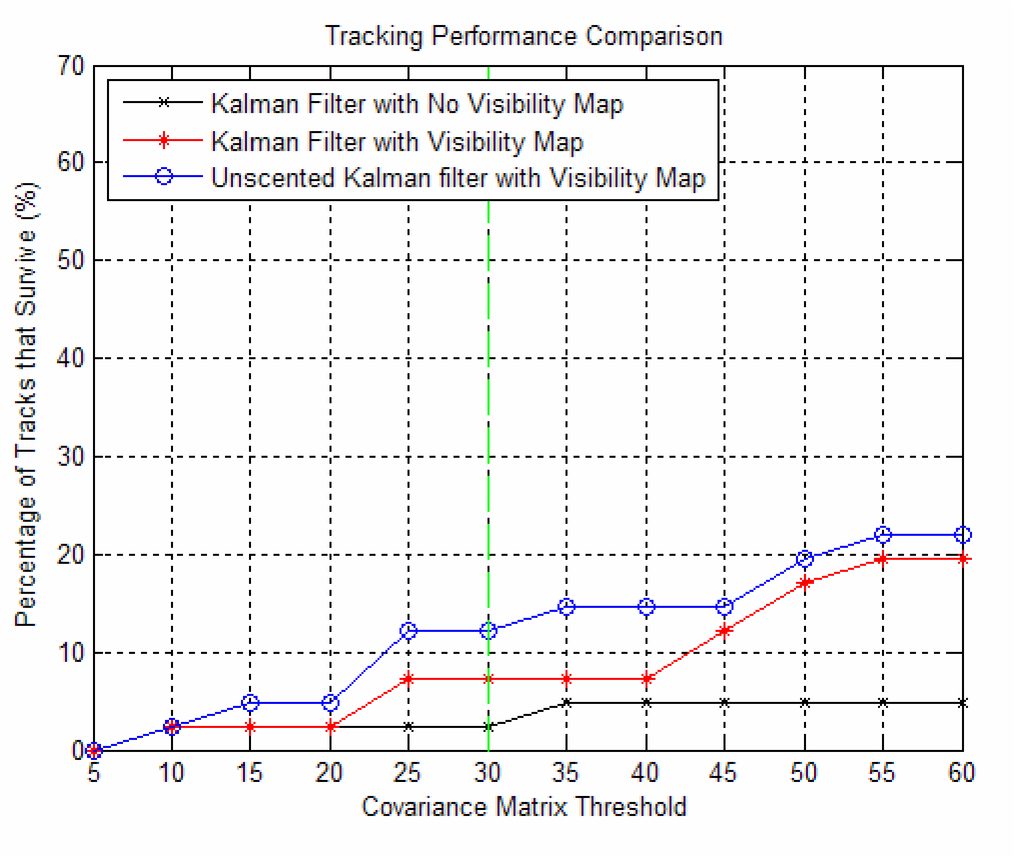

Figure 5-6: Tracking performance for Simulation 3.

5.1.2.4 Simulation 4 (S4)

Map is changed. The invisible region is smaller compared to the previous map. With this map first simulation uses small process and measurement noises.

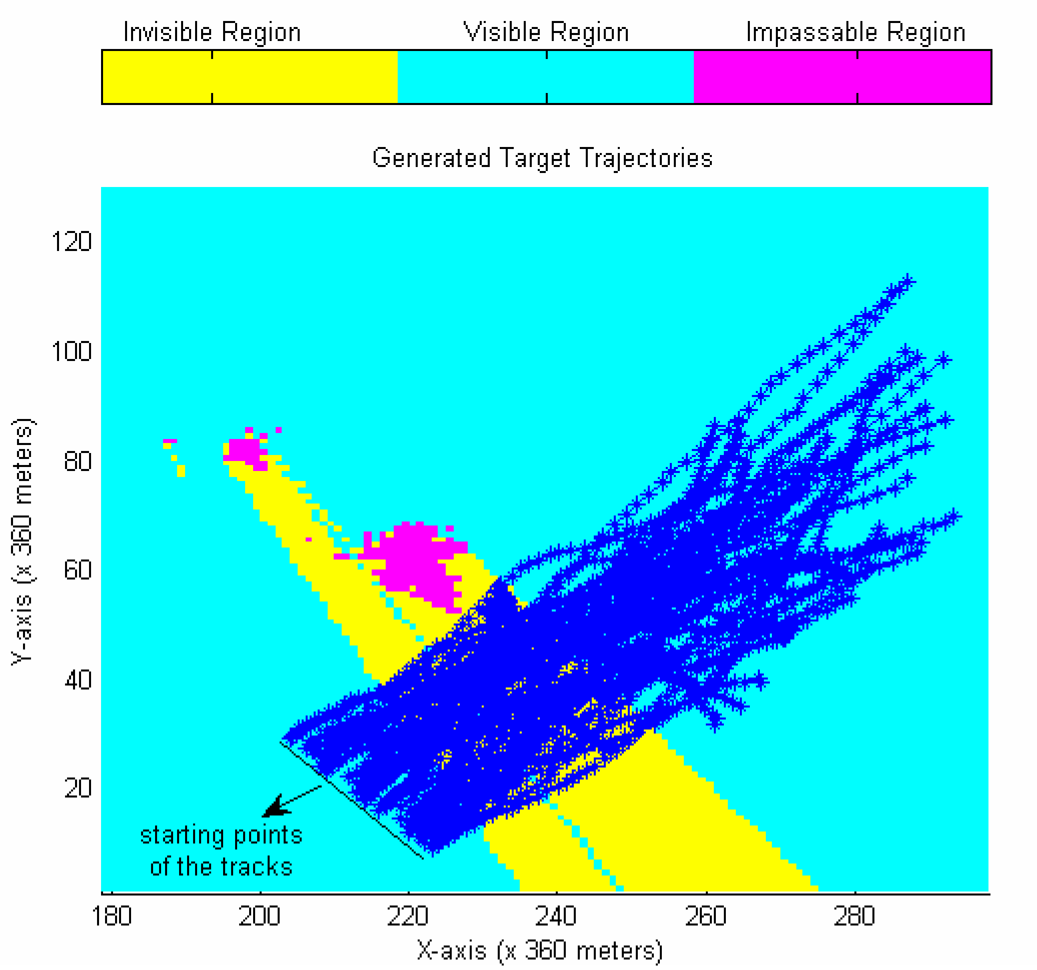

Figure 5-7: Target trajectories for Simulation 4.

Table 5-6: Parameters for Simulation 4.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0017$
Number of Tracks	49

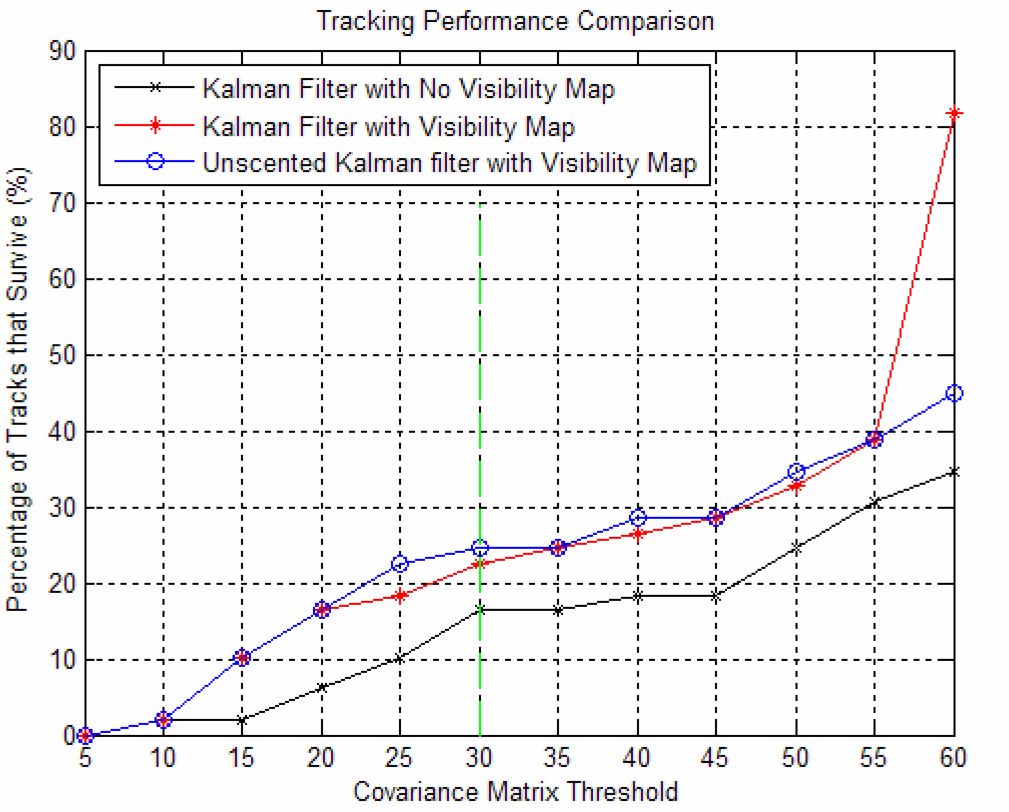

Figure 5-8: Tracking performance for Simulation 4.

5.1.2.5 Simulation 5 (S5)

Second map with low process and high measurement noises are used in this simulation.

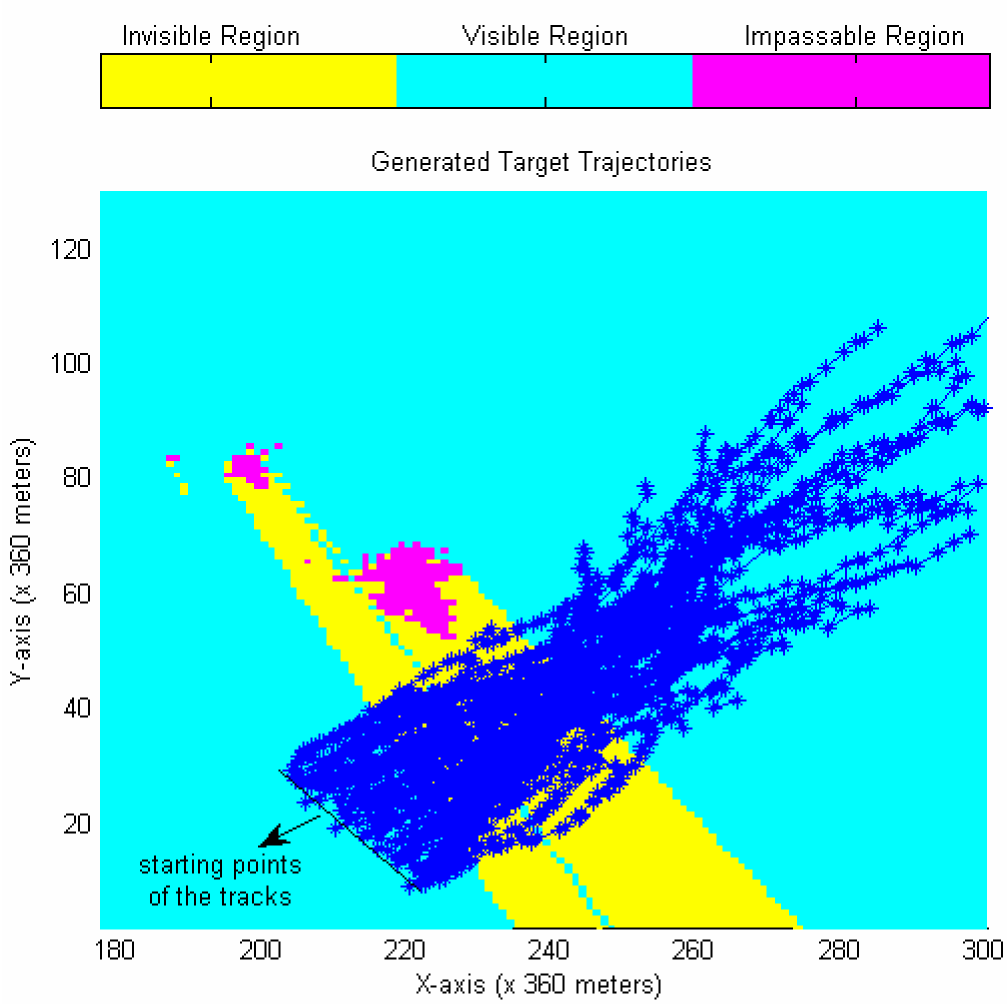

Figure 5-9: Target trajectories for Simulation 5.

Table 5-7: Parameters for Simulation 5.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 10$ Azimuth (rad): Gaussian, $\sigma = 0.0087$
Number of Tracks	46

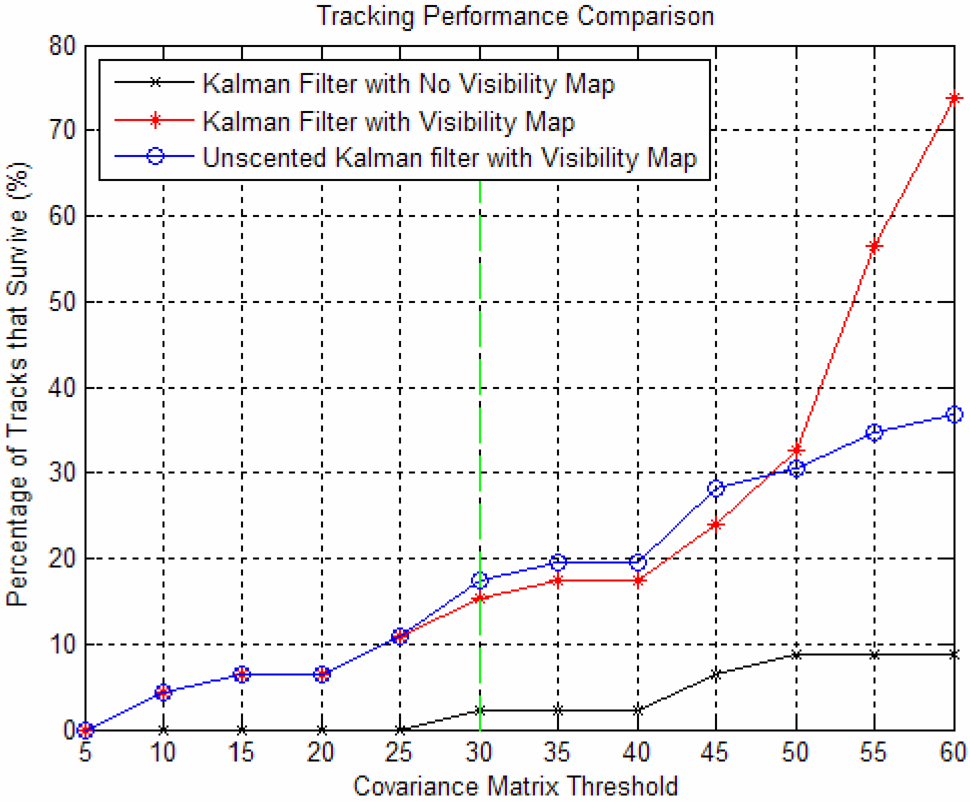

Figure 5-10: Tracking performance for Simulation 5.

5.1.2.6 Simulation 6 (S6)

This simulation uses second map with high process and low measurement noises.

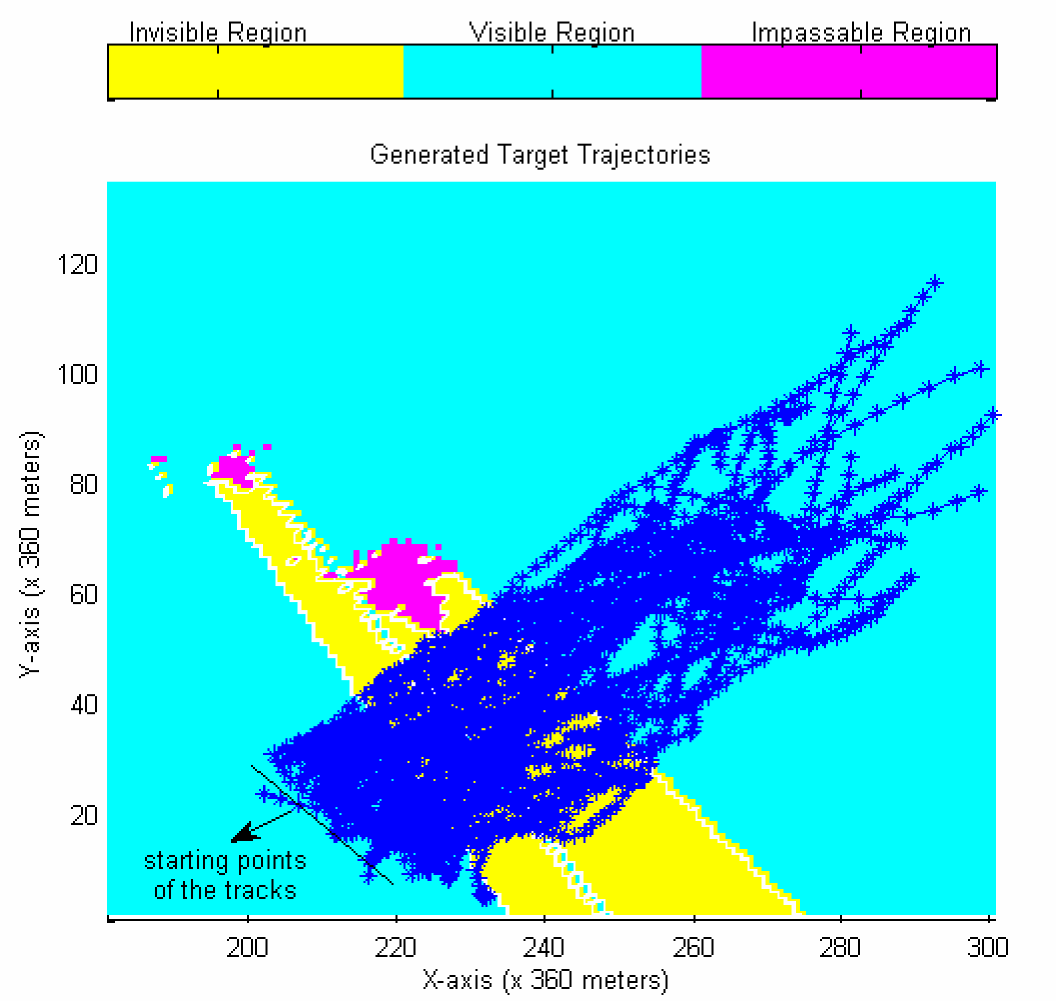

Figure 5-11: Target trajectories for Simulation 6.

Table 5-8: Parameters for Simulation 6.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 113.8, \sigma_y = 113.8$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 10$ Azimuth (rad): Gaussian, $\sigma = 0.0017$
Number of Tracks	59

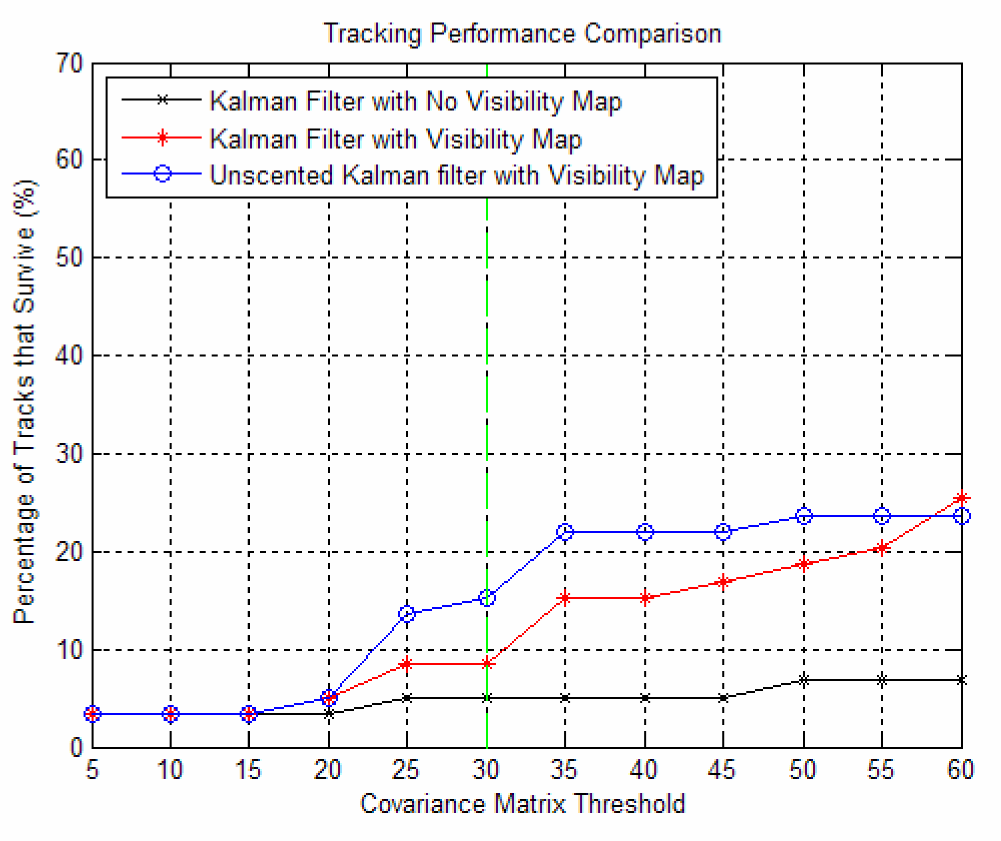

Figure 5-12: Tracking performance for Simulation 6.

5.1.3 Remarks about the Simulations

We first note that the obvious result that as the threshold increases, the number of survived tracks increases in all algorithms. But because of the unrealistic widths of the gates, the comparisons at very high threshold values are not meaningful. A gate width around 1400 meters which corresponds to a threshold value in the vicinity of normalized threshold of 30 units will be considered. For that reason the comparisons are done around this threshold value.

For Simulation 1 (S1), there exist a process noise of 51 meters standard deviation and measurement errors with standard deviation of 10 meters in range and 0.0017 radians in azimuth. At the comparison range, around 20 percent of all tracks are survived with visibility map information. On the other hand without using the visibility map around 4 percent of the tracks could be survived. The performances of Unscented Kalman filter and standard Kalman filter are not distinguishable.

In S2, both the range and the azimuth errors in the measurements are increased. In this case only 16 percent of the tracks survive if map is used. This percentage decreases to 2 if map is not used. Note that surviving tracks are smaller in percentage compared to S1 as expected.

In S3, both the process noise and the measurement noise are increased. In this case both of the algorithms give poorer results. High process noise degrades the algorithm performance. This is because, in high process noise, the nonlinearity in the generated trajectories increases.

S4, S5 and S6 use a different map for which the invisible region is smaller. Because of this reason an increase on the performance for all cases is expected. Furthermore, map information may become more unvaluable since a track may pass through the invisible region without a large increase in its covariance.

In S4, the range error in measurements is increased. In this case without the map information a survival performance around 17 percent is obtained while with the additional map information, the detection performance is around 22 percent.

In S5, a measurement error with a low range noise and a high azimuth noise is applied. It can be seen that by the use of visibility map better detection results are obtained.

In S6, process noise is increased and similar results are obtained.

Note that for UKF, $W^{(0)}$ is a free parameter which is the weight on mean. In our simulations we take it as 0.90. The smaller $W^{(0)}$, the narrower the gate width.

As it is mentioned before, for the simulations 4 to 6, the width of invisible region is smaller than the invisible region at simulations 1 to 3. The simulation results reveal that, there is no considerable difference between the algorithms when the invisible region width is small and there is not much noise in the environment (S4). On the other hand, for noisy environments even though the invisible region width is small, visibility map usage gives high detection performance (S5 and S6).

5.2 TRACK INITIATION

In this part, the use of track initiation probability map in IPDA-MAP algorithm presented in [17] is used. The performance comparison is made for true track confirmation and false track rejection.

Performance evaluation is done on simulated data. A drawback of the evaluation is that the track generation and evaluation algorithms use the same track initialization probability map. In order to evaluate the algorithm given in [17] with simulated data, initial points of the tracks must be sampled from TIPM. We will first give a very basic algorithm of sampling a given distribution. Then this subsection continues with generation of tracks and their initiation.

5.2.1 Sampling from a Probability Distribution: Discrete Rejection Method

This method is used to generate random numbers by using a probability mass function. The algorithm given in Reference 19 to draw a sample from a probability mass function $P[k,n]$ is explained below:

a. Choose a density function $Q[k,n]$ and a constant c such that $P[k=i,n=j] \leq cQ[k=i,n=j]$ for all i and j.

b. Generate a random number u from a uniform density $U[0,1]$.

c. Select a point from the density function Q i.e. $Q[k=v,n=y]$.

d. Let $r = \frac{cQ[k=v,n=y]}{P[k=v,n=y]}$. If $ru \leq 1$, accept and return v and y.

e. Otherwise, reject v and y and go to "b".

Note that $Q[.,.]$ is a density function that we are able to sample. In the simulations $Q[.,.]$ is taken as uniformly distributed.

5.2.2 Generation of the Input Data

Evaluation uses two types of tests, true track confirmation test and false track rejection test.

For true track confirmation test, firstly, track initiation probability map is generated as in Section 4.1.1. Using TIPM and discrete rejection method initial coordinates of the target trajectories are obtained. Then using target motion and measurement models at Section 5.1.1, true target trajectories are formed. The trajectories that pass through the invisible and the impassable regions are excluded from the true target trajectories. The remaining trajectories are used in the algorithm. A sample data generation process is given below.

a. Generate track initiation probability map. In Figure 5-13 an example of a TIPM is shown. Yellow and red cells correspond to high probability regions while blue cells represent low probabilities.

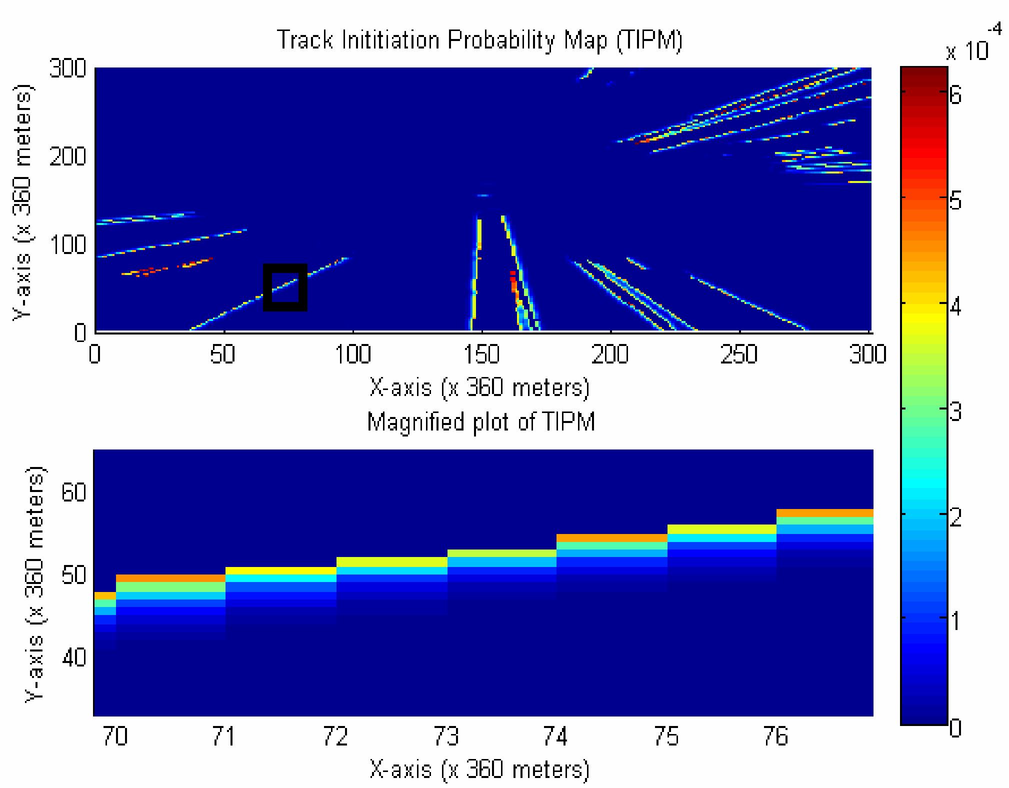

Figure 5-13: Track initiation probability map example.

b. Use TIPM as a probability mass function in discrete rejection method to generate initial coordinates of the target trajectory. An example of possible initial coordinates is given below.

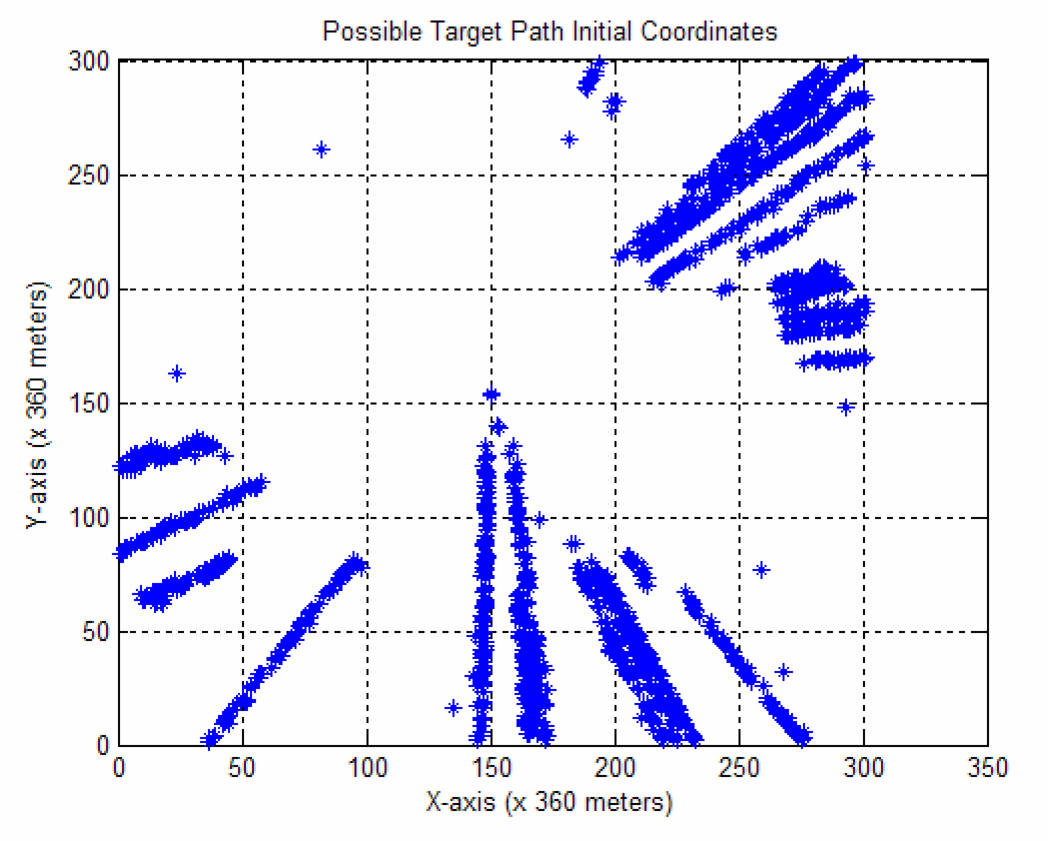

Figure 5-14: Initial coordinates.

c. Use target motion and measurement model at 5.1.1, and generate target trajectories by using initial coordinates found in (b). An example of all target trajectories is given in Figure 5-15.

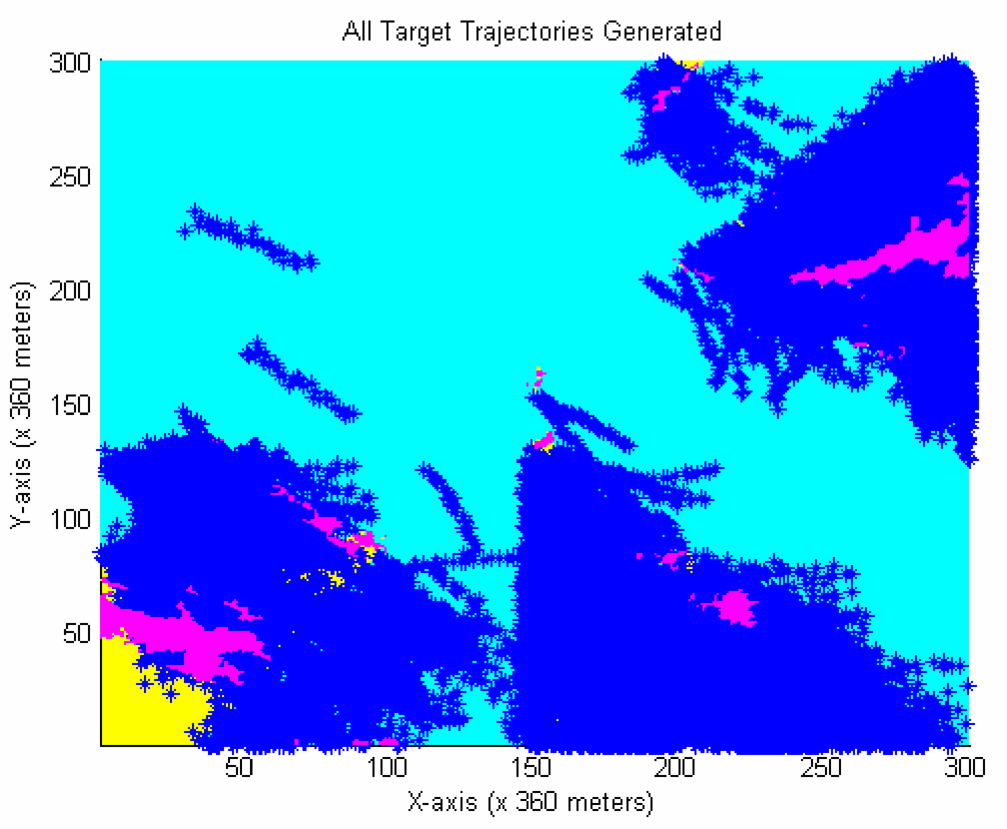

Figure 5-15: Generated target trajectories.

d. Remove the trajectories that pass through the impassable region and also remove the trajectories that pass through invisible region only in the first four steps. The remaining trajectories which are shown below can be used as inputs to the algorithm.

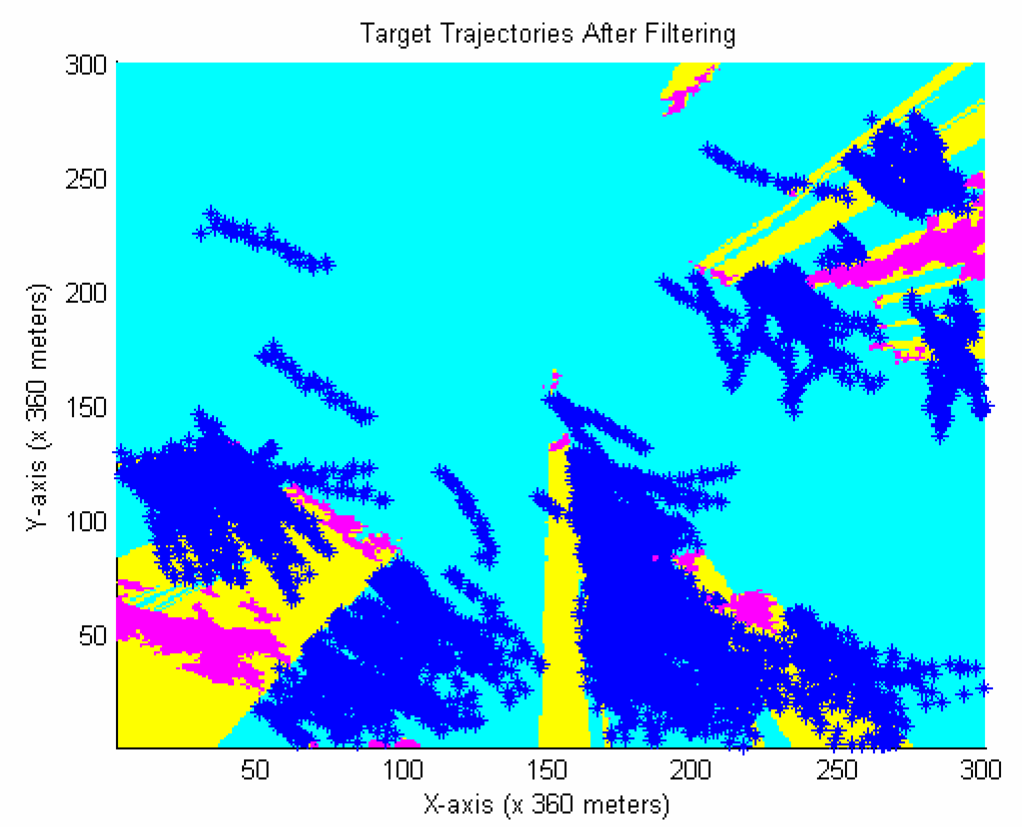

Figure 5-16: Target trajectories after filtering.

For false track detection test, a constant false alarm rate is used and it is assumed that false detections are uniformly distributed. For a constant false alarm rate and a cell size, false target trajectories are generated by sampling from a uniform distribution. The number of false detections is assumed to be Poisson distributed. The mean value of this distribution can be found by multiplying probability of false alarm with the number of cells. A sample false track generation process is given below:

a. Find the mean value of the number of false detections (m_F) by multiplying probability of false alarm with the number of cells. Then take a sample from Poisson distribution with mean m_F. This sample gives us the number of false detections at this scan.

b. Using the sample found in (a), generate random false measurements from a uniform distribution. As an example, false measurements that are detected in a scan can be seen in Figure-5-17.

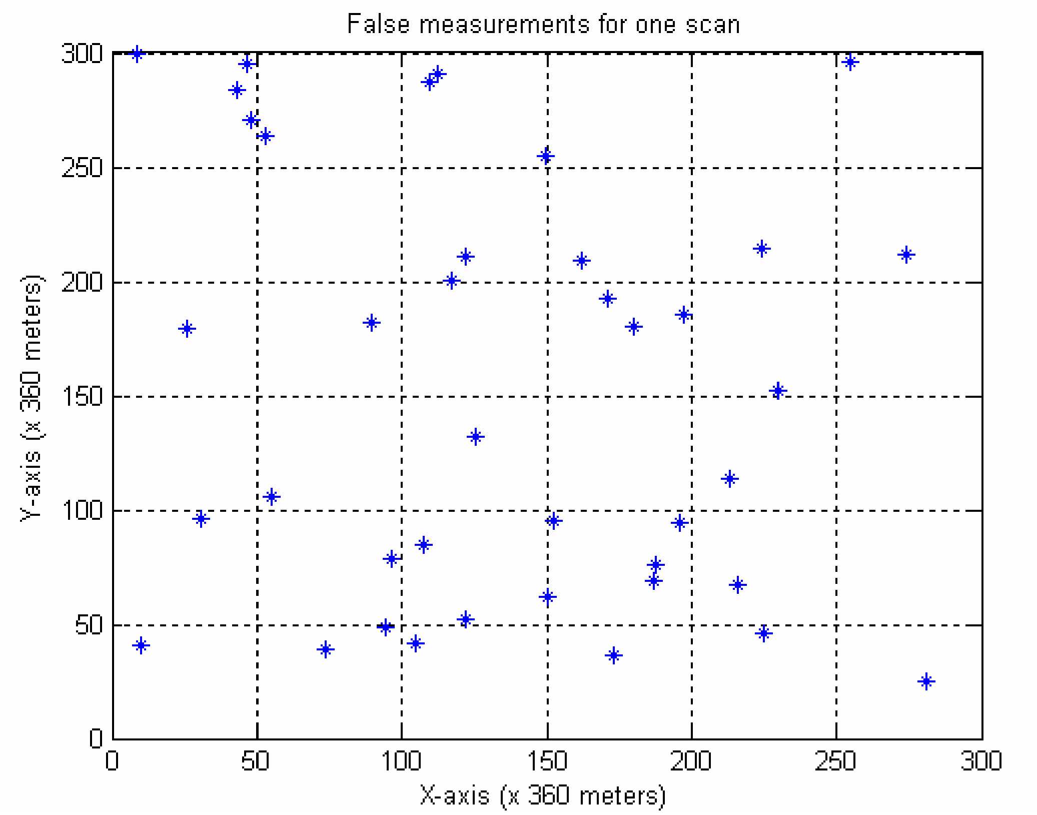

Figure 5-17: False measurements taken from a single scan.

c. By connecting the false measurements that are sufficiently close to each other in subsequent scan, we obtain the false tracks. Here, with “sufficiently close” we mean a maximum of distance that can be traveled by a target between 2 adjacent scan in any axis, x or y. In simulations, this value is determined as 504 meters. This value is found using maximum velocity of the target and the standard deviation of the measurement noise using Equation 5.5. This equation is formed using the gate volume calculation in [22].

$$\text{Max. Distance} = V_{\max} + 2\sigma \tag{5-5}$$

The number of false tracks is the performance criterion for the algorithms. In the simulations TIPM denotes the track initiation probability map.

5.2.3 Simulation Results

Four Monte Carlo simulations are done in this part. The basic parameter that will affect the performance is thought as clutter density. So the first and second simulations compare the sensitivity of the algorithm to clutter density. Third simulation is done to see the effect of measurement noise on the performance. Finally the fourth one compare initiations with and without using TIPM. The parameters common to all simulations in Section 5.2 are given in Table 5-9.

Table 5-9: Common parameters for the simulations in Section 5.2.

Parameters Related with Visibility Map Information	
Used SRTM Data	N39E033.hgt
Target Height (meter)	1400
Algorithm Parameters	
Prob. of Detection (P_D)	0.90
Gate Probability	0.989
Confirmation Threshold	0.90
Termination Threshold	0.10
Markov Chain Matrix	$\begin{bmatrix} 0.98 & 0.02 \\ 0 & 1 \end{bmatrix}$
Cell Size (meter)	20

5.2.3.1 Simulation 1 (S1)

Table 5-10: Parameters for Simulation 1.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0035$
Clutter Density	0.00005 false meas./*cell*
P_{FA}	0.00005
Total Number of True Tracks	131

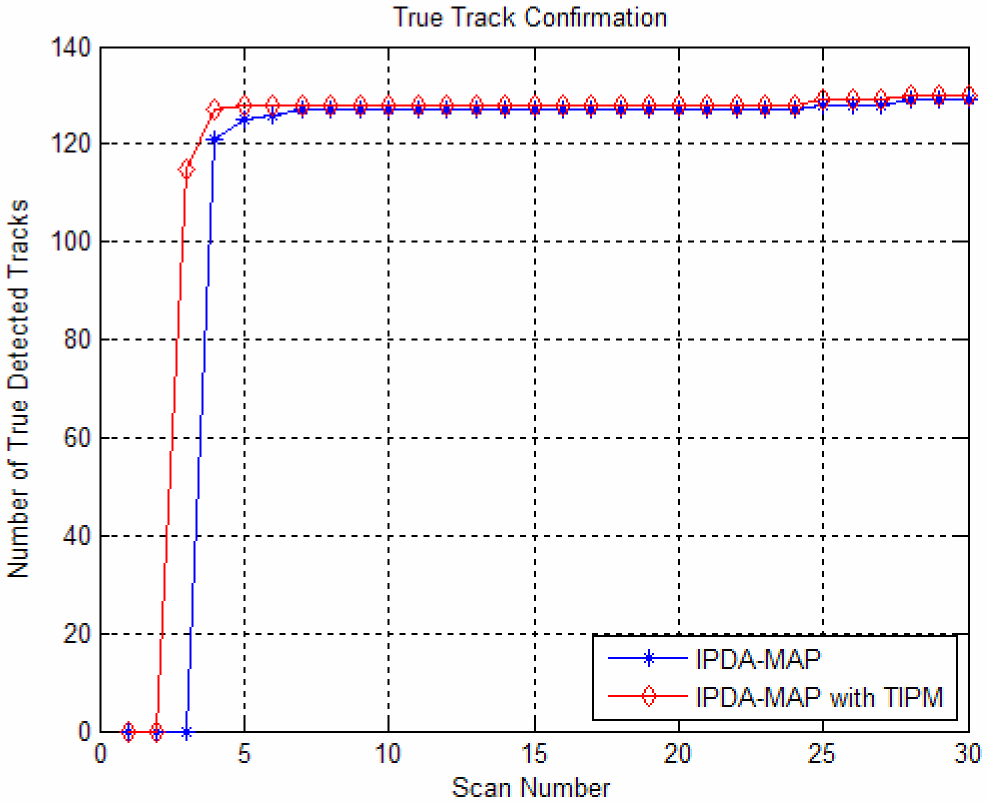

Figure 5-18: True track confirmation for Simulation 1.

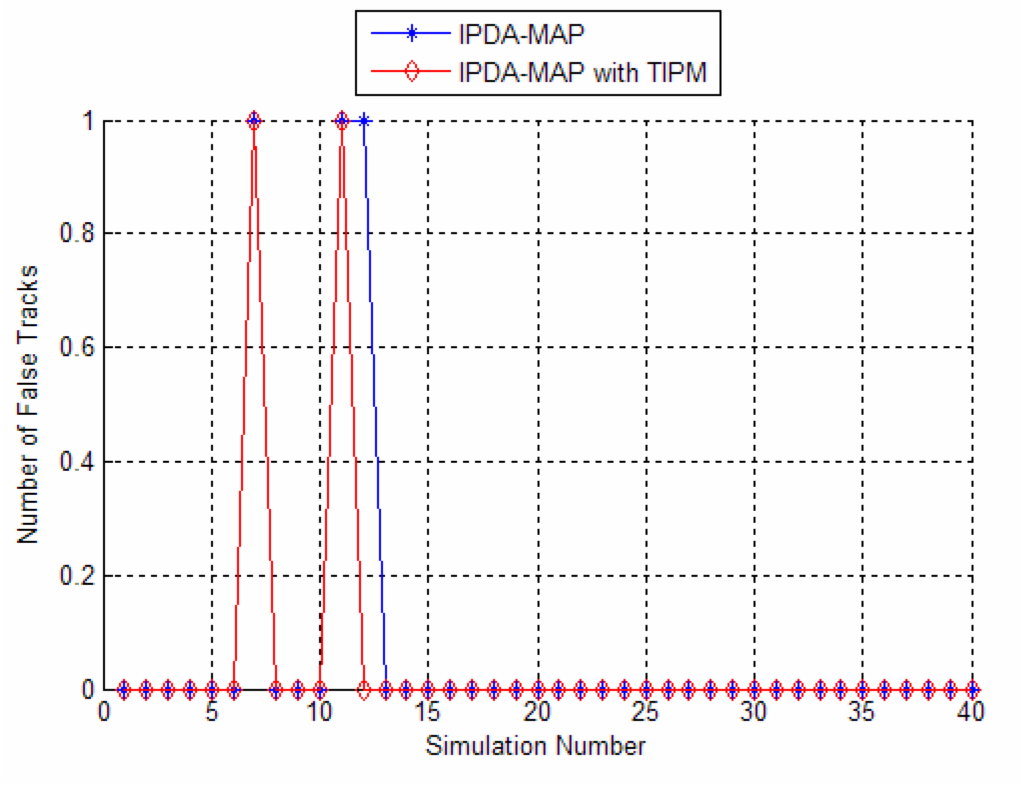

Figure 5-19: False track test for Simulation 1.

A summary of the simulation can be written as:

Mean of false tracks: 0.075 for IPDA-MAP, 0.05 for IPDA-MAP with TIPM.

Variance of false tracks: 0.0712 for IPDA-MAP, 0.0487 for IPDA-MAP with TIPM.

Here we have low clutter density. True track confirmation speed of proposed method is slightly better than IPDA-MAP. False track rejections are almost same.

5.2.3.2 Simulation 2 (S2)

Table 5-11: Parameters for Simulation 2.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0035$
Clutter Density	0.0002 false meas./*cell*
P_{FA}	0.0002
Total Number of True Tracks	129

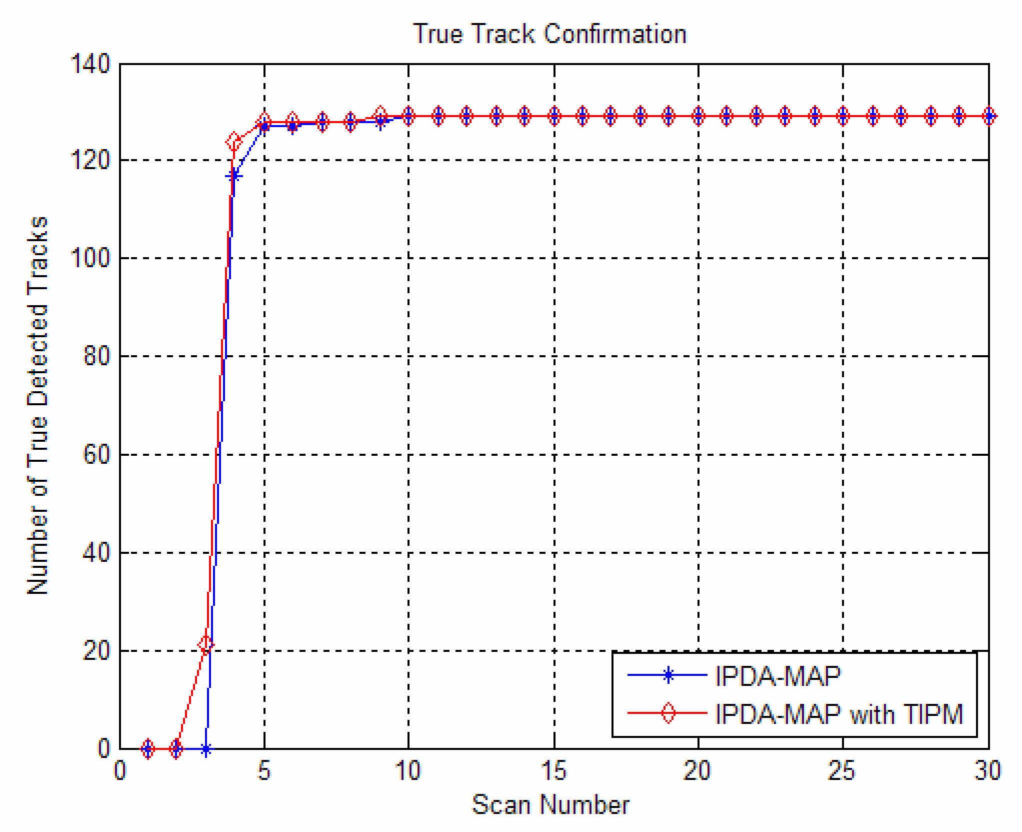

Figure 5-20: True track confirmation for Simulation 2.

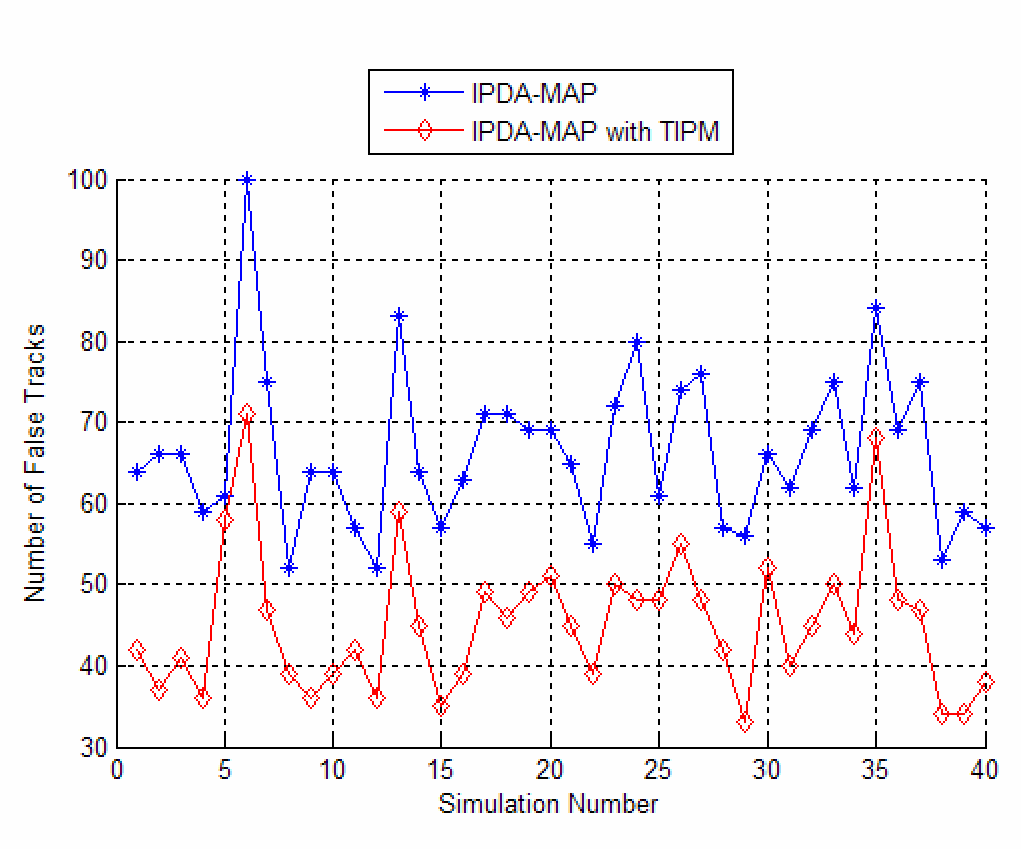

Figure 5-21: False track test for Simulation 2.

Mean of false tracks: 66.35 for IPDA-MAP, 45.12 for IPDA-MAP with TIPM.

Variance of false tracks: 97.46 for IPDA-MAP, 75.29 for IPDA-MAP with TIPM

In this simulation we have high clutter density. So the number of false tracks in both algorithms increases. Proposed method gives a good performance in false track rejection than the other one.

5.2.3.3 Simulation 3 (S3)

Table 5-12: Parameters for Simulation 3.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 100$ Azimuth (rad): Gaussian, $\sigma = 0.018$
Clutter Density	0.00005 false meas./ *cell*
P_{FA}	0.00005
Total Number of True Tracks	114

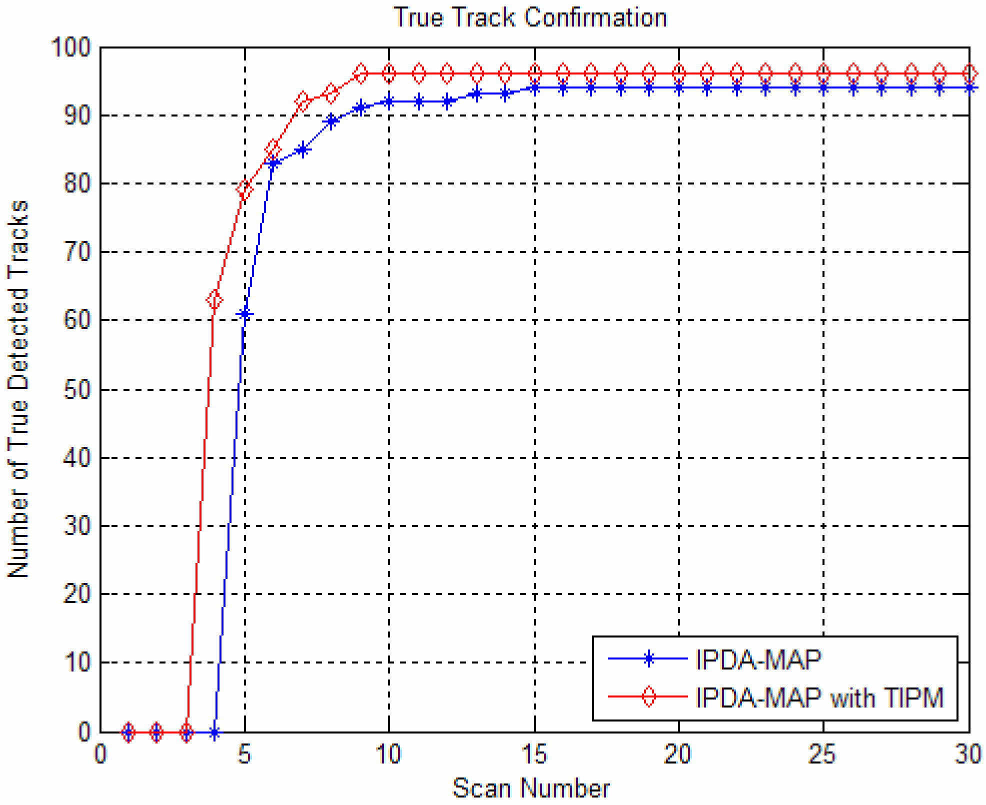

Figure 5-22: True track confirmation for Simulation 3.

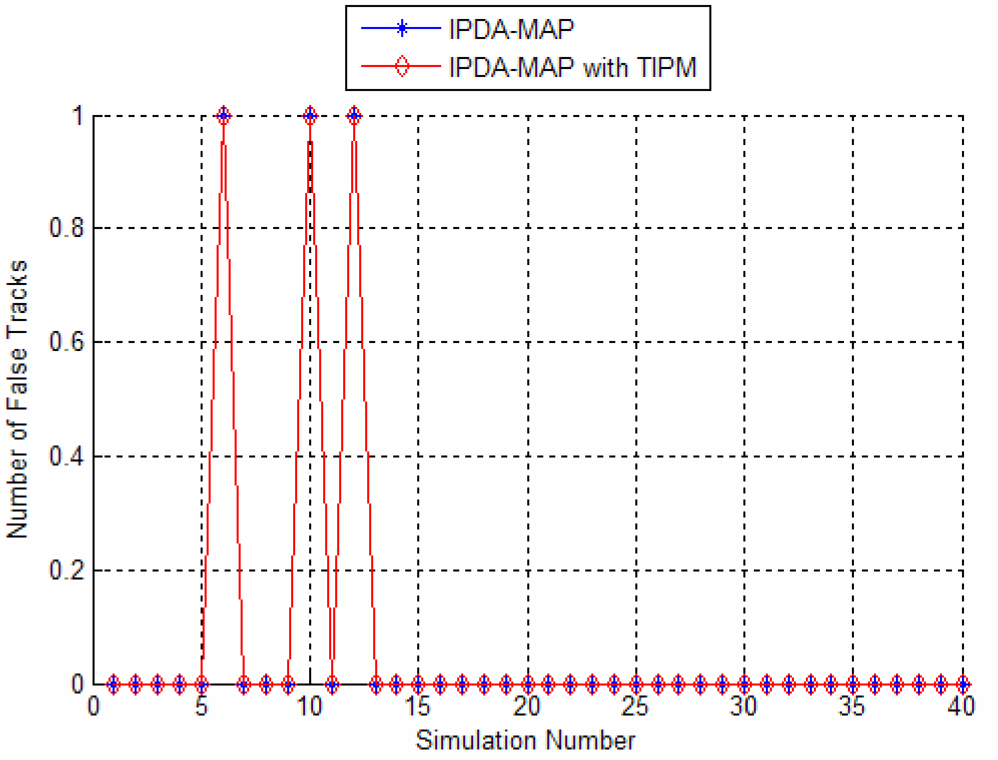

Figure 5-23: False track test for Simulation 3.

Mean of false tracks: 0.075 for IPDA-MAP, 0.075 for IPDA-MAP with TIPM.

Variance of false tracks: 0.0712 for IPDA-MAP, 0.0712 for IPDA-MAP with TIPM.

Simulation 3 is for high measurement noise. Since the measurements are highly noisy both algorithms perform much worse than low measurement noise cases.

5.2.3.4 Simulation 4 (S4)

Table 5-13: Parameters for Simulation 4.

Process Noise Model	X-Y coordinates (meter): Gaussian, $\sigma_x = 50.9$, $\sigma_y = 50.9$
Measurement Noise Model	Range (meter): Gaussian, $\sigma = 50$ Azimuth (rad): Gaussian, $\sigma = 0.0035$
Clutter Density	0.0001 false meas./*cell*
P_{FA}	0.0001
Total Number of True Tracks	155

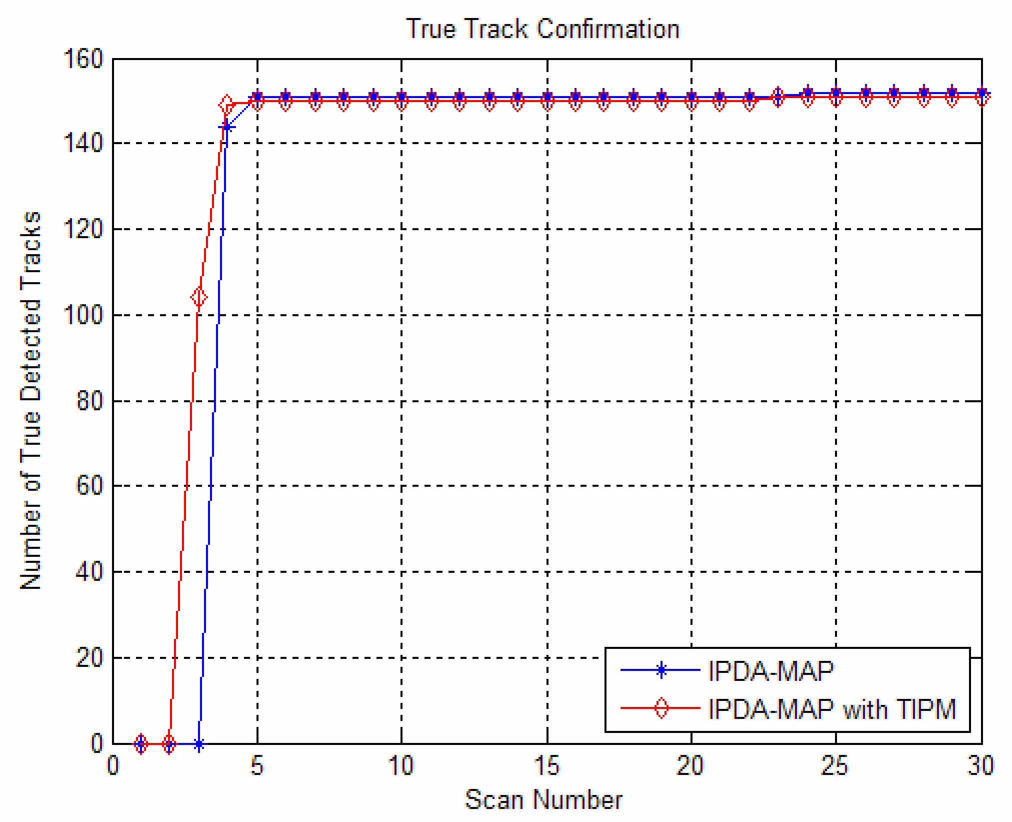

Figure 5-24: True track confirmation for Simulation 4.

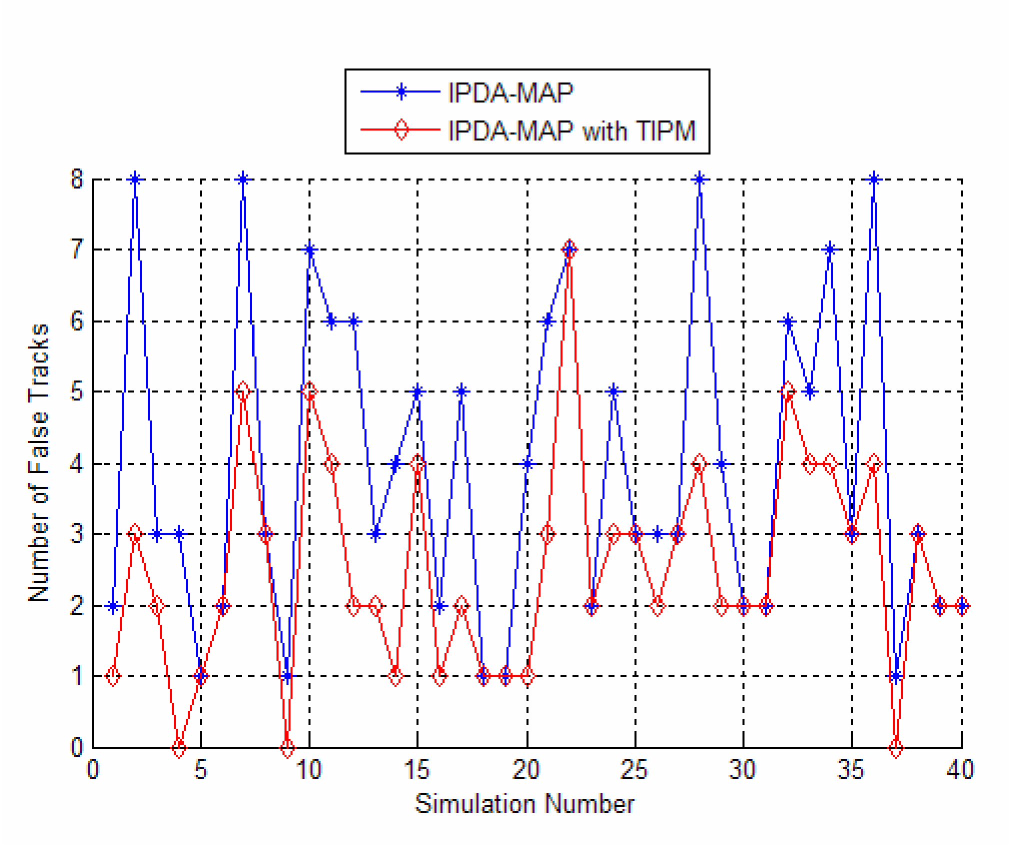

Figure 5-25: False track test for Simulation 4.

Mean of false tracks: 3.93 for IPDA-MAP, 2.53 for IPDA-MAP with TIPM.

Variance of false tracks: 5.05 for IPDA-MAP, 2.36 for IPDA-MAP with TIPM.

5.2.4 Remarks about the Simulations

From the results of Simulation 1 (S1), for a relatively low clutter density, IPDA-MAP algorithm and IPDA-MAP with TIPM algorithm have approximate false track rejection performances. True track confirmation speed of IPDA-MAP with TIPM is higher than IPDA-MAP in this case.

In S2, clutter density is increased. In case of high clutter, the number of false tracks is increased in both algorithms. True track confirmations are similar.

S3 is accomplished to see the effect of measurement noise on the performance. This simulation shows that the proposed algorithm can bring much increase in true track confirmation speed at high measurement noise.

In S4, for clutter density in allowable rate, similar results are obtained.

From the simulation results, we can say that using TIPM at the initialization part of IPDA-MAP algorithm makes the true track confirmation speed faster, although the number of true confirmed tracks is almost same in both cases. At low clutter density, there is no significant difference at false track rejection rate. On the other hand, false track rejection performance of IPDA-MAP with TIPM is higher than IPDA-MAP at high clutter densities.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this study, use of elevation map information of the terrain with tracking algorithms is investigated to improve tracking and track initiation performance of a target tracker.

Firstly, a practical elevation model is obtained to find viewpoint of the target tracker. The SRTM (Shuttle Radar Topography Mission) data is used for that purpose. For a given tracker location and a known constant target height, visible and invisible regions are formed. The regions that are higher than the constant altitude of the target are defined as impassable region, since the target can not pass through these regions at this altitude. This map is called as “visibility map”.

Visibility map is used in tracking algorithms to detect the targets that reappear after passing through an invisible region. Visibility map reduces the size of the covariance matrices both in normal and Unscented Kalman filters. In both filters, using visibility map enables the survival of some tracks that are lost in case of no visibility map. In general visibility map improves the tracking performance of the tracker.

When we compare the two algorithms, standard and unscented based, that use the visibility map we observe that Unscented Kalman filter is much simpler to implement with less computational effort since it works with just a few sigma points (5 sigma points). Furthermore the performance of the Unscented Kalman filter based method is better than the first method for most of the cases.

Visibility map information can also be used for track initiation. For this purpose track initiation probability map (TIPM) is generated from the visibility map of the tracker. After generation, TIPM is used at the initialization part of the track initiation algorithm. As a track initiation algorithm, IPDA-MAP is chosen. This algorithm uses a target probability density for the initial probability of track existence. This density is taken from track initiation probability map and the performance of this algorithm is compared with the one that does not use TIPM. Especially for the regions of high clutter density and for the situations where the measurement errors are high, both the true track confirmation and false track rejection rates are improved by the use of track initiation probability map.

To conclude, use of prior information about the target tracker environment in tracking and track initiation algorithms improves the performance of the target tracker.

6.2 FUTURE WORK

An essential assumption in this work was that the target has a constant altitude. Otherwise, the problem would involve using a different visibility map for each altitude levels. Thus obtaining three dimensional gates by using time varying visibility map is left as a future work.

Another assumption in this thesis is there is a single target in the environment. It will be interesting to work on same problem for multi target case.

Track initiation probability map (TIPM) being used by track initiation algorithm is found by a simple algorithm. It is possible to improve the performance of the tracker by getting a detailed track initiation probability map from visibility maps. This may be another topic to work on it in the future.

REFERENCES

[1] D. Simon, and T. Chia, "Kalman Filtering With State Equality Constraints", IEEE Transactions on Aerospace and Electronic Systems, Page(s): 128-136, 2002.

[2] D. Simon and D.L. Simon, "Kalman filtering with inequality constraints for turbofan engine health estimation", Control Theory and Applications, IEE Proceedings, Volume 153, Issue 3, Page(s): 371-378, 9 May 2006.

[3] D. Simon and D.L. Simon, "Aircraft Turbofan Engine Health Estimation Using Constrained Kalman Filtering", International Gas Turbine and Aero engine Congress and Exhibition, Atlanta, GA, ETATS-UNIS (16/06/2003), Volume 127, Page(s): 323-328, 2005.

[4] Anders Erik Nordsjo, "A constrained extended Kalman filter for target tracking", Radar Conference, 2004, Proceedings of the IEEE Volume, Page(s): 123-127, 26-29 April 2004.

[5] Christopher V. Rao, James B. Rawlings, Jay H. Lee, "Constrained linear state estimation, a moving horizon approach", Automatica 37, Page(s): 1619-1628, 2001.

[6] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", Proceedings of the IEEE Volume 92, Issue 3, Page(s): 401-422, March 2004.

[7] Rambabu Kandepu, Lars Imsland and Bjarne A. Foss, "Constrained State Estimation Using the Unscented Kalman Filter", 16th Mediterranean Conference on Control and Automation Congress Centre, Ajaccio, France June 25-27 2008.

[8] Giorgos Kravaritis and Bernard Mulgrew, "Multitarget Ground Tracking with Road Maps and Particle Filters", IEEE International Symposium on Signal Processing and Information Technology, 2005.

[9] Adam M. Fosbury, John L. Crassidis, Tarunraj Singh, Clyde Springen, “Ground Target Tracking Using Terrain Information”, Information Fusion, 2007 10th International Conference on Publication Date: 9-12 July 2007.

[10] P. Nougues and D. Brown, “We Know Where You Are Going: Tracking Objects in Terrain”, IMA Journal of Mathematics Applied in Business & Industry, Volume 8, 1997, Page(s): 39-58.

[11] B. Pannetier, K. Benameur, V. Nimier, M. Rombaut, “Ground moving target tracking with road constraint”, Signal Processing, Sensor Fusion, and Target Recognition XIII, edited by Kadar, Ivan, Proceedings of the SPIE, Volume 5429, Page(s): 138-149, 2004.

[12] Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan, “Estimation with Applications to Tracking and Navigation”, John Wiley & Sons, 2001.

[13] Yaakov Bar-Shalom, Xiao-Rong Li, “Estimation and Tracking: Principles, Techniques, Software”, Artech House, 1993.

[14] P. R. Kumar, Pravin Varaiya, “Stochastic Systems: Estimation, Identification, Adaptive Control”, Prentice Hall, 1986.

[15] Yaakov Bar-Shalom, Xiao-Rong Li, “Multitarget-Multisensor Tracking: Principles and Techniques”, Clearance Center, 1995.

[16] D. Musicki, R. Evans and S. Stankovic, “Integrated Probabilistic Data Association”, IEEE Transactions on Automatic Control, Volume 39, No: 6, June 1994.

[17] D. Musicki and R. Evans, “Clutter Map Information for Data Association and Track Initialization”, IEEE Transactions on Aerospace and Electronic Systems. Volume 40, Issue 2, Page(s): 387-398, April 2004.

[18] Ning Li Li, X. R, “Tracker Design Based on Target Perceivability”, IEEE Transactions on Aerospace and Electronic Systems, Volume 37, Issue 1, Page(s): 214-225, January 2001.

[19] Neal Madras, “Lectures on Monte Carlo Methods”, American Mathematical Society, 2002.

[20] R. Kandepu, B. Foss and L. Imsland, “Applying the Unscented Kalman filter for nonlinear state estimation”, Journal of process control, Article in press, 2008.

[21] E. A. Wan and R. Van Der Merwe, “The Unscented Kalman Filter for Nonlinear Estimation”, in Proc. of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, Oct. 2000.

[22] Y. B. Shalom, K. C. Chang, H. A. P. Blom, “Automatic Track Formation in Clutter with a Recursive Algorithm” Proceedings of the 28th Conference on Decision and Control, Florida, December 1989.

[23] Wikipedia, http://en.wikipedia.org/wiki/Kalman_Filter, last visited on February 2009.

[24] NASA, ftp://e0srp01u.ecs.nasa.gov/srtm/version2/SRTM3/, last visited on October 2008.

[25] NASA, ftp://e0srp01u.ecs.nasa.gov/srtm/version2/Documentation/SRTM Topo.pdf, last visited on October 2008.

[26] Wikipedia, http://en.wikipedia.org/wiki/Shuttle_Radar_Topography_Mission, last visited on October 2008.

[27] International Water Management Institute, http://www.iwmidsp.org/dsp2/rs-gis-data/River-basins/Aral-sea-basin/02-dem-90m-SRTM-USGS-2003-AOI/ReadMe/readme-first.pdf, last visited on October 2008.

APPENDIX A

SHUTTLE RADAR TOPOGRAPHY MISSION (SRTM)

The Shuttle Radar Topography Mission (SRTM) is an international research of U.S. National Geospatial-Intelligence Agency (NGA) and U.S. National Aeronautics and Space Administration (NASA) to generate digital elevation model of earth. The elevation models are arranged into tiles. These tiles cover one degree of latitude and one degree of longitude [25, 26]. The resolution of the cells is one arc second (approximately 30 meters), called as SRTM-1 data, or three arc seconds (approximately 90 meters), called as SRTM-3 data [25, 26]. SRTM data naming conventions are shown in Table A-1.

Table A-1: SRTM data naming conventions [25].

posting (sample spacing)	**SRTM name**	**DTED equivalent**	**Other data sets**
1 arc-second	SRTM-1	DTED2 (indicating level 2)	-
3 arc-seconds	SRTM-3	DTED1	-
30 arc- seconds	SRTM-30	DTED0	GTOP30

SRTM Data Formats:

The names of individual data tiles refer to the latitude and longitude of the lower left corner of the tile [25]. As an example, SRTM data named as N20W100 are 20 degrees north latitude and 100 degrees west longitude.

Because SRTM-1 data are sampled at one arc-second in latitude and longitude, each file contains 3601 lines and 3601 samples [25, 26]. For SRTM-3 data there are 1201 lines and 1201 samples as a result of sampling at 3 arc-seconds. An example of SRTM data is shown in Figure A-1.

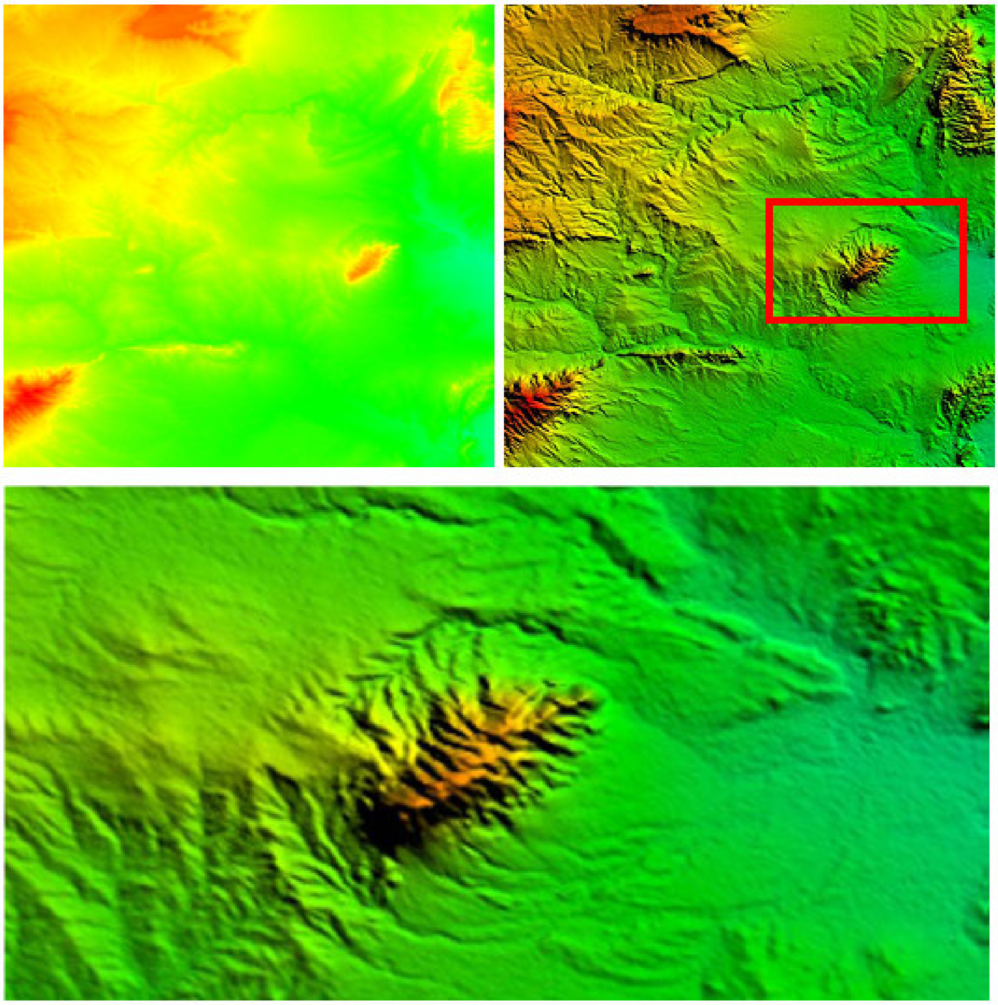

Figure A-1: SRTM sample image [27].

In this thesis SRTM-3 data is used in simulations as SRTM-1 data have only been released over United States. The availability of SRTM data is shown in Table A-2 below.

Table A-2: SRTM data availability [25].

	SDDS **'Seamless server'** **http://seamless.usgs.gov/**	**Mail order**	**LP DAAC** **ftp://e0srp01u.ecs.nasa.gov/srtm/**
Vers. 1	-	-	1" U.S. 3" world – averaged 30" world – averaged Format: SRTM
Vers. 2	1" U.S. 3" world – sub sampled Formats: Arc Grid, Bil, TIFF, Grid Float	1" U.S. 3" world – sub sampled Formats: DTED, SRTM	1" U.S. 3" world – averaged 30" world – averaged Format: SRTM

DEM File (.HGT):

Digital elevation model (DEM) is provided as 16-bit signed integer data in a sample binary raster with elevations are in meters [25]. Elevations can range from -32767 to 32767 meters. The data are stored row by row (all the data for row1, all the data for row 2, etc.). Byte order is big-endian standard. Data voids because of shadowing, phase unwrapping anomalies, or other causes are flagged by the value -32768 [25].

APPENDIX B

CONSTRUCTION OF THE VISIBILITY MAP

SRTM data introduced in Appendix A is used to construct the visibility map information for a target tracker. In this thesis the visibility map is constructed for a constant target height and constant target tracker location. First the elevation map of the terrain in which the target tracker located is obtained from SRTM data. In Figure B-1 an example of an elevation map is shown. The resolution of this map is 360 m.

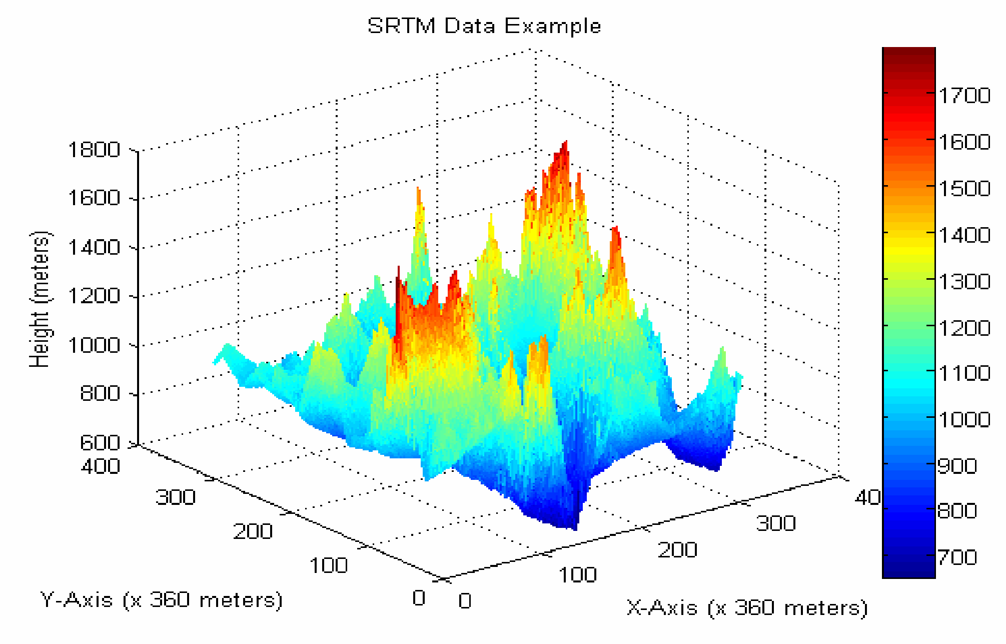

Figure B-1: Elevation map data (SRTM data).

Then the location of the target tracker (x_r, y_r, z_r) and the height "h" of the target are determined. By using these information the points that are on the $z = h$ are checked whether they are in line of sight of the target tracker which is located at (x_r, y_r, z_r). The points that are in line of sight with the target tracker form the visible region while the points that are not in line of sight form the invisible region. On the elevation map, the points which are higher than h form the impassable region.

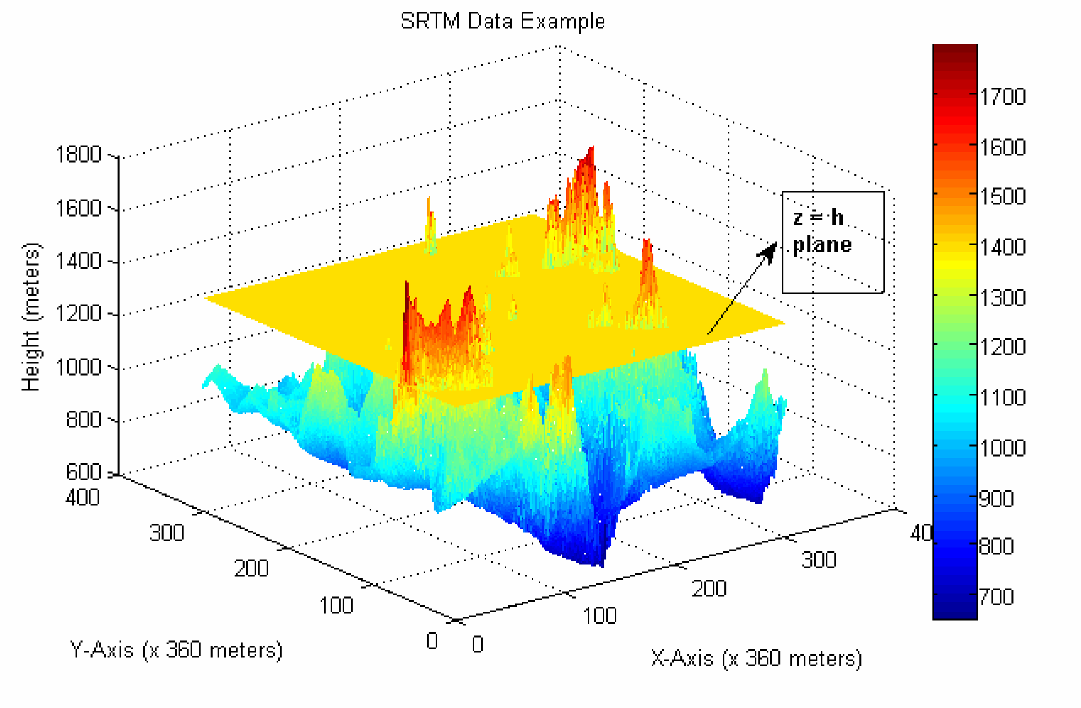

Figure B-2: Elevation map data with z=h plane.

For the example in Figure B-3, for an airborne target with height h1, the point belongs to impassable region as h1 is below the altitude of the terrain. For target height h2, the point is in invisible region. Finally, for the target height h3, the point is at visible region.

It is seen that the visibility map depends on the radar position and the constant height of the target. So the visibility map changes when these parameters are varied.

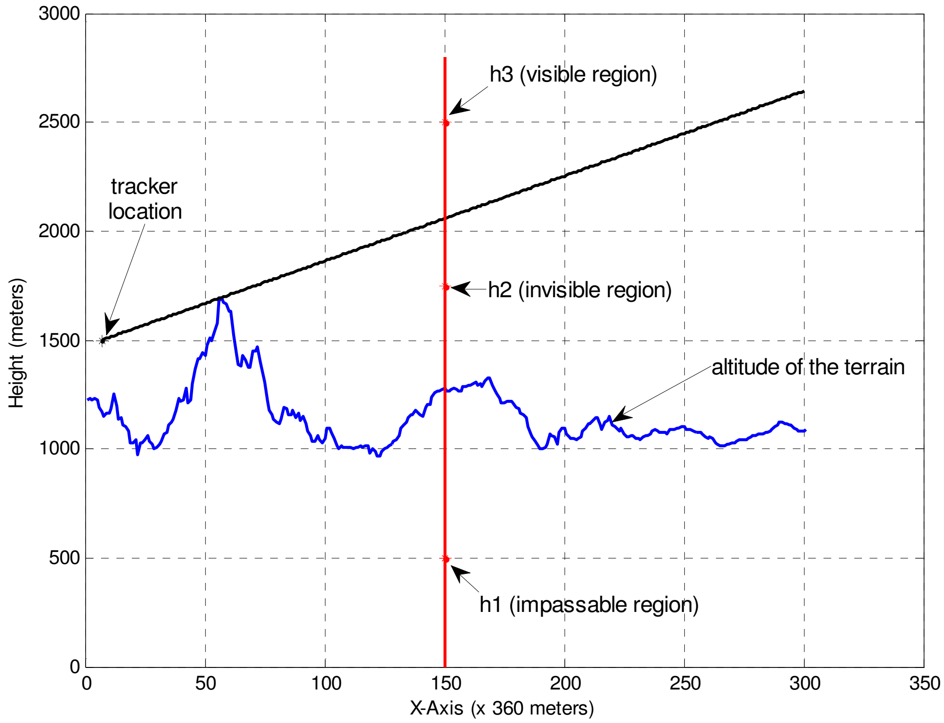

Figure B-3: Visible, invisible and impassable regions.