GLOBAL URBAN LOCALIZATION OF AN OUTDOOR MOBILE ROBOT USING SATELLITE IMAGES

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

BY

CAN ULAŞ DOĞRUER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

FEBRUARY 2009

Approval of the thesis:

GLOBAL URBAN LOCALIZATION OF AN OUTDOOR MOBILE ROBOT USING SATELLITE IMAGES

submitted by **CAN ULAŞ DOĞRUER** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Suha Oral _____
Head of Department, **Mechanical Engineering**

Assist. Prof. Dr. A. Buğra Koku _____
Supervisor, **Mechanical Engineering Dept., METU**

Assist. Prof. Dr. Melik Dölen _____
Co-Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. M. Kemal Özgören _____
Mechanical Engineering Dept., METU

Assist. Prof. Dr. A. Buğra Koku _____
Mechanical Engineering Dept., METU

Prof. Dr. Tuna Balkan _____
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Afşar Saranlı _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Veysel Gazi _____
Electrical and Electronics Engineering Dept., TOBB ETU

Date: **11.02.2009**

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 : Can Ulaş DOĞRUER

Signature :

ABSTRACT

GLOBAL URBAN LOCALIZATION OF AN OUTDOOR MOBILE ROBOT USING SATELLITE IMAGES

Doğruer, Can Ulaş

Ph.D., Department of Mechanical Engineering

Supervisor : Assist. Prof. Dr. A. Buğra Koku

Co-Supervisor : Assist. Prof. Dr. Melik Dölen

February 2009, 233 pages

In this dissertation, the mapping of outdoor environments and localization of a mobile robot in that setting is considered. It is well known that in the absence of a map or precise pose estimates, localization and mapping is a coupled problem. However, in this dissertation this problem is decoupled in to two disjoint steps; mapping and localization on the acquired map. First the images of the outdoor environment is downloaded from a website such as *Google Earth* and then these images are processed by utilizing several artificial neural network topologies to create maps. Once these maps are obtained, the localization is done by using Monte Carlo localization.

This dissertation addresses a solution for the information which is most of the time taken for granted in most studies; a prior map of environment. Mapping is solved by using a novel approach; the map of the environment is created by processing satellite images. Several global localization techniques are developed and evaluated to be used with these map so as to localize a mobile robot globally.

The outcome of this novel approach presented here may serve as a virtual GPS. Mobile phone applications can localize a user within a circle of uncertainty without GPS. This crude localization may be used to download relevant satellite images of the local environment. Once the mobile robot is localized on the map created from the satellite images by using available techniques in the literature i.e. Monte Carlo localization, it may be claimed that it is localized on Earth.

Keywords: Satellite Images, Mapping, Outdoor Mobile Robot, Localization, SLAM.

ÖZ

UYDU GÖRÜNTÜLERİ KULLANARAK, BİR ARAZİ MOBİL ROBOTUNUN EVRENSEL YERSELLEŞTİRİLMESİ

Doğruer, Can Ulaş
Doktora, Makina Mühendisliği Bölümü
Tez Yöneticisi : Yard. Doç. Dr. A. Buğra Koku
Ortak Tez Yöneticisi : Yard. Doç. Dr. Melik Dölen

Şubat 2009, 233 sayfa

Bu tezde, arazi ortamlarının hartilanması ve mobil bir robotun konumunun kestirilmesi çalışılmıştır. Çok iyi bilinmektedir ki harita olmadığı zaman veya mobil robotun konumu hassas olarak bilinmediği zaman haritalama ve konum kestirme tümleşik bir problemdir. Ancak, bu tez de konum kestirme ve haritalama birbirinden bağımsız iki alt probleme indirgenmiştir. Öncelikle arazi ortamının görüntüsü mesela Google Earth internet sayfasından indirilmiş ve daha sonra farklı yapay sinri ağları topolojileri kullanılarak harita haline dönüştürülmüştür. Bir kez bu haritalar elde edilince, konum kestirme Monte Carlo konum kestirme metodu ile yapılmıştır.

Bu tez bir çok çalışmada kullanılan ancak nasıl elde edilidiği fazla irdelenmeyen çevre haritasını adreslemektedir. Haritalama özgün bir yaklaşımla elde edilmiştir; harita uydu görüntülerini işleyerek oluşturulmuştur. Mobil robotun global konum kestirimini başarmak için bu haritalarla kullanılmak üzere değişik konum kestirme metotları geliştirilmiş ve değerlendirilmiştir.

Bu özgün yaklaşımın çıktısı yapay GPS olarak kullanılabilir. Cep telefonu uygulamaları GPS kullanmadan kullanıcıyı belli bir dairesel alan içerisinde konumlandırabilmektedir. Kabaca bulunan konum, ilgili uydu görüntülerinin internetten indirilmesi için kullanılabilir. Bir kez robot bu uydu görüntülerinden oluşturulan haritada mesela Monte Carlo metodu ile konumlandırıldığı zaman, mobil robotun Dünya üzerindede konumu kestirilebilir.

Anahtar Kelimeler: Uydu Resimleri, Haritalama, Arazi Mobil Robot, Yerselleştirme, Eş Zamanlı Konum Belirleme ve Haritalama.

To my parents
Nermin and Şadi

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisors Assist. Prof. Dr. A. Buğra Koku and Assist. Prof. Dr. Melik Dölen for their guidance, advice, criticism, encouragements and insights throughout this research.

I would also like to thank my parents, Şadi Doğruer and Nermin Doğruer, and my brother MSc. Civil Engineer (METU), Alper Doğruer for their support, shelter and encouragements.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

x_{ij}	Elements of Error matrix
$\widehat{\boldsymbol{x}}_t^+$	Updated state
$\widehat{\boldsymbol{x}}_t^-$	Predicted state
$\widehat{\boldsymbol{y}}_t^+$	Updated Information vector
$\widehat{\boldsymbol{y}}_t^-$	Predicted information vector
O_n	Distance to the n^{th} landmark
$g_{t+1}(.)$	Trial distribution
r_l	Left wheel radius
r_r	Right wheel radius
w_k	Importance weight
x_l	x coordinate of point landmark
y_l	y coordinate of point landmark
$\boldsymbol{A}_t$	Jacobian of Process model
$\boldsymbol{H}_t$	Jacobian of Measurement model
$\boldsymbol{K}_t$	Kalman Gain
$\boldsymbol{P}_t^+$	Updated Covariance
$\boldsymbol{P}_t^-$	Predicted Covariance
$\boldsymbol{P}_{ll}$	Covariance matrix of landmarks
$\boldsymbol{P}_{vl}$	Cross covariance matrix of vehicle and landmark states
$\boldsymbol{P}_{vv}$	Covariance matrix of vehicle

$\boldsymbol{Q}_k$	Process noise covariance matrix
$\boldsymbol{R}_t$	Measurement noise covariance matrix
$\boldsymbol{U}_{0:t}$	Set of action command from time 0 up to time t
$\boldsymbol{X}_{0:t}$	Set of Dynamic system / robot states vector from time 0 to time t
$\boldsymbol{Y}_t^+$	Updated Information matrix
$\boldsymbol{Y}_t^-$	Predicted information matrix
$\boldsymbol{Z}_{0:t}$	Set of observation vector from time 0 to time t
$\boldsymbol{u}_t$	Action command at time t
$\boldsymbol{v}_t$	Measurement noise
$\boldsymbol{w}_t$	Process noise
$\boldsymbol{x}_{l,n}^T$	State of nth point landmark
$\boldsymbol{x}_l$	Set of state of l^{th} landmark
$\boldsymbol{x}_{sys}$	Augmented system state vector
$\boldsymbol{x}_t$	Dynamic system / robot states vector at time t
$\boldsymbol{z}_t$	observation vector at time t
α'_{0i}	Weights of Extended Kalman filter
λ_k	Mixture index
ξ'_0	Number of Extended Kalman filter
π_t	Conditional probability of states at time t given the observations upto time t
ω_l	Left wheel angular velocity
ω_r	Right wheel angular velocity
b	Wheel base

$\mathbf{I}$	Identity matrix
L	Axle distance
W,F	Weights
β	Steering angle
$L(\hat{\boldsymbol{x}}_{i,t+1} - \boldsymbol{x}_{i,t+1}^{k})$	Cost function
$N(\boldsymbol{x}_0 : \boldsymbol{x}'_{0i}, \boldsymbol{P}'_{0i})$	Gaussian distribution whose mean is and covariance is
$P[\boldsymbol{x_t}\|\boldsymbol{x_{t-1}}, \boldsymbol{u_t}]$	Probabilistic motion model
$P[\boldsymbol{x_t}\|\boldsymbol{Z}_{0:t-1}]$	Prior distribution of states
$P[\boldsymbol{x_t}\|\boldsymbol{Z}_{1:t}]$	Posterior distribution of states
$P[\boldsymbol{z}_t\|\boldsymbol{x_t}]$	Probabilistic observation model
$P(x)$	Probability of x
$P(x\|y)$	Conditional probability of x given y
T	Sampling period
x	Cartesian coordinate of reference point on mobile robot
y	Cartesian coordinate of reference point on mobile robot
$\boldsymbol{P}$	Covariance matrix
$\boldsymbol{f}$	Process model
$\boldsymbol{h}$	Measurement model
θ	Heading angle of mobile robot, measured CCW from x-axis
κ	Kappa coefficient

LIST OF ABBREVIATIONS

ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
CEKF	Compressed Extended Kalman Filter
DBN	Dynamic Bayesian Network
DGPS	Differential Global Positioning System
DP-SLAM	Distributed Particle SLAM
EKF	Extended Kalman Filter
FastSLAM	Factored Solution to SLAM
FCM	Fuzzy C-Means
FIS	Fuzzy Inference System
GPS	Global Positioning System
GUL	Global Urban Localization
HMM	Hidden Markov Model
HSV	HSV colors cape
HYMM	Hybrid Metric Map
KF	Kalman Filter
LTR	Local Triangular Regions
MAP	Maximum A Posterior
MCL	Monte Carlo Localization
OG	Occupancy Grid
PDF	Probability Density Function
PF	Particle Filter

PNN	Probabilistic Neural Network
RST	Remote Sensing Technology
SEIF	Sparse Extended Information Filter
SIR	Sampling Importance Resampling
SIS	Sequential Importance Sampling
SLAM	Simultaneous Localization And Mapping
SMC	Sequential Monte Carlo
UIF	Unidirectional Information Flow

CHAPTER 1

INTRODUCTION

Autonomous navigation is one of the major capabilities that a mobile robot should possess. Autonomous navigation includes many subtasks such as path planning, obstacle avoidance/reactive navigation and localization. .Above all, localization is the major capability that leads to absolute autonomy for a mobile robot. Therefore, in the last decade, many researchers have studied localization problem. Some researchers studied ***S****imultaneous* ***L****ocalization* ***A****nd* ***M****apping* (*SLAM*) problem in order to localize the robot with respect to the best map obtained so far while others have focused on localization problem assuming that *a priori map* of the environment is available. However, the question of "*How the map is acquired?"* cannot be answered easily, even for simple indoor environments. The problem becomes more complicated in a large scale outdoor environment. Hence, in this dissertation this fundamental question is addressed while the robot is being localized. In short, the *global localization* of an outdoor mobile robot is investigated in this dissertation. The *SLAM* problem is avoided by focusing on localization while using satellite images as an alternative to mapping. It is believed that this novel method will contribute to the autonomy of outdoor mobile robots. Next, typical approaches to localization are discussed qualitatively. This discussion will lead to informal statement of the objective of this dissertation.

Localization may be grouped as relative positioning (a.k.a. "dead-reckoning") and absolute positioning (so called "reference based systems") that requires the knowledge on some references whose locations are known precisely to localize without a map i.e. *SLAM* problem. Indeed, *SLAM* is the ultimate problem that must be solved in mapping and localization domain. However, it is proven to be difficult to solve in every operational scenario, so dead-reckoning and absolute localization have still been subject of several studies.

It is well known that the map of the local environment may not be available all the time. This problem has been recognized since the early days of mobile research. Consequently, *Localization and Mapping* issues have been regarded as coupled for a long time. This resulting problem is referred to as *SLAM* in the robotics literature.

Later, the *SLAM* has attracted attention from different mobile robotics research groups, these attempts are founded on *Extended Kalman Filter* (*EKF*). *EKF* based solution to *SLAM* was first proposed by the seminal paper of (Randall 1990) and later developed and optimized by different researchers. Although there are practical applications of *EKF* based solution to *SLAM*, some issues still require attention. First, the run time and storage associated with the original *EKF* based solution to *SLAM* is quadratic in the number of features in the environment. Therefore, in its original form, it cannot be implemented in large-scale environments. In fact, the original solution to the *SLAM* is restricted to environments where the number of landmarks is only a few hundred. Generally, the solution to *SLAM* problem cannot represent the environment in dense form but rather sparse maps are common. These sparse maps are not satisfactory for navigation purposes. Another big deal of *SLAM* problem is the data association. When the vehicle pose uncertainty is high during loop closing, data association is not an easy task. Since *EKF* based solution to *SLAM* holds unimodal belief in the robot pose and environment map, any data misassociation will lead to catastrophic failure of the method where the *EKF* filter cannot recover true solution again.

A number of researchers have studied localization alone. In these studies, the position tracking of mobile robot is investigated. It is assumed that the initial pose of the robot is known. The aim is to estimate the state of the mobile robot with a reasonable accuracy. Different sensors with various characteristics can be utilized for this purpose. The most common sensors are odometer, inertial navigation system (*INS*), sonar, laser range finders, and cameras. The underlying difficulty in estimating the state of a mobile robot stems from the

fact that the sensor readings are corrupted with noise and that the sensor models along with the process model of the dynamic system (mobile robot) is inaccurate. Since the estimation of the mobile robot's position when the initial pose of the robot is given is known as a tracking problem, the *EKF* is based on a recursive solution that is performed in two stages: prediction and update respectively. It is well known fact that the covariance matrix of states grows in time which in turn implies that as the robot navigates through a local environment, its pose is known with less certainty. Consequently, in order to limit the covariance of states, the update stage requires absolute information to reset the filter at regular time intervals. However, this requires a map of the landmarks around the environment.

In the light of previous discussion, the followings can be said: *The SLAM problem is difficult to solve because the problem itself scales quadratically with number of features in the map and data association might be challenging for large scale environments. In SLAM, the map representation is sparse which is not suitable for other basic tasks but the localization. Furthermore, the tracking techniques cannot cope with multi-hypothesis, so they cannot solve global localization. Besides, the process and measurement noise characteristics have big impact on the performance. Tuning these parameters is non-trivial especially when the process noise is time dependent.* Therefore, in this dissertation, a novel solution is proposed to tackle with the *global localization* of outdoor mobile robots operating in urban settings, which may be classified as map based localization.

In this dissertation, the localization problem is addressed. It is assumed that a mobile robot operates in urban settings and has a map of the environment. The *SLAM* options can be dropped out due to that respect. Thus, the whole operation focuses on localization of a mobile robot using a given map. The most important contribution of this dissertation is the method of acquiring the map of the urban setting. That is, the satellite images of urban areas are downloaded from the Internet and are converted into meaningful maps to be

used along with suitable localization techniques. Consequently, this dissertation is proposes a solution to localize a mobile robot in urban setting by using techniques in the literature together with satellite images downloaded from the Internet free of charge.

The rest of the dissertation is organized as follows. Chapter Two is devoted to literature survey. At the end of this chapter, the formal statement on the scope of dissertation is stated and the objective is justified by considering previous related works. Similarly, Chapter Three concentrates on the review of probabilistic state estimation (*tracking*) since the localization of a mobile robot is considered as a special case of general probabilistic state estimation. In Chapter Four, mapping techniques are studied as separate prerequisite. The potential use of satellite pictures as map of environment is discussed and several methods are proposed and their performances in converting satellite pictures into maps are evaluated. In Chapter Five, the novel solution to GUL[1] problem is elaborated and several issues related with this approach is illustrated and discussed. Likewise, Chapter Six investigates alternative solutions for GUL problem while Chapter Seven presents the experimental studies. Finally, Chapter Eight summarizes the key points of this dissertation. The future work is stated here to highlight the possible research directions.

[1] GUL is an acronym for Global Urban Localization problem defined in this dissertation. It is definition is given in scope of dissertation section of Chapter 2

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

In this chapter, related work is reviewed. First, localization problem alone is considered. Here the efforts are split up into two groups (Borenstein 1997):

- Relative position measurements (also called dead-reckoning)
- Absolute position measurements (reference based systems)

In the second part of this chapter, SLAM related works will be discussed. Here the aim is to describe the SLAM history and current state of the art methods to solve SLAM problem.

The general taxonomy of approaches to localization that is used in this study follows the style in the study of (Borenstein 1997) and the literature review about SLAM follows the style of paper published by (Bailey 2006), but the cited materials are original to this work, with the possible up to date papers added whenever necessary. Besides the Extended Kalman Filter based approaches are elaborated further. The sole aim of this chapter is not just to review the related works but to justify the scope of dissertation. So at the end of this chapter, a discussion is made about pros and cons of different approaches to localization and SLAM problem. This discussion leads to the scope of dissertation which is stated at the end of this chapter.

2.2 Survey of Mobile Robot Localization Papers

2.2.1 Definition of Mobile Robot Localization Problem

An autonomous mobile robot is placed in a known environment and equipped with a number sensors e.g. inertial navigation system, gyroscopes, accelerometers, inclinometers, odometer and sonar, laser or camera. The sole

aim of the robot is to track its position in the environment. Right at this point, following the style of (Borenstein 1997), approaches to localization can be divided into two categories:

- Dead-reckoning
    - Odometer
    - Inertial navigation
- Absolute position measurements
    - Magnetic compasses
    - Active beacons
    - Global positioning systems
    - Landmark navigation
    - Model matching

In the *dead-reckoning*, the robot monitors its internal parameters and the relative displacements are computed. By using these data, the position of the robot with respect to previous location and in general with respect to the known starting position is determined.

Odometer is an attractive solution for dead-reckoning because it is cheap and provides accurate position data for short-distance navigation. Generally, odometer is used as an integral part of any navigation system of a mobile robot. However, the use of odometer is not free of problems. Odometer simply measures the revolution of wheels and converts the angular speed into translation of the mobile robot base. In order to compute relative displacement, it integrates the angular velocity of both wheels. Hence errors in the position of the mobile robot accumulate in time. Sometimes, the conversion formula proves wrong due to error sources which are generally divided into two:

- Systematic errors
- Non-systematic errors

Systematic errors are attributed to unequal right/left wheel diameter, uncertainty in the wheelbase. *Non-systematic errors* are attributed to wheel slippage or collision with an obstacle in the environment.

Gyroscopes and accelerometer are used in *inertial navigation*. The output of gyroscope and accelerometer, the rate of rotation and acceleration, respectively must be integrated twice to determine the relative displacement. Thus, when they are operated for long times output of inertial navigation devices drift and errors tend to grow in time. It is well known that these errors in both odometer and inertial navigation system are due to summation operation. Although, odometer and inertial navigation devices are self-contained, mobile robot cannot depend on them for long term operation.

In *absolute position measurement*, the absolute measurements of robot states are made which does not depend on the previous measurements. This approach is more stable than dead-reckoning since the absolute measurements resets error at certain time intervals thus error changes in certain bandwidth, but never diverges.

It is well known that heading is the most important parameter that cause large drift in position, because any small error in heading will grow in time proportional to the length of path traversed. *Magnetic compasses* which measures absolute heading parameter, may be used to force the heading error to oscillate within certain band. However, some remarks must be made; the earth magnetic field is distorted by power lines and nearby object and they make compass less reliable as an absolute measurement systems.

Active beacons are a second class of absolute position measurements system. It uses a set of active beacon located in the environment precisely. One of the main disadvantages of this positioning systems; the environment must be engineered beforehand. Basically there are two types of active beacons system: Triangulation, Trilateration.

The *Global Positioning System* (GPS) is a satellite based positioning system; today it is the prevalent means of positioning in outdoor environment. However, GPS cannot be used in indoor environments and the GPS data is distorted intentionally, so it is not precise. There are other problems; when there are large objects nearby the GPS may not be very accurate or position data may not be received.

Landmark navigation is another subclass of absolute position measurement systems. The mobile robot is supposed to measure /detect distinct features in the environment in order to localize itself. Basically, the landmarks are divided into two:

- Artificial landmarks
- Natural landmarks

Landmark's definitions are selected and loaded into memory of mobile robot, before robot is operated. The robot is to identify the preloaded landmarks in the environment; it assumed that the precise locations of the landmarks are known. This requirement is a big deficiency for this method.

Map based positioning systems requires that the robot build its own map and the compare it with a priori map of the environment so as to localize itself. However, there are some questions that must be addressed. First, the robot needs an exploration strategy for mapping and the mapping demands high precision sensors. Finally, acquisition of a prior map is an open challenge.

Finally, both the dead-reckoning and absolute position measurements system has pros and cons. Therefore, mobile robots are generally equipped with a hybrid positioning system which combines the best properties of both systems

2.2.2 Mathematical Statement of Mobile Robot Localization Problem

Assuming that the mobile robot operates in a planar world, the robot states can be represented with $\{x, y\}$, Cartesian coordinates of the mobile robot reference

point and the heading angle denoted by θ, measured *CCW* from the *x-axis.* Assuming that the landmarks are point features, their locations can be represented with $\{x, y\}$ in Cartesian coordinates. In short the robot states can be represented in vector form:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2.1}$$

The sole objective is to compute the best estimate of the vehicle states while it navigates in the environment. Indeed this is a typical tracking problem where the object that is being tracked is a mobile robot. There are a number of methods in the literature that can be used to solve this problem. The general definition of this problem can be stated as follows

The main objective of *localization/tracking* can be defined as *estimation* that is to *track states/pose of robot* as it change in time. At discrete time intervals *action commands* which modifies the system according to the *system model*, are executed. At certain times observations are received that corrects the states predicted by *system model*.

The simplest approach to localization is to integrate the odometer data once and compute the relative position by using the kinematic model of the mobile robot. Similarly, inertial navigation system may integrate the sensor data so as to compute the relative displacement and heading of the mobile robot. However, these approaches are not free of errors and they are not dependable for long-term navigation. In multi-sensor approaches, Kalman filter is the best choice to fuse data coming from different sensors. It may produce an optimal estimate which represents relative displacement of the mobile robot.

In general, the *estimation* is realized in two stages: *prediction* and *correction*. After each action is executed, states are modified according to the *system model* including the process noise (*prediction stage*). Then predicted states are corrected with the help of most recent measurements (*correction stage*).

Formally, assuming the states/pose of the robot is known in previous time instants, the effect of action is considered by obtaining a *prior PDF* by using the system model and the command itself, in other words the states/pose of the robot is predicted. Whenever new sensory information is available, it is used to correct /update the prior *PDF* so as to obtain posterior *PDF* which is certainly more accurate than the prior *PDF*. The general estimation may be represented by recursive Bayes filter. These formulas are given below:

$$P[\boldsymbol{x_t}|\boldsymbol{Z}_{0:t-1}] = \int P[\boldsymbol{x_t}|\boldsymbol{x_{t-1}}, \boldsymbol{u_t}]\ P\left[\boldsymbol{x_{t-1}}\middle|\boldsymbol{Z}_{1:t-1}\right] d\boldsymbol{x_{t-1}} \tag{2.2}$$

Then at time step t a measurement y_t becomes available and may be used to *update the prior PDF via Bayes rules.*

$$P[\boldsymbol{x_t}|\boldsymbol{Z}_{1:t}] = \frac{P[\boldsymbol{z}_t|\boldsymbol{x_t}|]P[\boldsymbol{x_t}|\boldsymbol{Z}_{0:t-1}]}{P[\boldsymbol{z}_t|\boldsymbol{Z}_{1:t-1}]} \tag{2.3}$$

Where the normalizing denominator is given by

$$P[\boldsymbol{z}_t|\boldsymbol{Z}_{0:t-1}] = \int P[\boldsymbol{z}_t|\boldsymbol{x_t}]P[\boldsymbol{x_t}|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x_t} \tag{2.4}$$

There are different approaches to simulate this recursive Bayes formula; Particle filter, sample based technique, and Kalman filter. Kalman Filter provides a closed form solution for the problem considered. Note that there are number variants of Kalman filter that may be used for this purpose: Extended Kalman Filter, Unscented Kalman Filter, and Information Filter.

2.2.3 History of Mobile Robot Localization

The literature review starts with dead –reckoning systems. A number of papers are enlisted to highlight state of the art of this group

In (Bonnifait 1999), a 3Dlocalization algorithm is developed for robotics application which moves slowly, and the dynamic and acceleration of the robot is ignored. This study aims the civil engineering applications which require

6DOF of the machine precisely i.e. Computer Integrated Road Construction. Thus use of inclinometers is proposed for attitude angle measurements. The use of inclinometers along with odometer is discussed and it is concluded that the long term drift of the relative positioning system must be corrected by an absolute positioning system. For this purpose relative positioning system is augmented with artificial beacon navigation. Artificial beacons located at known positions are surveyed by a camera and the geometric information is computed, and the unavoidable divergence in relative positioning system is corrected by using this information. Having described the theory, the system is first tested in a virtual environment, sımuCIRC a specialized Computer Integrated Road Construction algorithm developed in MATLAB Simulink model. The simulation results are later used for tuning of Extended Kalman Filter that is used in sensor fusion. Real experiments are performed in field. Indeed this study is a good example which selects one method from two basic localization methods group (*relative localization and absolute localization*). Hence it may be classified under the *Odometer/Relative Measurement Group* or *Active Beacons /Absolute Measurements Group*.

In (Ushimi 2003) an odometer system is developed to estimate the position and orientation of Omni-directional vehicles.

In (Meng 2003) the position and orientation of the robot is computed with the odometer data. A novel method for wheel slippage is developed based on the ratios of two wheel incremental distances.

In (Tsai 2005) a hybrid localization method is described. A new indoor 3D posture determination system of the robot based both on inertial and ultrasonic measurements, is developed. This posture tracking system is expected to be suitable for every mobile vehicle traveling over uneven indoor terrain. The method is expected to address the problem of the 3D posture determination for a mobile robot in the absence of wheel encoder or odometer information. So it consists of inertial navigation unit and ultrasonic sensor where inertial

navigation sensor, i.e. gyroscope, accelerometers, provides angular and linear acceleration. Since the integration of these measurements gives the position data, this inertial navigation unit is self-contained. However the inertial navigation unit tends to diverge in long runs, be cause of the integration operation on the raw sensor data. Hence the output of this unit must be corrected frequently. For this purpose a 3D ultrasonic localization system is used with inertial navigation unit to reduce localization error further. The inertial navigation system together with the ultrasonic system is capable of building an alternative integrated navigation system for three-dimensional vehicle navigation over uneven indoor terrain. The proposed INS system has been shown capable of providing low cost, high-precision posture information for the vehicle traveling over short distances. Through experimental results over the uneven terrain, the INS/Ultrasonic system has been proven to be capable of having position accuracy of less than 1 cm and orientation accuracy of less than 1◦.

In (Ernest 2004) a method of combining the GPS and Inertial Navigation System (INS) is devised. INS provides data when GPS is not available. GPS corrects the long term drift of INS. The sensor fusion is done by Kalman Filter. The method specifically is designed for trains.

In this subsection studies about the absolute localization methods are referred; Magnetic Compasses, Active Beacons, Landmark Navigation, Model Matching.

In this study (Suksakulchai 2000), localization of an indoor mobile robot which is equipped with electronic compass is studied... Disadvantages of the magnetic compass in indoor applications are discussed; it is known that the magnetic inference disturbs the output of the magnetic compass due to nearby objects. In this work, this disturbance is used for localization. Before the real navigation is started, the robot takes tour and records the magnetic compass output, which is not necessarily correct as a signature of local region. In the

real test the magnetic compass data is matched the recorded signatures, thus the robot localizes itself. This information is proposed as an external reference to be used with other techniques. Lastly, artificial landmarks are placed in the environment to create original signature for the environment

In (Ojeda 2000) a commercial compass KVH C-100 is analyzed. Description of the compass is given and then various error types are discussed; absolute error and relative errors. The absolute error is further decomposed into two subgroups; error introduced by variation and error introduced by inclination. The relative errors are decomposed into magnetic inference, i.e. natural inference, sunspot, solar magnetic storm, man made inference, vehicles buildings, vibration. Several correction methods are discussed.

Active beacons are similar to landmark based navigation where trilateration or triangulation methods can be used to compute position of the robot with respect to the beacons located at known positions in the environment. (Bonnifait 1999, Margrit 1997) are typical application of active beacon navigation

Satellite navigation (Ochieng 2002) systems may be used along with multi-modal transport navigation requirements. In road transport, the global positioning system (GPS) is currently supporting a wide variety of in-car navigation. The performance of GPS has in the past been limited by the artificial degradation of the signal through the process of *selective availability* (SA). With SA operational, the instantaneous horizontal positional accuracy was 100 m 95% of the time. In the following script, several GPS applications are presented.

In (Spagnol 2005) a method is devised to combine a special derivative of GPS, DGPS with inertial navigation system (INS), because GPS is not available always. The algorithm is designed to reconstruct the trajectory of van.

In (Lahrech 2004) advantages and disadvantages of the GPS are discussed; it is stated that the GPS data must be augmented by some other method for

continuous navigation. In this study, odometer is chosen for this purpose. Later, the availability of a map is discussed, since in transportation vehicle is bounded to follow roads and then map is modeled as an extra sensor. All sensor data are fused with Kalman Filter.

Landmark navigation is one of the simplest and reliable methods to correct the localization error by using external reference sources provided by the features of the environment. In such systems, the robot uses its sensory data and internal descriptions, which represents its environment to estimate its location. The estimated location is verified by matching the external (landmark) features with the features stored in the robot's internal representation. Here, a number studies below that are classified under landmark navigation are surveyed.

In (Dulimarta 1999), an indoor mobile robot system with a self-localization capability is described. The proposed system described here uses "ceiling lights" and "door number plates" for self- localization. It is observed that a reliable detection and recognition of the landmark features plays a crucial role in the performance of the mobile robot system.

In (Margrit 1997), a method whose computation time is linear in the number of landmark present in the environment, is proposed. It is assumed that the landmark can be identified by the robot; the robot has sensors that identify landmarks and measure their bearings relative to each other. When the accurate landmark information is obtained the robot can find its position on a 2D map. The robot operates indoors and the objects like doors, picture, and fire extinguishers are supposed to be landmarks of the environment. The mobile robot senses the environment through a camera. Once the landmark is sensed and matched with the previous pictures taken, the position estimation search is started.

In (Ratner 2003) it is pointed out to the link between navigation and sensory system/data. It is mentioned that while biological systems exploits the information rich sensory data, electronic system suffer from low quality, poor

information content data. Hence, perception of plant with an information rich ultrasonic sensor is studied. The sensors provide range information about the object present in the environment. A method is developed to exploit this information. The robot and the method developed, is tested in outdoor. Research on navigation along concrete paths through a park using the edge of the path as a continuous landmark is reported. Robot measures its location relative to the path edge with an ultrasonic sensor and steers to follow a trajectory relative to the edge.

This paper (Howard 1999) describes the navigation system used by a small vision-guided robot that roams the corridors at the University of Melbourne. In this study, a probabilistic technique is implemented to localize the robot in a static environment. Simple odometer system is used to estimate robot position but the use of any sort of global localization technique, i.e. GPS, DGPS, or magnetic compasses, is excluded. It is stated that the environment is not necessarily modified either. An implicit assumption is made; the robot can identify the landmarks. However the method requires that the robot navigates in the environment before it localizes itself for the first time. This technique is devised for indoor navigation.

In (Oussalah 1997) landmark based navigation of a mobile robot is studied. The landmarks are designed specifically for easy detection with infrared led on them. The sensor readings are modeled as possibility distribution.

In (Lin 1997) a method is devised in which a camera detects the artificial landmarks designed for localization purpose. A modified version of Hough Transform is used to measure the properties of landmark in the image plane. The system is tested on a robot.

In this part, localization methods which computes metric localization based on map matching are focused on. These methods can be divided into two groups depending on the representation of the reference map: Set of points (raw map), Set of geometric features (geometric map).

The first group includes algorithms performing map matching by using either the points captured by the sensor device or occupancy grids, without any geometrical assumption on these data. A common feature of these approaches is that they can be robust to sensor noise, ambiguous situations, and partial model descriptions. However, map matching tends to be computationally expensive, and in some cases these methods require heavy optimizations to implement an effective real-time localization on a mobile robot.

The second group of methods makes use of geometric features instead of raw points. Therefore, they require a preprocessing step in order to extract features (or natural landmarks) from sensor data. Most of these methods deal with lines, segments, corners, and the reference map is thus represented by a set of these features. These methods are usually more robust to noise given by unknown objects, as long as unknown objects do not affect feature recognition and reconstruction. However, their main drawback is that they rely on availability of features in the environment and on ability of the robot's perception system to detect them.

In (Kosecka 2005) a probabilistic scheme is described to localize an indoor mobile robot globally. While the robot is exploring, the environment is divided into sub locations and described by a set of scale-invariant features. The final environment is represented by a set of images and extracted features are matched with environment model using a standard voting mechanism. The localization is done in two stages. In the first stage, the current view is compared with pictures taken previously and portioned into individual locations. This comparison is not trouble free, be cause the algorithm must be robust to pose, illumination, scaling and partial occlusion. Therefore, individual locations that are to be used in the first stage of localization are represented by a set scale invariant key points. In the second stage, a neighborhood relations captured by a Hidden Markov Model is used to solve problem that arises due to dynamic changes in the environment. After the global localization is done, the relative pose of the mobile robot is computed.

In (Iocchi 2002), the importance of localization and different representation of sensory data is discussed. Then a localization method, called Hough Localization, is presented. It is reasoned that the method is called Hough Localization be cause the map matching is done in the Hough domain, not in the reference frame of the robot. This method can be summarized as: sensory data is represented by a set of lines and circles by Hough Transformation, and then the sensory data is matched with the map in Hough domain. This localization method is intended to be used with an odometer that provides the relative position estimate of robot base and any sort range finders, i.e. sonar, laser range finders, that measure the distance to objects in the environment. The basic assumption must be justified to use the method effectively; good position estimate and polygonal environment. The superior aspects of this method are; robustness to dynamic environment, linear computation time in the size of sensor data and number of reference lines. Several experiments in indoor environment have been done successfully.

In (Amato 2000), a new localization method using range sensor data i.e. distance measurements which is based on simple geometric properties of the environment, is proposed. In many common situations, information regarding the environment is provided a priori for path planning. During preprocessing, the workspace is partitioned into sectors using simple visibility computations, and a small identifying label is computed for each sector. The localizer analyzes the range sensor readings and extracts characteristic points, which are compared with the pre-computed sector labels to localize the robot, first to a sector, and then to a particular configuration within that sector. This two step process is computationally very simple, and allows precise localization without any landmarks/beacons.

In (Weber 2002), two types of maps; metric and topological are discussed. The advantages and disadvantages of both map types are presented; the metric maps are precise and robust to noise and sensor measurement errors, however their computational cost is high and in large-scale environment where mobile robot

position estimate gets worse and worse in long term navigation new measurement cannot be integrated into the map consistently. Topological maps are simplified form of the environment in the form graphs. They are more robust to odometer errors in large-scale environments than metric maps. Path planning and the integration of metric, non-metric data are easier. The method proposed in this study is a hybrid scheme; general position related to a certain region is expressed in topological map; where as the alignment of the current scan to stored scans is done in metric map domain. The generation of alignments depends on similarities between the current scan and the reference scans. The resulting multiple position hypotheses are tracked and evaluated in a hybrid topological/metric world model by a Bayesian approach. This probabilistic technique is especially designed to integrate position information from different sources, e.g. laser scanners, computer vision, etc.

In (Guivant 2002), a number of articles are reviewed and it is concluded that one of the main challenges in SLAM algorithm is that the computational cost is proportional to $\sim O(N^2)$ where N is the n number of landmarks in the environment. Hence, the SLAM problem is attacked to reduce the computational time at a level, proportional to$\sim N$, the computational time is linear in the number landmarks present in robot world. The following problem is addressed; when the robot cannot observe any landmarks in considerable time period. Aided SLAM algorithms are presented that incorporate absolute information in a consistent manner so they augment the SLAM method with a GPS data. Finally, the test results of the robot that operates in an unstructured outdoor environment, is presented.

The CyCab (Pradalier 2004) robot is the size of a golf-cab capable of attaining up to 30 km/h. Its “natural” environment is the car-park area of the INRIA Rhone-Alpes (about 10,000m2). For localization purposes, the detection of natural features in the environment is not preferred, since such detection is often subject to failure and not very accurate. So, in order to ensure reliability, it is decided to install artificial landmarks in the environment. These landmarks

have to be detected easily and accurately, and they should be identified with a reasonable computation effort. Landmarks; cylinder covered with reflector sheets, are specially designed for the Sick laser range finder. This led to the use of SLAM methods. It consists of building a map of features uncorrelated with the robot state. These features are, for instance, the distance between landmarks or angles between three of them.

In (Lee 2002), a reliable position estimation method of the indoor service robot is proposed. The service robot is a wheeled mobile manipulator which navigates in office buildings. Its localization method is a map-matching scheme using scanned range data, without using any artificial landmark. The proposed algorithm can provide solutions for both a global localization problem and a local position tracking. A probabilistic position estimation scheme is designed based on MCL (Monte Carlo localization). Two measure functions are developed for computing positional probabilities. The robot automatically decides whether it uses geometric pattern matching (i.e. walls, pillars) by Hough transform. The proposed scheme shows reliable performance in both polygonal environments and non-polygonal environments even there are many obstacles. Experimental results demonstrate the validity and feasibility of the proposed localization algorithm for the service robot to navigate in an office building, using the natural environmental characteristics.

Most car navigation systems estimate the car position by dead-reckoning and the Global Positioning System (GPS). However, because of the unknown GPS noise, the estimated position has an undesirable error. To solve this problem, a map-matching method is introduced in (Kim 2001), which uses a digital road map to correct the position error. In this paper, a novel adaptive-fuzzy-network-based –measure algorithm is proposed, which can find the exact road on which a car moves. The measure algorithm is easy to calculate, and calculation time does not increase exponentially with increase in the number of functions. For these experiments, a car navigation system is implemented with a small number of sensors.

Robust real-time vehicle positioning is a critical requirement for many vehicle technologies. However, no standalone positioning system is capable of providing uninterrupted and accurate vehicular navigation data in all environments. This paper (Basnayake 2005), presents a portable vehicular navigation system that combines high-sensitivity GPS, inertial sensors and map-matching techniques to provide uninterrupted vehicular navigation information in even the harshest environments for individual sensor technologies. This system eliminates the need for complex initialization routines required for high-end inertial sensors, making it a truly portable system. The map-matching component establishes a link between the vehicle and spatial information useful for many more vehicle technologies such as route guidance and location-based services. The system performance is evaluated and presented in an urban centre with extreme levels of GPS signal degradation.

In (Burgard 1996) the position tracking of the mobile robot in known environment is studied, the environment is modeled by a map or a CAD model. The robot's current position is modeled by a discrete probability function defined over the virtual grids on the environment; the probability is computed by matching the measurements with the map of environment. The position tracking of a mobile robot can considered to be a special class of the absolute positioning, since the search space is restricted to search space over the robot predicted position whereas in global positioning problem the whole space must be searched. This study is an extension of the previous study (Burgard 1996) on position probability grids. In the previous study, it has been shown, the method, position probability grids, is robust against; uncertain information, ambiguities, integration of sensor readings from different types of sensor. This method is found to be superior to Kalman filter. The fundamental difference between two methods lies how they model the probability distribution function. The previous one is a discrete approximation, whereas the latter one, Kalman filter, approximates the overall distribution by a Gaussian density function. It is stated that there are situations which cannot be represented by Gaussian density

function. The only precondition for this method is the availability of environment model/map.

In (Anousaki 1999) the simultaneous localization and map building problem is attacked with a high level representation of the raw sensor data. The mobile robot senses the environment with cheap ultrasonic sensor by which the mapping is done. The local world is represented with occupancy grid. The robot world constructed with line segments which are extracted from sensor data either by Hough transformation or clustering algorithm. The sensor data fusion is done by an Extended Kalman Filter for localization which is then used in map building. The distinct properties of this method developed in this is; the local world is composed of line segments, extracted for the raw sensor data by combination of Hough transformation and clustering algorithm

In (Zalama 2002), the problem of building a map of the environment from laser range finder data is studied. Having obtained a number of local maps the maps are segmented, and then the maps are matched to find a unique global map of the local world

In (Lu 1997) registration of the laser scan data to global map consistently, is studied. It is discussed that averaging the data or the use of Kalman Filter is a choice for register however it would diverge eventually. It is stated that in order to solve the inconsistency, any error detected must be chased back to previous local data obtained so far. A hybrid method consists of relation based and location based approach, used. In the first stage the pose estimate of the mobile robot and the scan data is integrated by using the uncertain odometer data. In the second stage the relations are converted in to local frame position by optimization.

In (Burgard 1999) the result of an experimental study with a mobile robot that operates in highly populated dynamic environment is published. The robot is tested in Deutsches Museum Bonn to guide the visitors in the museum. In this study several aspects of the mobile robotics; localization, mapping, obstacle

avoidance, path planning are examined. The emphasis is placed on the software architecture of the mobile robotics, which is distributed control architecture both in hardware design and software implementation. The mobile robot algorithms are based on probabilistic approaches developed in previous studies of the authors. The main limitation of the approach is the requirement that a map of the environment must be given to the robot before starting its guide.

Particle Filter is considered to be a distinct approach than Kalman filter. It allows accounting highly nonlinear systems. Hence a separate thread is started for this filter. Particle Filter is a power technique to deal with nonlinear non-Gaussian processes. It approximates complex PDF with random samples. (Gordon 1993) introduced the general framework of particle filter for target tracking. In mobile robotics research this filter is used to localize a robot and the method is called Monte Carlo Localization (Thrun 2001). (Thrun 2001) presented the results of particle filter in mobile robot localization. Although this filter is powerful, there are a number of issues that must be handled. The very first one is the computational cost, a large number of particles are necessary to represent the true shape of complex PDF. (Abdallah 2007) deals with the computational complexity. (Fox 2003) addresses the efficiency; the number of particles is adjusted so that the approximation error is minimized. (Kwok 2004) studied the real-time implementation of particle filter. Second, after a few iterations, most the particles have almost zero weights and only few particles contribute to the simulation, this is known as particle degeneracy. There are number of remedies for this deficiency; choose large number of particle, choose a well designed proposal distribution or do resample when necessary (Doucet 2001). Other problems are when the process noise is small, the particles collapse to limited area, and this is called particle depletion (Doucet 2001). In spite of these difficulties Particle Filter has found many applications in mobile robotics research (Fox 2003, Thrun 2001). Particle Filter which can cope with non-linear non-Gaussian process models is certainly a powerful approach to global localization. However, note that computational complexity which is proportional to number of particles is the main problem.

The Markov Localization has been introduced to the mobile robotics community by (Fox 1998). Markov localization precedes Particle filter, so it is considered as a separate issue. Markov Localization divides the state space of the problem into regular grids and evaluates the complex PDF on these grid points. Markov Localization can produce precise solution if the state space is divided into fine grids but the computational cost would be high, because all the grid points must be processed irrespective of the importance of those grid points. It is well known after the transient period the contribution of most grid points will be insignificant and only a small number of grid points are necessary for describing the PDF of the robot states. Although particle filter and Markov Localization suffers from computational cost, Note that particle filter offers more flexibility to adjust and direct the limited computational power to valuable regions which is likelihood is high. Thereby it is more efficient and accurate.

At the start of survey, it is stated that the Extended Kalman Filter is the right tool for multi-sensor fusion. Hence the rest of this section is devoted to typical papers that address the use of Extended Kalman filter in localization, explicitly.

(Leonard 1991) studied the localization of mobile robots equipped with sonar sensors. It is assumed that map of the geometric beacons in the environment is given a priori. Extended Kalman Filter (EKF) is employed to localize the mobile robot which starts from a known location. The EKF operates in the following cyclic manner: First it predicts the position of the mobile robot by using the process model, control input and posterior PDF of the mobile robot position in the previous time step. Then it computes the predicted observation based on the predicted mobile robot position and the map of the geometric beacons. When the real observations are received the measurement innovations and the covariance of the measurement innovation is computed. These innovations and innovation covariance matrix are used to setup a validation gate to match the real observations with the beacons in the map. The matched observations are augmented to form the composite observations. In the update

stage the composite observations measurement model are used to compute the posterior PDF of the robot position from the prior PDF of the robot position. The study investigates the use of geometric beacons instead of artificial beacons. Two experiments are performed for proof. In the first experiment a robot with point kinematics and rotating sonar is tested, in the second experiment a robot with a ring of six fixed sonar are tested.

(Giulio 2007) studied adaptive Kalman Filtering for GPS-based robot localization. In order to run Kalman Filter, the process and measurement noise must be determined experimentally before the filter starts. However the noise characteristics may not reflect the truth at all times. For instance in mobile robot operations; an on the spot turn or rugged surfaces or the objects in the environment may alter the noise characteristics. In this study the measurement innovations are used as indicator of filter divergence. If the filter runs properly the residuals must stay white all time, so Divergence Indicator is defined by using the measurement residuals and one input one output fuzzy inference is setup where the input is the Divergence indicator and the output is the scale factor. The scale factor shapes the prior covariance matrix. The method is tested on mobile robot equipped with GPS based localization system.

(Jetto 1999) proposed a fuzzy logic adapted Kalman Filter. It is pointed out to the fact that the process and measurement noise has big impact on the filter performance. Thus a proper setting of the noise characteristics of measurement and process models is vital. In the proposed scheme an EKF is used to fuse the odometer and sonar sensors. The environment is represented with planar surfaces normal the operational plane of the mobile robot. The statistical measurement characteristics of the sonar's arrays are assumed to be independent from each other and the noise characteristics is known up to a multiplier. The objective is to guess this unknown multiplier in an adaptive scheme. This is achieved by using fuzzy logic. It is observed that as the filter runs normally the measurement innovations must stay zero mean white noise. Any deviation from the ideal case signals the filter divergence. Hence this fact

is used to change the unknown multiplier in an adaptive fashion. If the measurement innovations are within certain limits the original value of the multiplier is used, if the measurement innovations are very close to ideal the unknown multiplier is set to smaller values, since measurements are more accurate. If the measurement innovations exceed certain threshold, the unknown multiplier set to higher values, because the measurements are coarse.

(Ashokaraj 2003) proposed an adaptive EKF for the localization of mobile robot. Several types of sensors are discussed: the odometer provides accurate data when calibrated and it is inexpensive. However, the odometer gives incremental displacements and errors tend to accumulate in time. Inertial sensors, accelerometers, gyroscopes, provide non-radiating, unjammable data. The accelerometers reads the accelerations, once they are integrated gives the velocity and displacement, gyroscopes gives incremental angular displacement, and compass reads the absolute heading. These sensors are not affected by environment. In this study, multiple low cost sensors, accelerometer, gyroscopes, odometer and compass are used together, which suggest that there are overlapping information for heading and displacement. This redundancy can be used to have better estimate of the states of the robot by using EKF. However it is well known that process and measurement noise characteristics are important for a proper filter run. The estimation of process noise is the most difficult one. So, in this study the process noise is adaptively estimated based on the Matrix Ricatti Equation in order to prevent the filter from divergence.

(Kim 2007) developed a hybrid strategy for outdoor mobile robot navigation. The robot is localized by means of EKF which fuses data coming from odometer, GPS, and a heading sensor. The robot has two mode autonomous and teleoperated.

(Von der Hardt 1996) developed a dead reckoning system for localization. The EKF is used to fuse the odometer, compass, and gyroscope data. The redundant information and the measurement residuals are used to detect misreading.

Methods are given to calibrate odometer and gyroscope and a new method is devised to calibrate the compass. The compass limits the unbounded growth of heading error and gyroscope gives redundant angular rate in addition to the odometer, thus the proposed system delays the need for an absolute measurement. It is stated that the system inevitably needs an absolute measurement when the system uncertainty increases beyond acceptable limits.

(Allesandri 1997) developed a dynamic model of the nonholonomic mobile robot model. The rotational and translational dynamic equation of the wheels, odometer model and the error model of the gyroscope is included in the process model. The odometer and gyroscope readings are considered as observations. By using these models an EKF is setup to estimate the position of the mobile robot.

(Ippoliti 2005) investigates the use of odometer and fiber optic gyroscope. In this study, first the odometer is used with simple integration technique to estimate robot position. A second implementation incorporates the fiber optic gyroscopes as the heading sensor. Lastly, odometer and fiber optic gyroscope is used with EKF to estimate the robot position. It is stated that the use of gyroscope gives better results than pure odometer and simple integration, be cause it is the heading that is most important. Hence, EKF which fuse the odometer and fiber optic gyroscope readings gives the best result.

(Barshan 1993) investigates the error characteristics of inertial navigation systems; a gyroscope and accelerometers. First, the error model for the accelerometers and gyroscope are obtained through extensive testing. Then process and measurement are constructed for EKF. The measurement model includes the accelerations along the three axes of Cartesian coordinates and the angular rate about the three axes. The plant model includes the position, velocity, acceleration and the error model along three axis and angle and derivatives of the angle up to order three and the corresponding error model of the gyroscope. The error models are tested in the following situations: the

performance gyroscope is measured against system with no error model, the accelerometer is tested in a simple experiment where thee robot accelerated and decelerated in short distances. It is stated that true error models are necessary to prevent the EKF from divergence. Nevertheless the current accuracy of inertial navigation system requires resetting by absolute measurement at regular time intervals.

(Hakyoung 2001), first study the error characteristic of fiber optic gyroscope. A precise calibration method is devised for this purpose. The fiber optic gyroscope and odometer are used with EKF for mobile robot localization. Experimental results are given for with/without calibration and/or with/without EKF.

(Tsai 1998) proposes an EKF framework to fuse the multi sensor; Dead-Reckoning, magnetic compass, gyroscope and odometer with ultrasonic measurement system. A voting scheme is proposed to ignore the unwanted measurement before merging the sensor readings. So the wheel slippage or magnetic inference can be eliminated at the sensory level.

(Qi 2008) studied adaptive Unscented Kalman Filter for mobile robots. Among all the parameter that affects the filter performance, the important ones are process and measurement noise. The measurement noise is generally well known, but the process noise is ambiguous and it changes in time. Therefore, this study focuses on adaptive estimation of the process noise by optimizing a cost function. The cost function setup based on the fact that measurement innovation must stay white noise. Since the measurement innovation covariance is diagonal, the cost that is to be minimized is the sum of the square of real measurements and the predicted measurements. Gradient descent optimization technique is employed successfully for estimation of the process noise of Adaptive Unscented Kalman Filter.

(Jetto 1999), developed an Adaptive Extended Kalman Filter. Both the process and measurement noise characteristics are optimized online. The Adaptive

EKF fuses the sonar and odometer readings where a planar world composed of lines are known a priori. The experimental test is made with Labmate, the process model of the robot is assumed to the kinematic model of unicycle vehicle.

2.2.4 Summary of Localization Techniques

Having surveyed a number studies, some positioning techniques and advanced algorithms have been encountered which are used by different authors, indifferent studies. The localization techniques are enlisted in Literature Survey Chapter. Therefore the list is not rewritten for the sake of brevity.[1] The advanced algorithms for localization are:

- ✓ Bayesian Inference
- ✓ Probability Theory
- ✓ Extended Kalman Filter
- ✓ Markov Localization
- ✓ Monte Carlo Localization

Two different kinds of mobile robot localization exist: relative and absolute. The first one is based on the data provided by sensors measuring the dynamics of variables internal to the vehicle; absolute localization requires sensors measuring some parameters of the environment in which the robot is operating. The mapping of the environment is a must for an autonomous system for auxiliary tasks. The measurement systems for mobile robot localization only based on relative or dead-reckoning methods, such as encoders, gyroscopes, ultrasounds, etc., have the considerable advantage of being totally self-contained inside the robot, relatively simple to use, and able to guarantee a high data rate. A drawback of these systems is that they integrate the relative increments and the localization errors may considerably grow over time if appropriate sensor-fusion algorithms are not used. Typical internal sensors are optical incremental encoders which are attached to the axis of the driving wheels or to the steering axis of the vehicle. At each sampling instant the

[1] Here the studies in literature are not classified according to the given convention again as the literature survey is devised accordingly

position is estimated on the basis of the encoder increments along the sampling interval. A drawback of this method is that the errors of each measure are summed up. This heavily degrades the position and orientation estimates of the vehicle, especially for long and winding trajectories. Other typical internal sensors are gyroscopes and accelerometers which provide angular rate information and velocity rate information, respectively. The information provided by these inertial sensors must be integrated to obtain absolute estimates of orientation, position, and velocity. Therefore, like for the odometers, even small errors in the singular measure may give rise to unbounded errors in the absolute measure. The problem of a continuous growth in the integrated measurement error can be overcome by periodically correcting the internal measures with the data provided by absolute sensors like sonar, laser, GPS, and vision systems. To reduce the frequency of these costly corrections it is necessary to improve the performance of the internal sensors.

GPS is not very accurate due to *selective availability* noise. The next choice DGPS has better resolution but it could be obscured in unusual times i.e. war. Neither the GPS nor the DGPS data is sufficient to navigate in unknown environments. Because they only provide information about the position of the object, they don't provide information about the nearby objects, so the robot can not rely on GPS data to affect on the environment. The use of a map with the GPS data seems to be a possible solution. However maps are static and very hard to build and maintain. They don't include every detail of the environment, so a mobile robot in an environment cannot count on a precompiled digital map.

The active beacons are not a good solution for outdoor environment because they must be installed at known position and at least three of them must be observable. This requires the manipulation of the environment.

Landmark navigation and localization seems to the most suitable method. However it suffers from the problem as that of the active beacons technique.

Most of the current object recognition methods require an object database to match with landmarks and/or the modification of the objects to be recognized easily. Visual methods are available to recognize the objects; however the identifying OOI, *Object of Interest*, is very difficult. In case it is achieved the available methods are generally designed for 2D objects or 3D objects recognized under the *weak projection method* assumption. These problems will couple with unstructured environment and scattered object in the background.

None of these methods mentioned above provides the autonomous mobile robot with means of localizing itself by self-contained techniques. It is apparent that an autonomous mobile robot that operates, in an unknown territory, on its own must be self-contained.

Recently, the simultaneous localization and map building problem, SLAM, has also been deeply investigated for increasing the autonomy of navigation of mobile robots. The idea of developing a mobile robot that can build a map of its environment, while simultaneously using that map to localize itself, promises to allow this vehicle to operate autonomously for a long period of time in unknown environments. The next subchapter investigates SLAM applications

2.3 Survey of Solutions to SLAM Problem

2.3.1 Definition of SLAM Problem

An autonomous mobile robot which is placed in an unexplored environment at an unknown location is asked to map the environment and localize itself with respect to best the map acquired incrementally. This problem is called **S**imultaneous **L**ocalization **A**nd **M**apping (**SLAM**) problem in mobile robotics literature. The first formal solution to SLAM problem which is based on *EKF* is proposed by the seminal paper of (Randall 1990). It has been recognized that *SLAM* problem is a milestone in developing a fully autonomous mobile robot. SLAM has found many applications in different field ranging from indoor

environments to outdoor environments, undersea operations and even airborne systems. Although SLAM is considered to be fully understood and to be a solved problem, there are still challenging issues which describe direction for future studies i.e. computational complexity, data association and rich environment representation. It was recognized that the correlations between the landmarks are crucial for the solution and convergence of SLAM problem and the more these correlations grow the better the solution is. Indeed, there is a high degree of correlations between estimates of the location of landmarks in a map and as vehicle navigates and makes new observations, these correlations grow.

2.3.2 Mathematical Statement of SLAM Problem

The first formal solution is provided by (Randall 1990). Assuming that the mobile robot operates in a planar world, the robot states can be represented with $\{x, y\}$ in Cartesian coordinates of the mobile robot reference point and the heading angle θ, measured *CCW* from the *x-axis*. Thus, the state/pose of the mobile robot can be represented with $\{x, y, \theta\}$ and assuming that the landmarks are point features their locations can be represented with $\{x, y\}$ in Cartesian coordinates. A navigating robot which takes relative measurements to the landmark is shown in Figure 2.1. In short the robot states can be represented in vector form:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2.5}$$

The state of a particular point landmark can be represented with

$$\boldsymbol{x}_l = \begin{bmatrix} x_l \\ y_l \end{bmatrix} \tag{2.6}$$

The solution can be described as: The robot starts exploring the environment, as it detect features, first it searches the recently updated map and look for a correspondence with the landmark detected so far and the current observations,

this is called *data association*. *Data association* is indeed and an effort to find the hidden states that generates the so called current observation. *Data association* is one of the critical issues in *SLAM* problem and is responsible for most of the filter divergence problem. This issue, *data association*, will be discussed in detail later. If a correspondence is found it updates the full covariance matrix thus it updates the **P**robability **D**ensity **F**unction (**PDF**) of the robot pose and *PDF* of the location of the landmarks in the map using an *EKF*. If it cannot find any correspondence between the current observations and the landmarks in the

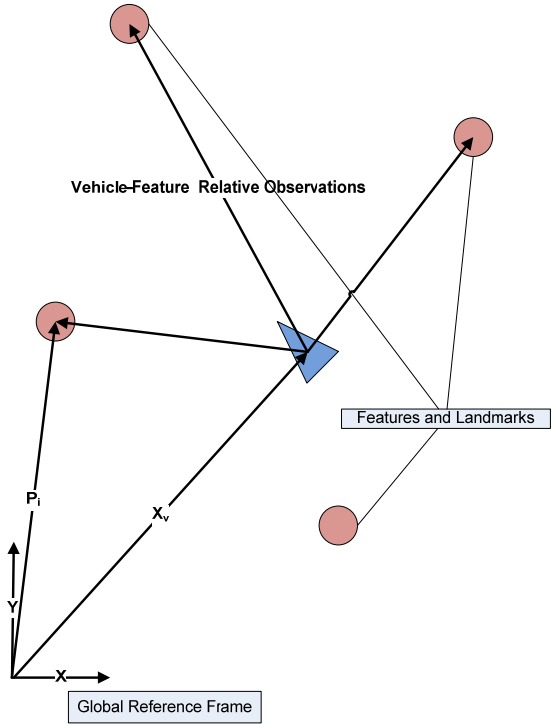


Figure 2.1 Vehicle taking relative measurements to environmental landmarks

most recent map, it augments the system states which are comprised of the states of the vehicle and the states of landmarks, with observed landmark states

and place it in the map. The system state made of the vehicle and landmark states can expressed as:

$$\boldsymbol{x}_{sys} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_l \end{bmatrix} \tag{2.7}$$

where $\boldsymbol{x}$ is defined with Equation (2.5) and

$$\boldsymbol{x}_l = \begin{bmatrix} \boldsymbol{x}_{l,1}^T & \boldsymbol{x}_{l,2}^T \ldots \ldots & \ldots \ldots \boldsymbol{x}_{l,m}^T \end{bmatrix}^T \tag{2.8}$$

In Equation (2.8), it is assumed that there are *m* landmarks in the maps. The aim is to *recursively compute the posterior PDF* which is defined as

$$P\left[\boldsymbol{x}, \boldsymbol{x}_{l,t} \middle| \boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}\right] \tag{2.9}$$

where $\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}$ are the history of observation from time *zero* to time *t* and the history robot control commands from time *zero* to time *t*, respectively. The *covariance matrix* of the system state is expressed as

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{vv} & \boldsymbol{P}_{vl} \\ \boldsymbol{P}_{lv} & \boldsymbol{P}_{ll} \end{bmatrix} \tag{2.10}$$

where $\boldsymbol{P}_{vv}$ and $\boldsymbol{P}_{ll}$ are the *covariance matrix* of the vehicle states and feature states in the map, respectively and $\boldsymbol{P}_{lv}$ is the *cross covariance matrix* between the vehicle states and the states of the landmarks in the map. It is this *covariance* that maintains the relation/update between vehicle and the landmarks. Note that landmarks are observed by the vehicle and located with respect to the current vehicle location.

In general, a recursive solution to SLAM problem is sought. By using Bayes Theorem, the joint posterior is computed by starting with an estimate for the distribution $P\left[\boldsymbol{x}_{t-1}, \boldsymbol{x}_{l,t-1} \middle| \boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t-1}\right]$ at time $t-1$, following the control input $\boldsymbol{u}_t$and observation $\boldsymbol{z}_t$. In order to compute this joint posterior, a state transition model and observation model are defined.

The observation model describes the probability of making an observation $\boldsymbol{z}_t$ when the vehicle location and landmark locations are known and is generally described in the form of

$$P(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{m}) \tag{2.11}$$

Generally, it is assumed that once the vehicle location and map are defined, given the map and the current vehicle state, observations are conditionally independent.

The motion model for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \tag{2.12}$$

The state transition is assumed to be first order Markov process in which the next state $\boldsymbol{x}_t$ depends only on the immediately preceding state $\boldsymbol{x}_{t-1}$ and the applied control input $\boldsymbol{u}_t$ and is indepent of both the observations and the map.

The SLAM algorithm is now implemented in a standard two-step recursive (sequential) prediction, update form:

Time-update

$$P(\boldsymbol{x}_t, \boldsymbol{m}|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0) = \int P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)\, P(\boldsymbol{x}_{t-1}, \boldsymbol{m}|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t-1}, \boldsymbol{x}_0) d\boldsymbol{x}_{t-1} \tag{2.13}$$

Measurement Update

$$P(\boldsymbol{x}_t, \boldsymbol{m}|\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0) = \frac{P(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{m}) P(\boldsymbol{x}_t, m|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0)}{P(\boldsymbol{z}_t|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0)} \tag{2.14}$$

Equation (2.13) and (2.14) provides a recursive procedure for calculating the joint posterior $P(\boldsymbol{x}_t, \boldsymbol{m}|\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0)$ for the robot state $\boldsymbol{x}_t$ and map $\boldsymbol{m}$ at a time $\boldsymbol{t}$ based on all observations $\boldsymbol{Z}_{0:t}$ and all control inputs upto and including

time $\boldsymbol{t}$. The general recursive solution is a function of a motion model $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$ and observation model $P(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{m})$. Note that problem of map building may be cast as computing the marginal conditional density $P(\boldsymbol{m}|\boldsymbol{X}_{0:t}, \boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}, \boldsymbol{x}_0)$. This assumes that the location of the vehicle $\boldsymbol{x}_t$ is known at all times. A map $\boldsymbol{m}$ is then constructed by integrating observations from different locations. Conversely, the localization problem may be cast as finding the marginal probability distribution $P(\boldsymbol{x}_t|\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}, \boldsymbol{m})$. It is assumed that map of landmark locations are known with certainty and objective is to estimate the vehicle location with respect to these landmarks.

2.3.3 The History of SLAM

The seminal paper (Randall 1990) presented the first formal solution to SLAM problem. The stochastic map is introduced to represent the spatial relationships between objects in the map. The methods which are required to build and update and read these relationships from the map are given. The EKF is used to estimate the mobile robot position and the spatial relationships between the landmarks and between the landmarks and mobile robots are given. The Dynamic Bayesian Network which illustrates a navigating robot is shown in Figure 2.2.

(Dissanayake 2001) made a literature review about SLAM. Then, a theoretical study about the existence of a solution to SLAM problem is conducted, and the convergence of SLAM method is studied. A proof that the estimated map converges monotonically to a relative map with zero uncertainty is then developed. It is then shown that the absolute accuracy of the map and the vehicle location reach to a lower bound defined only by the initial vehicle uncertainty. Together, these results show that it is possible for an autonomous vehicle to start in an unknown location in an unknown environment and, using relative observations only, to incrementally build a perfect map of the world and to compute simultaneously a bounded estimate of vehicle location. A proof of the following statements is published; as the number of observations increases the landmark observations become fully correlated, in the limit the

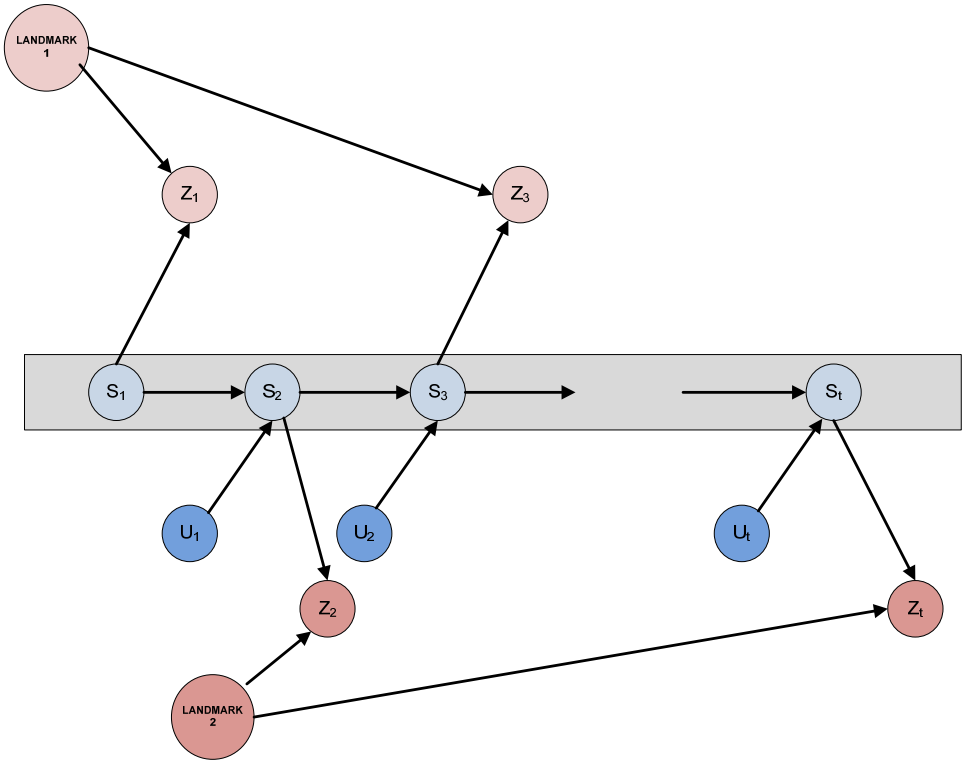


Figure 2.2 SLAM problem: The robot moves through poses $\boldsymbol{s_1}$ ….. $\boldsymbol{s_t}$ based on a sequence of controls, $\boldsymbol{u_1}$ ….. $\boldsymbol{u_t}$. As it moves, it observes nearby landmarks. At time *t = 1*, it observes landmark 1. The measurement is denoted by $\boldsymbol{z_1}$. At time t = 2, it observes the other landmark 2, and at time t = 3, it observes landmark 1 again.

covariance associated with a single a landmark location estimate is determined only by the initial covariance of the vehicle location estimate. After the theoretical study is completed, field experiments with a vehicle that has a millimeter –wave radar, are conducted. The convergence of EKF based SLAM methods by using linear process and a measurement model is proved. However the linear process and measurement models assumptions are sever and later these assumptions received critics. This study proves the following important results[2]

[2] This part is reprinted here not to distort the original statements.

- The entire structure of the SLAM problem critically depends on maintaining the complete knowledge of the cross correlation between landmark estimates. Minimizing or ignoring cross correlations is precisely contrary to the structure of problem.
- As the vehicle progresses through the environment the errors in the estimates of any pair landmarks become more and more correlated and indeed never become less correlated.
- In the limit, the errors in estimates of any pair landmarks become fully correlated. This means that given the exact location of any one landmark, the location of any other landmark in the map can also be determined with absolute certainty.
- As the vehicle moves through the environment taking observations of individual landmarks, the error in the estimates of the relative location between different landmarks reduces monotonically to the point where the map relative locations is known with absolute precision.
- As the map converges in the above manner, the error in the absolute location of every landmark reaches a lower bound determined only by the error existed when the first observation is made.

It is stated that a solution to SLAM problem does exist and a precise map can be obtained.

Recently, several improvements to the original SLAM algorithm are proposed. Several researchers attempt to cut the computational cost by exploiting the structure of SLAM problem.

(Guivant 2000), studied localization and map building by using the information form of Kalman Filter. Information Filter is quite attractive when a large number of landmarks are processed sequentially because it is not required to update the full covariance matrix of Kalman Filter. Both artificial beacons and

natural features are used as landmarks and experimental proof is presented for both cases. Laser range finder is used for sensing and a utility vehicle is used.

(Guivant 2001) proposed a *Compressed Extended Kalman Filter* (CEKF) for SLAM problem. The *sparseness of measurement model* and the fact that the mobile robot mostly operates in local regions, only detects a number of landmark which is much smaller than the total number of landmarks leads to the CEKF. The cost of CEKF is $\sim O(N_a^2)$ where $N_a$the number of landmarks in the local region is. The local maps are transferred to the global map in single iteration with a cost of single full SLAM iteration. As long as the mobile robot does not switch between local regions very often, CEKF is quite cost efficient. Experimental proofs are presented where the tree trunks are used as landmarks.

(Thrun 2004) introduced the Extended Information Filters. The *Sparse Extended Information Filter* (**SEIF**) achieves constant time in solving SLAM problem. The key points that is exploited in developing SEIF is that when the inverse of covariance matrix is computed ,*information matrix,* it is observed that only local off-diagonal terms are significant, these off-diagonal terms are like constraints between the landmarks. These structures suggest that the landmarks are locally connected; the geographically nearby landmarks are more correlated than landmarks which are distant from each other. Since the robot sensors have limited range, a set of landmarks, called active landmarks are linked to the mobile robot at the current time. Thus web like structure connecting the landmarks and mobile robot simplifies the computational cost of SEIF for the solution of SLAM problem. SEIF requires linear space for storage and updates can be performed in constant time regardless of the number landmarks in the map.

(Csorba 1997) developed the relative filter. It is well known that the Absolute Global Filter is computationally intractable. Be cause each time the vehicle makes an observations, it updates the full covariance matrix, including the cross correlations between landmark. In this study, the relative distance and

relative angle between the landmarks are stored, thus a relative map is developed. The relative distance between two landmarks and the angle between three landmarks can be measured without knowing the absolute location of the vehicle but the relative distance traveled during measurements, the vehicle displacement. These measurements will be uncorrelated to the previous displacement measurements. Hence a simple filter, called relative filter is developed which updates the information locally. This leads to a considerable computation time savings. However the relative filter is criticized for not taking into the cross correlations between landmarks and the original relative filter is considered to be suboptimal. It is known that geometrically close landmarks are correlated more than distant landmarks. In (Guivant 2000) it is shown that keeping this cross covariance terms is a must for the absolute convergence of EKF based SLAM methods. Therefore, suboptimal filters diverge eventually. Moreover, the relative map may be accurate but when then relative distances are represented in the global framework the local uncertainties propagate through the relative map network and a gross error may occur.

(Martinelli 2007) takes on completely different way. The distances and angle between point and corner landmarks which are invariant under translation and rotation are used in mapping. The computational cost is independent from the number of feature in the map. Since the estimation is local, it is not affected from the linearization of the process and measurement models. Indeed this approach was first developed by (Csorba et.al.) However in that study a suboptimal filter is used in the update stage which does not take into account the interdependencies between the invariants representing relationships between the landmarks in the map. This approach requires attention when the local invariants are transformed into a global map. Although the local invariants may be quite accurate when they are transformed into global map, the local inaccuracies may add up and the global map may turnout to be not very precise. Newman proposed the projection filter to transform the local invariants into global maps in that study a system of nonlinear equations are

solved. (Martinelli 2007) points out the fact that multiple observation for the same landmark in global coordinates may disagree with each other due to uncertainty in the local invariants. In order to solve this inconsistency, EKF based estimation is proposed. This approach, *Relative Map Approach to SLAM,* is quite different from the common approach, EKF based SLAM and only a limited number of papers are available in the robotics literature.

(Wang 2007) developed a decoupled solution to SLAM problem where state of vehicle augmented with the states of landmarks, is transformed into a new form. This new form has the landmark states independent of the vehicle state and the part includes the vehicle state and the connecting states. It is proved that this transformation has the effect of sparse extended information filter. Hence computational cost is reduced dramatically.

(Nguyen 2006) developed the orthogonal SLAM. In this study only the parallel and orthogonal lines are mapped. These types of works are common in indoor environment. This leads to computational efficiency.

Some researchers focused on data association problem. For instance FastSLAM proposed by (Montemerlo 2002, 2007) decomposes the SLAM problem into a localization problem and K independent landmark estimation problems conditioned around the robot pose estimates.

(Montemerlo 2002, 2007) developed FastSLAM in a series of paper. The approach based on the idea, conditional independence of landmark location estimation once the robot poses are known. This conditional independence can be visually shown with Dynamic Bayes Network. Thus the FastSLAM decomposes the SLAM problem into a localization problem and a landmark state estimation based on the so called localization problem. In FastSLAM the localization is performed by particle filters, M particle. Then the estimation of K landmarks requires K Kalman Filters for each particle representing the trajectory of the robot in time. So the naive implementation of FastSLAM is

*O(MK).*However the computation cost may reduced by a tree-based structure to *O(MlogK)*. It is demonstrated that FastSLAM can work with as many as 50,000 landmarks and even with unknown data association and unknown number of landmarks. The FastSLAM has been tested with *Victoria Park* data set.

(Eliazar 2003, 2004, 2006) developed the Distributed Particle SLAM (DP-SLAM). The starting point of this work is very similar to FastSLAM. The unique side of this study is that it maintains the robot pose and map in multiple particles. In this work the naïve approach to representing the robot pose and the maps in particles is discussed and a novel approach is developed to cut the computational cost. First the particles are represented in terms of ancestor trees; thereby it keeps a limited number of particles to represent the particle trajectory. Secondly it does not maintain a map for every particle but it maintains a single occupancy grid map in which each grid maintains a list of particles that access to the grid and change it. These modifications have made great computational cost cuts. It is stated that the complexity of DP SLAM is log quadratic in the number of particles and linear in the area swept by the laser.

(Masson 2003) studied mobile robot navigation and mapping in large environments and hybrid *architecture* of *Monte Carlo Filter* and EKF based SLAM is proposed. It is known that the original EKF based SLAM is prone to diverge to due *nonlinear process* and *measurement models*. Since these nonlinear equations are linearized to run the EKF, in large scale environments where the mobile robot may navigate without observing landmarks detected and placed in the map before, the uncertainty associated with the pose of the mobile robot may reaches to level where the mobile robot cannot perform data association. As the uncertainty of mobile robot pose takes higher values, the landmarks that are close to each are may be mistaken for each other. In this study, when the EKF fails, *Monte Carlo localization* (MCL) is started and it continues to run until the data association problem is solved then the system switches to EKF based SLAM again. The *Compressed Extended Kalman Filter*

(CEKF) is used in here. The authors demonstrate the robustness of the method with experimental studies.

(Estrada 2005) studied the hierarchical SLAM. Two level mapping is used, in the higher level a global map is maintained which is indeed a tree. The nodes of the tree refer to the local maps and the arcs represent the adjacency and metric information about the relative distances between local maps. In the lower level, local maps are used for environment representation. These local maps are initialized with vehicle location when the map is created and the vehicle path and the features are located with respect to this base reference. The local maps are created when the vehicle location uncertainty takes higher values, or new features cannot be associated with the mapped ones anymore. These local maps are guaranteed to be statistically independent suggesting that the covariance matrix of the adjacency information is block diagonal which offers great opportunity in term of computational cost. The unique side of this study, it imposes loop closing constraint when it is detected. It performs loop closing linear in the number of local maps exist in the loop. Since loop closing is certainly nonlinear due rotations, nonlinear optimization technique, Sequential Quadratic Programming (SQP), derived from the Kuhn-Tuker equations is employed. It is shown that similar results can be reached with Iterated Kalman Filter, by assuming zero observation error. Lastly the method is verified in environment of size 100x100 m^2. It is demonstrated that loop closing can be achieved in real time and loop closing constraints improves the mapping

(Neira 2001) studied the data association problem and proposed Joint compatibility test as an alternative to the well known nearest neighbor approach which does not take into account the correlations between measurements and assume that measurements are independent.

Stable landmark detection is difficult in outdoor environment. It is relatively easier to find natural landmarks in indoor environments such as corners and

walls. However, outdoor settings do not have so many distinct /structured features.

(Madhavan 2004, 2005) proposed to use the points of maximum curvature as natural landmarks in unstructured outdoor environments. It is easy to find natural landmarks in indoor environments such as corners, walls but outdoor settings does not have so many distinct features so curvature scale space method is develop to find the points of maximum curvature as robust, stable landmarks. These landmarks are invariant under rotation and translation so they can safely be used with odometer for localization.

Due to their computational cost, the SLAM methods cannot produce dense maps. Although sparse representation of the environment is sufficient for localization and mapping, it is not adequate for navigation.

(Guivant 2004) developed a hybrid metric map (HYMM). Due to computational cost SLAM methods cannot produce dense maps. Although sparse representation of the environment is sufficient for localization and mapping, it is not adequate for navigation. In this study the sparse maps of features, produced by solving SLAM is combined with occupancy grid (OG) maps to represent the environment more accurately. The features/landmarks in the environment are used to divide up the environment into a set of connected Local Triangular Regions (LTR). Since the geographically close landmarks are highly correlated, though they may have large uncertainty in the global frame, these LTR are solid regions on which several properties can be represented in the form OG maps. These OG maps are then used for path planning, reactive navigation and optimal exploration or they can be used to avoid the filter divergence by maintaining a bounded uncertainty for the vehicle pose, using information maps. These OG maps can also be used for data association where landmarks cannot be detected from a single vantage point. The Compressed Extended Kalman Filter (CEKF) and Monte Carlo Filters are used together to solve the SLAM problem.

(Nieto 2006) addresses the dense representation of environment. Due to computation burden, EKF based SLAM approach can represent the environment in sparse form with isolated landmarks. This sparse representation is proved to be sufficient for localization purposes but it is inadequate for other tasks such as navigation and other properties of environment e.g. occupancy, salient feature of environments such humidity concentration of chemical in the environments. In this study several observations are made about the SLAM: The landmarks observed at a time the vehicle displacement is small, are highly correlated and the relative distances and angles between these landmarks are independent from the rest of the map. This property enables to decouple the small regions from the rest of map and landmarks. Thus the feature based sparse landmark maps can be delimited to small Local Triangular Regions (LTR) in which a dense representation of the environment is possible. Since the local landmarks are highly correlated their relative distance and shape of the LTR are affected by the observations made at other time though the LTR may be rotated and translated the shape is maintained. The other properties of the environment are then represented in this LTR in Occupancy Grid or Sum of Gaussians form. In this study it is stated that the dense representation does not send any information to the EKF based SLAM and only receive information from he EKF based SLAM, Unidirectional Information flow (UIF). Hence it is demonstrated that the EKF based SLAM convergence and computational properties are the same irrespective of the dense map representation.

(Dissanayake 2002) developed a map management strategy for efficient computation. In this work the distant landmarks are safely deleted and later added when they are detected again. This leads to considerable cost reduction.

2.3.4 Summary of SLAM Techniques

SLAM is the process by which a mobile robot can build a map of the environment and at the same time use this map to compute its location. Majority of effort has aimed improving computational efficiency while maintaining consistent and accurate estimate of the mobile robot pose and

landmark locations. However there are also other researches subjects i.e. nonlinearity and data association and landmark characterization.

The computational complexity of SLAM in its naive form grows quadratically with the number of features in the map. For real-time implementation, this complexity is a limitation of SLAM methods. Many approaches have been developed to reduce this complexity i.e. the linear-time state augmentation, sparsification in information form, partitioned updates and sub-mapping methods. A second major issue is to be solved in implementation of SLAM methods is data association; associate observations of landmarks with landmarks held in the map. Incorrect association may result in catastrophic failure of the SLAM algorithm. Data association is indeed challenging when a mobile robot visits a place more than once i.e. returns to a previously visited region after a long voyage; the so-called loop closure problem. There are different data association methods used in SLAM. These are batch validation methods that exploit constraints inherent in the SLAM formulation, appearance-based methods and multi-hypothesis methods. A recent research interest in SLAM is to represent the environment in rich appearance based form and it is major contemporary research direction.

SLAM methods have now been understood well. Future research will focus on methods enabling large scale applications in natural environments and especially where GPS-like solutions are unavailable or unreliable; in urban canyon, under foliage, underwater or on remote planets.

2.4 A Discussion about SLAM and Localization

Localization of a mobile robot is vital to autonomous navigation, since the robot must know where it is in order to navigate safely and to plan paths or to act on the surrounding environment. Therefore this subject has been an active research field for the last two decades. Different approaches to control architecture and position control/navigation have been employed in the history of mobile robot. The mobile robotics research field has evolved in time through

different phases where different paradigms are dominant. In 80's the model based approaches were employed intensively. However it was evident that mobile robot models are not free of errors and they are not capable of modeling environmental vagueness. So, the operations of mobile robot designed with hardcoded model of itself and the environment are limited to controlled/engineered environments, besides they demand costly high precision actuators and sensors in order to limit the inherent noise characteristics. Although accurate model based approach to robotics is successful in industrial robotics field where the task and environment have clear definitions and the robot is asked to do no more than pre-specified actions in a repetitive manner, this approach was terminated in 90's. In early 90's a new paradigm had been started that would last late 90's. This paradigm is called reactive/behavior based robotics. According to this approach, the mobile robot is equipped with a number hardcoded rules and the robot acts in the environment responding to its sensory information at low levels without integrating and high level intelligence. The most successful implementation of this approach can be found in obstacle avoidance. As it is aforementioned, there is no high level intelligence in behavior based mobile robotics and everything solved at the sensory level. Although they are successful solving a number of problems and they are still being used as an integral part of an autonomous mobile robot, they cannot respond to the need of a fully autonomous mobile robot. In late 90's probabilistic methods appeared as a powerful technique to solve many problem researchers have faced for many years, since probabilistic approaches have the ability to define, propagate and update vague data, they are suitable for the inaccurate sensory and actuator information. They can handle the noisy environment and model vs. sensor information or model vs. environment mismatch in a convenient and natural way. They do not require exact models of robot itself, sensor model or actuator model or the model of the environment and they do not need high precision expensive sensor, all that is needed is an approximate model and the statistical properties of the sensors, actuator and the robot itself.

In its simple form the localization problem can be solved by integrating the odometer data once and deriving the heading angle from the kinematic model of the robot as well. A similar alternative is to use Inertial Navigation System in simple form; that is to integrate the accelerometer and gyroscope data twice to obtain the displacement and angular rate. However these simple looking integration techniques are likely to fail due to excessive sensor noise and inadequate system model and/or sensor model. Behavior based mobile robot can act on its own relying on its sensory information but they do not address the localization problem explicitly. However they are an important part of autonomous mobile robotics in navigation, i.e. obstacle avoidance. A true treatment of sensory information is to use Kalman Filter to integrate actions and measurements to the robot belief in its pose. Kalman Filter is successful in sensor fusion which fuses redundant sensory information and uses statistical properties of the measurements and actions so as to come up with optimal and better results. Kalman Filter is optimal in the sense that covariance matrix of the states are optimized in the least square sense. It uses two types of sensors the ones that measure the internal states and the ones that measure the environment. Both type of sensor must be used together because first group of sensor measures the internal parameters of the system and they cannot bound the covariance of the states without help of the latter group of sensory information. Latter group of sensors receives information about the environment, put it in other words they receive absolute information whereas first group of sensors receive incremental information. Some typical examples these sensors are: for instance; gyroscope and odometer may be classified into first group sensors whereas compass and laser range finder or ultrasonic sensors may be classified into the second group of sensors.

Kalman filter uses odometer and compass, odometer, compass and gyroscope and odometer and gyroscope to localize a mobile robot. Note that odometer provides necessary information to compute displacement and heading. If compass and/gyroscope is used together with the odometer, there is redundant information for the heading state. It should be stressed that heading is the most

important state in tracking the pose of a mobile robot since any small error in it causes large deviation in its location. Therefore, the redundant information from the gyroscope and/or compass is used to detect the sensors functioning abnormally and an optimal estimate of heading is computed by using the redundant information and Kalman Filter. Kalman filter, odometer, compass and/or gyroscope can be used to track the position of mobile robot but they cannot be used to localize a mobile robot from scratch, in other words they cannot be used to localize a mobile robot globally. Their main advantage is that they are self contained and independent of surrounding unless earth magnetic field is distorted by the nearby objects. Inertial Navigation System has the advantage that they cannot be jammed and is self-contained, independent of any external source.

A second implementation of Kalman Filter is to use odometer and low cost sensors i.e. ultrasonic sensor. Here the strategy is the same as the INS but with a different measurement model. It must be point out to the fact that here it is assumed implicitly; a model of the local world is available. This type of localization is better than previous Kalman applications be cause an absolute information about environment received through ultrasonic sensors are integrated to robot belief in the update stage of filter, this would certainly keep the localization error in certain error bound. Here the success of localization depends on the world model and its accuracy.

Localization is investigated with other probabilistic approaches that have emerged recently, for instance Markov localization, a grid based approach, or particle filter, a Sequential Monte Carlo methods. Indeed localization problem is studied alone, assuming that a model/map of the environment is available. This problem is assumed to be solved in static environments and quasi-static environments. A second difficult problem can be raised by dropping out the assumption of an available map; this problem is described as *Simultaneous Localization and Mapping* (SLAM) in robotics literature. This problem has been studied for the last ten years. The SLAM problem is tried with Kalman

Filter based approaches and they are quite successful in solution. However the time complexity of Kalman filter based solution $\sim O(N^2)$ in the number of features in the environment and Kalman Filter uses multi-dimensional Gaussian distribution to update and propagate the mean and the covariance matrix which is certainly unimodal, put it in other words, Kalman Filter based solution to SLAM cannot hold more than one hypothesis. Indeed the problem occurs when the Kalman Filter based SLAM methods fail in data association by using the validation gate procedure, this leads to a catastrophic failure of the filter and it cannot recover from failure.

As, now it is apparent that Kalman Filter based SLAM solution has some disadvantages and localization methods alone that assumes an available map of environment cannot be justified. In outdoor settings using SLAM methods to map the environment is infeasible. In the next section the scope of dissertation will be stated firmly. This statement will answer the localization problem in outdoor setting which is a strong prerequisite for autonomy.

2.5 Scope of Dissertation

In the light of previous qualitative discussion, the followings conclusion can be drawn: *The SLAM problem is difficult to solve and data association might be difficult in large scale environments. The map representation is sparse and is not very useful for several tasks a robot has to carry out but for localization. Position tracking needs absolute measurements to reset the filter otherwise it diverges. Map based localization is considered to be solved but drawing such a map is an open challenge*[3]. Therefore in this study a novel solution is proposed to tackle the global localization of outdoor mobile robots operating in urban settings, which can be classified into map based localization.

In this study, very large scale urban environments are considered and the mobile robot is localized globally. The size of the environment considered are about 1 million square meters which is far larger than a typical environment

[3] This text is reprinted here for the sake of completeness

considered in previous studies in the literature. It is evident that current SLAM methods are not suitable for such environments. If simultaneous mapping and localization is not to be used, remote sensing provides the right answer to finding maps of vast outdoor areas. In last decade the satellite pictures of most urban areas have become freely available on the internet. In this study the potential use of these images for outdoor mapping is exploited. Then generated maps are used in conjunction with localization methods to globally localize the mobile robot. The global localization claim can be justified by using large enough satellite images. By using the built-in cell phone infrastructure in a city, it is possible to localize one's location in a circular region the diameter which is generally about one kilometer. @Google Earth cell phone application is an example application which provides such a crude localization by making use of cell phone infrastructure. Hence it can be assumed that ability to localize a robot in an approximate area of one million square meters is sufficient for global localization when the cell phone infrastructure present in urban settings are utilized. Following this approach, the SLAM problem is avoided and localization is performed in mutually independent steps; first the map of the region of interest is obtained by making use of satellite images, second localization is achieved by using the maps obtained from these images. The scenario of the operating conditions of the mobile robot can summarized with the Global Urban Localization (GUL) problem:

"*A robot with a Wi-Fi enabled device (such as a laptop, a PDA or a cell phone) wakes up in an unfamiliar urban setting. In the mobile robot's foremost task is to determine its location. Luckily, a wireless Internet connection is detected in the area so that the IP address obtained through this connection can be used to identify this region. With the virtue of this wireless connection, the robot accesses a website (such as Google Earth) where the satellite images of the earth can be conveniently downloaded. By simultaneously going over the satellite images and checking out the environment using relevant on-board sensors, the robot can find common features that might pinpoint its location or at least prune the search space. Eventually, the robot performs global*

localization limited to urban settings. This problem will be referred to as Global Urban Localization (GUL) problem."

Provided that the assumptions (i.e. relevant satellite images are downloaded via a wireless connection) are satisfied, the effort to solve the presented problem raises two basic questions: Which sensors are suitable for this localization process? How should the satellite images be processed to match with local sensory data? This study tries to answer these questions and as a result proposes a novel technique suitable for the solution of the GUL problem. In this study, localization of an outdoor mobile robot is performed in two stages: First, the map of the environment is obtained using the satellite images that are freely available on the Internet (i.e. Google Earth). A hybrid approach employing Fuzzy C-Means (FCM) and Adaptive Neuro-Fuzzy Inference System (ANFIS) methodology is adapted to segment various elements in such images including buildings, forests, fields, and roads. Then, the Monte Carlo Localization (MCL) technique is used with these maps to find the pose of a mobile robot which senses around by a laser range finder.

Note that these assumptions; a wireless internet connection or an access to cell phone infrastructure may not be guaranteed always. During unusual times, this infrastructure may be destroyed. In order to cope with this situation, a satellite image database may be uploaded to the robot memory.

The main motivation of this dissertation is to elaborate this novel localization technique in which the localization task at hand is considerably simplified by processing satellite images so as to generate suitable digital maps. This study is to illustrate that when such maps are used in conjunction with a modest sensory data on the surroundings of a mobile robot (like laser range finder scans), even a simple (but yet effective) method like MCL can lead to very efficient localization of the robot. The pictorial description of dissertation organization is shown in Figure 2.3.

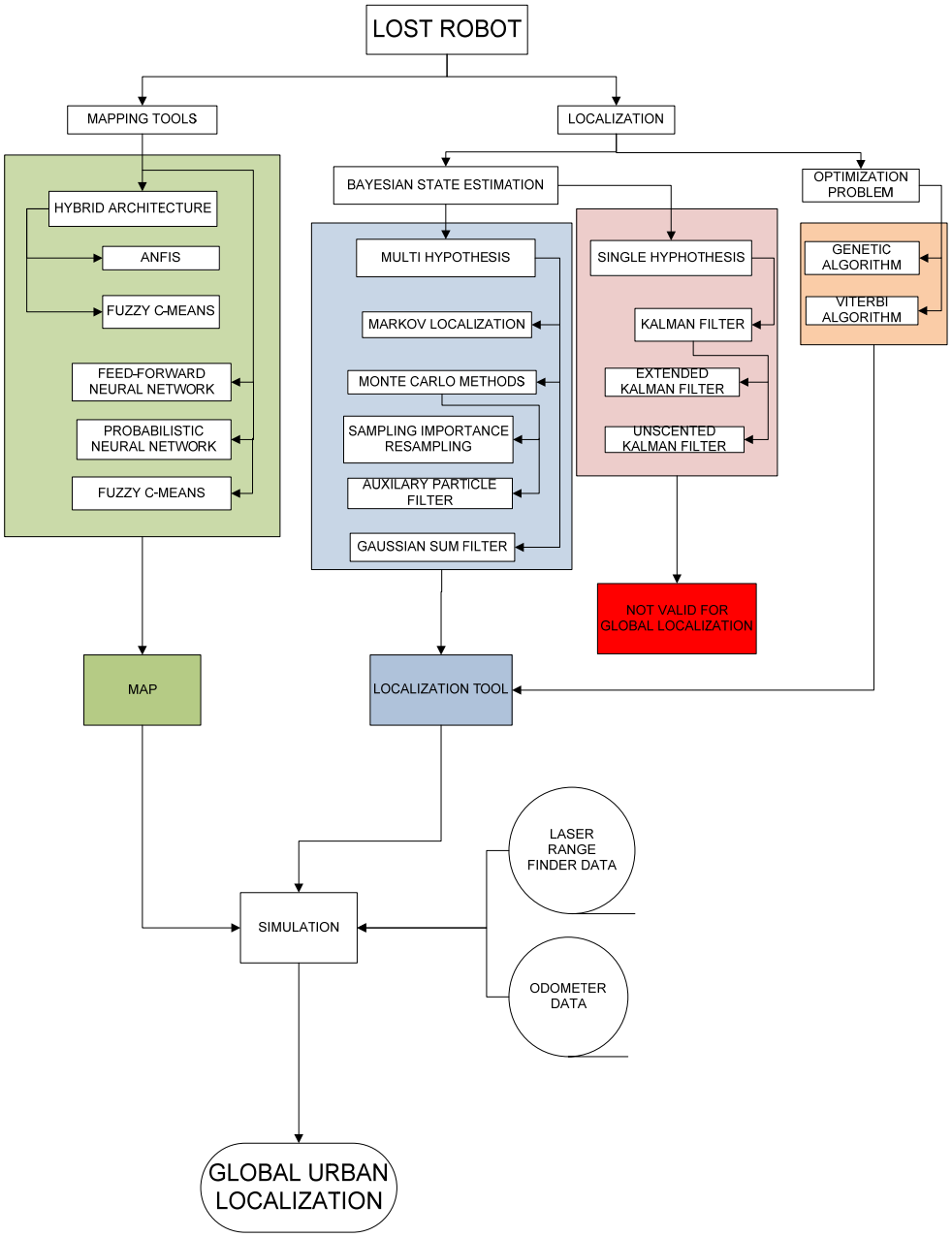


Figure 2.3 Flowchart of Global Urban Localization problem

CHAPTER 3

THEORY OF TRACKING PROBLEM

3.1 Introduction

This chapter reviews the general approach to tracking problem and it is compiled from several technical papers and reports and summarizes those studies (Christophe 2003, Arulampalam 2002, Haug 2005, Liu 1998, 1999)

Localization of a mobile robot can be figured as an estimation problem, where the unobserved states/pose of the mobile robot is sought by utilizing the observations made and actions taken, by the mobile agent. It is a well known fact that every action taken by the robot increases the uncertainty of robot belief in its states and every observation integrated to the robot belief in its states reduces this uncertainty to some extent. Contemporary, state of the art techniques to solve this problem, localization of mobile robot, are probabilistic approaches. Since probabilistic approaches can handle the uncertainty in sensory system and actuators in a convenient manner, they emerge as the right tool. Once the localization problem is defined probabilistically, the notion of uncertainty and probabilities emerge naturally. It can no longer be argued that the robot states are known deterministically but the robot states are defined in terms of *probability density functions* (*PDF*). That the states of mobile robot are defined in terms of *PDF*'s allows making different hypothesis. For instance if the *initial states* of the mobile robot are known in advance, a *Gaussian distribution* whose *mean* is the known states can be used, here the *covariance matrix* of the *states* defines the *uncertainty/certainty* of robot belief about its states. If the *initial states* of the mobile robot are not known at all, there is no information about the states, a *uniform distribution* can be assumed. After the initial pose of the mobile robot is defined in terms of proper *PDF'*s, the rest of the analysis is to recover the actual unobserved states from the observations

made by the robot. It is assumed that the robot navigates in the environment and records its sensory information about local setting. Mobile robot cannot observe its states directly; the robot cannot tell its states/pose. See Figure 3.1 for a *Dynamic Bayesian Network* representation of a navigating mobile robot. Only the *Global Positioning System* (***GPS***) allows the mobile robot observe its states/pose, most likely corrupted with noise. However, the mobile robot can only make observations about the environment, by which it is supposed to infer the actual states/pose. During its operation the robot can integrate its action measurements and sensory measurements about the local environment into its belief, if the statistical properties of these measurements are known. Bayes Filter may be used for this purpose; *Bayes rule* and *First-Order Markov Process* assumption leads to a recursive formula, known as *Bayes Filters*. In order to implement this recursive formula three probabilistic model must be known: *motion model*, *observation model* and the *initial PDF/a priori information* of robot states/pose. Once these *PDFs* are available, the *Bayes Filter* can be simulated with various approaches; *Deterministic Integration*, *Sequential Monte Carlo (SMC) Filters*, *Kalman Filter*.

In this chapter, the general approach to localization problem is studied. A top down approach is assumed, starting from *Dynamic Bayesian Networks*, several approaches enlisted above are derived and discussed. Different approaches having minor/major differences are taken into account and the inference of robot states from sensory measurement is investigated. The ultimate goal of this chapter is to review the related literature and establish a general framework for the particular problem of localization.

The rest of this chapter is organized as follows, in section two, mathematical definition of *localization problem* is introduced as a *state estimation*, section three is about different approaches to *discrete time estimation problem*, in section four the approaches presented in sections two and three are applied to *mobile robot localization*. The next section introduces the general framework of localization problem that is to estimate the state of a dynamic system.

3.2 Mathematical Definition of Tracking Problem

This section is a summary of (Gordon 1993). The main objective of *localization/tracking* may be defined as *estimation* that is to *track states/pose of robot* as it changes in time. *Action commands* which are executed, at discrete time steps, modify the system according to the *process model so as to determine the predicted PDF*. At certain times observations are received that corrects/updates the prior PDF of states predicted by *process model*. Formally, assuming the states/pose of the robot is known in previous time instants, the effect of action is considered by obtaining a *prior PDF* with the help of process model and the command itself, in other words the states/pose of the robot is estimated. Whenever new sensory information is received, it is used to correct /update the prior *PDF* in order to compute posterior *PDF* which is certainly more precise than the prior *PDF*. The *state vector* $\boldsymbol{x}_t$ is assumed to *evolve* according to the following *process model*:

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}_t(\boldsymbol{x}_t, \boldsymbol{w}_t) \tag{3.1}$$

where $\boldsymbol{f}_t$ is the *process model.* The *PDF* of process noise $\boldsymbol{w}_k$ is assumed to be known. At discrete times, measurements $\boldsymbol{y}_t$ are available. These measurements are related to states via the *measurement model*.

$$\boldsymbol{z}_t = \boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{v}_t) \tag{3.2}$$

where $\boldsymbol{h}_t$ is the *measurement function*. The process noise $\boldsymbol{w}_t$ and measurement noise $\boldsymbol{v}_t$ are *zero mean, white noise sequence of known PDF*, *independent of past and current states*. It is assumed that the initial *PDF* $P[\boldsymbol{x}_1|\boldsymbol{z}_0] \triangleq P(\boldsymbol{x}_1)$ of the state vector is available together with the process and measurement model $\boldsymbol{f}_i$ and $\boldsymbol{h}_i$. The available information at time step t is a set of measurements $\boldsymbol{Z}_{0:t} = \{\boldsymbol{z}_i : i = 1, \dots, t\}$. The aim is to construct the *PDF* of the current state, $\boldsymbol{x}_t$, given all the available information: $P[\boldsymbol{x}_t|\boldsymbol{Z}_{0:t}]$. It is known that this *PDF* may be computed recursively in two stages: *prediction* and *update* by using the Bayes filter. Suppose that the *PDF* $P[\boldsymbol{x}_{0:t-1}|\boldsymbol{Z}_{1:t-1}]$ at time step $t-1$ is

available. The using the *process model* it is possible to compute *prior PDF* of state at time step t

$$P[\boldsymbol{x}_t|\boldsymbol{Z}_{0:t-1}] = \int P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t]\ P[\boldsymbol{x}_{t-1}|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x}_{t-1} \tag{3.3}$$

Here the *probabilistic motion mode* $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}]$, which is a *First-Order Markov Process* model, is defined by the *process model equation* and the known statistics of process noise $\boldsymbol{w}_{t-1}$

$$P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t] = \int \delta[\boldsymbol{x}_t - \boldsymbol{f}_{t-1}(\boldsymbol{x}_{t-1}, \boldsymbol{w}_{t-1})]\ P[\boldsymbol{w}_{t-1}]d\boldsymbol{w}_{t-1} \tag{3.4}$$

Then at time step t a measurement $\boldsymbol{z}_t$ is observed and it may be used to *update the prior PDF via Bayes rules,* so as to compute the posterior PDF.

$$P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t}] = \frac{P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]}{P[\boldsymbol{z}_t|\boldsymbol{Z}_{1:t-1}]} \tag{3.5}$$

Where the normalizing denominator is given by

$$P[\boldsymbol{z}_t|\boldsymbol{Z}_{0:t-1}] = \int P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x}_t \tag{3.6}$$

Note that denominator is required to assure $P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t}]$ is a proper *PDF,* if it is ignored, then $P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t}] \propto \eta P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]$ is not known exactly but it is known up to an unknown proportional constant.

The probabilistic observation model $P[\boldsymbol{z}_t|\boldsymbol{x}_{v,t}]$ is defined by the *measurement model* and known statistics of $\boldsymbol{v}_t$

$$P[\boldsymbol{z}_t|\boldsymbol{x}_t] = \int \delta[\boldsymbol{z}_t - \boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{v}_t)]P[\boldsymbol{v}_t]d\,\boldsymbol{v}_t \tag{3.7}$$

where $\delta(.)$ is a dirac-delta function.

In the *update equation*, the measurement $\boldsymbol{z}_t$ is used to modify the *predicted prior* PDF, which is computed at previous time step to obtain the required *posterior PDF of state*. The recursive relations form the formal solution to *Bayesian recursive estimation problem*. Analytic solution is only possible for limited cases and restrictive choice of *process model* and *measurement models*, the most important being the *Kalman Filter* which assumes, linear process model $\boldsymbol{f}_t(\boldsymbol{x}_t, \boldsymbol{w}_t)$and linear measurement model $\boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{v}_t)$ and $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are additive Gaussian noise of known variance. Since analytical solutions are limited to few cases, numerical methods are employed to run this recursive formula. The critical problem arises when one tries to take the following integral; $\int P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x}_t$ may be evaluated on a *deterministic grid*, in that case the computational cost would be too high. Thus one may employ *statistical method* which places the *integration grid points* to locations with high importance. Therefore, *Sequential Monte Carlo (SMC)* methods are employed to evaluate these integrals.

In the rest of this chapter, several methods i.e. Sequential Monte Carlo techniques and Kalman filters are investigated because all these methods are used to simulate Bayes Filter. Note that the main difficulty arises from two problems:

- Sometimes, it is not possible to sample from a complex distribution which is not necessarily Gaussian. Hence the Bayes Filter cannot be simulated
- Nonlinear process and measurement models

Solutions are possible:

- Use available sampling techniques:
    - Importance Sampling

These methods allows us to represent complex *PDF* with samples i.e. particles. If the *PDF* is Gaussian switch to Kalman filter which is the only closed form

solution and the fastest one. If it is not possible, then the following methods may be used:

- Sequential Monte Carlo techniques
- Grid based Approach

3.3 Grid Based Approach

This section is a summary of (Fox 1998). It is almost impossible to sample a complex *PDF*; but these *PDF* may be evaluated on a deterministic grid. When the state space is discretized into finite size grids, the integrals in Equation (3.3), (3.4), (3.6), (3.7), are expressed as summation over the grid. Equation (3.3) can be written as

$$P[\boldsymbol{x}_t|\boldsymbol{Z}_{0:t-1}] = \sum_{i=1}^{N} P\left[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t\right] P\left[\boldsymbol{x}_{t-1}^i|\boldsymbol{Z}_{1:t-1}\right] \tag{3.8}$$

Equation (3.4) can be written as

$$P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t] = \sum_{i=1}^{M} \delta\left[\boldsymbol{x}_t - \boldsymbol{f}_{t-1}\left(\boldsymbol{x}_{t-1}^i, w_{t-1}^i\right)\right] P\left[\boldsymbol{w}_{t-1}^i\right] \tag{3.9}$$

Equation (3.6) can be written as

$$P[\boldsymbol{z}_t|\boldsymbol{Z}_{0:t-1}] = \sum_{i=1}^{N} P\left[\boldsymbol{z}_t|\boldsymbol{x}_{t-1}^i\right] P\left[\boldsymbol{x}_{t-1}^i|\boldsymbol{Z}_{1:t-1}\right] \tag{3.10}$$

Equation (3.7) can be written as

$$P[\boldsymbol{z}_t|\boldsymbol{x}_t] = \sum_{i=1}^{M} \delta\left[\boldsymbol{z}_t - \boldsymbol{h}_t\left(\boldsymbol{x}_t^i, \boldsymbol{v}_t^i\right)\right] P\left[\boldsymbol{v}_t^i\right] \tag{3.11}$$

These Equations may be used with Equation (3.5) to simulate the *Bayes recursive formula*. Note that the *PDF* are evaluated on a deterministic grid and

every node is updated without considering their importance. Hence the computational cost is very high. In the following text some intelligent approaches which approximate complex *PDF* by particles placed at positions whose likelihood is high. These methods are cost efficient and more flexible.

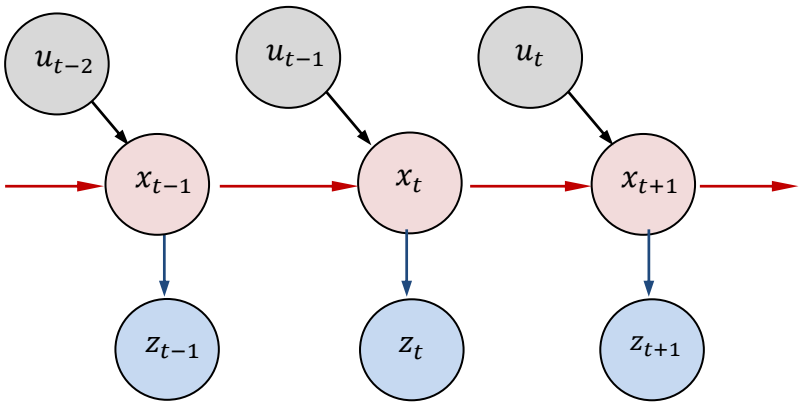


Figure 3.1 Dynamic Bayesian Network (DBN) representing a navigating mobile robot, where *u* is action, *x* is state and *z* is observation

3.4 Sequential Monte Carlo Methods for Dynamic Systems

3.4.1 Introduction

This section is a summary of (Liu 1998, 1999).The principle idea of *Monte Carlo (MC)* methods is to *generate a statistical sample that approximate a hard problem by a much simpler problem*. The backbone of Monte Carlo simulation is to draw a set of samples $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ from a *target density* $p(\boldsymbol{x})$ defined on a high-dimensional space. These *N* samples may approximate the *target density* with the *empirical point-mass function.* Thus, *the Monte Carlo principle places the grid points in regions of high likelihood so that it performs with a less number of integration points better than deterministic integration i.e. Grid based approach which is indifferent to the likelihood of integration points.* If the target *PDF* is Gaussian, it is easy to sample from that PDF using

easily available routines. However, when target *PDF*, $p(\boldsymbol{x})$ has a complex shape, *importance sampling* must be used to generate samples which imitates that target density.

Since complex, multimodal PDF are common in mobile robotics research, the next few sections will focus on *importance sampling*. Note that this method generates a set samples placed at positions whose likelihood is high. Therefore it constitutes a general framework to run *Bayes filter* in a cost effective scheme.

3.4.2 Importance Sampling

An *arbitrary importance proposal distribution* $q(\boldsymbol{x})$ is chosen such that its space covers the space of target *PDF* $p(\boldsymbol{x})$. Then $I(f) = \int f(\boldsymbol{x})p(\boldsymbol{x})dx$ can be rewritten as

$$I(f) = \int f(\boldsymbol{x})w(\boldsymbol{x})q(\boldsymbol{x})dx \tag{3.12}$$

where

$$w(\boldsymbol{x}) \triangleq \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{3.13}$$

is known as the *importance weight*. Thus, N samples $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ may be generated according to *importance proposal distribution, q(**x**)* and $w(\boldsymbol{x}^{(i)})$ is evaluated, a possible *Monte Carlo Estimate* of $I(f)$.

$$\hat{I}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}^{(i)})w(\boldsymbol{x}^{(i)}) \tag{3.14}$$

The *posterior density* $p(x)$ is approximated by

$$\hat{p}_N(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{x}^{(i)})\,\delta_{\boldsymbol{x}^{(i)}}(\boldsymbol{x}) \tag{3.15}$$

$\hat{I}_N(f)$ is may imagined as nothing but the function $f(\boldsymbol{x})$ integrated with respect to the *empirical measure* $\hat{p}_N(\boldsymbol{x})$.

When the *normalizing constant* of $p(\boldsymbol{x})$ is unknown and

$$w(\boldsymbol{x}) \propto \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \tag{3.16}$$

is now *only known up to an unknown normalizing constant*. It is still possible to apply the *importance sampling* method by rewriting $I(f)$ as follows

$$I(f) = \frac{\int f(\boldsymbol{x}) w(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x}}{\int w(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x}} \tag{3.17}$$

Since $w(\boldsymbol{x})$ is known up to an unknown constant, the constant in the numerator and denominator cancel each other. Thus the unknown constant may be safely ignored by using the above definitions.

Note that up to now, importance sampling method is investigated because it is capable of imitating complex PDF with a finite number of samples. The next subsection is an attempt to utilize this sampling method to simulate dynamic systems. This method which is used to simulate dynamics systems is in general referred as Sequential Monte Carlo methods.

3.4.3 Simulation of Dynamic Systems with Sequential Monte Carlo Method

Sequential Monte Carlo (SMC) methods are used to carry out on-line approximation of probability distributions using *samples (particles).* When data arrival is inherently sequential, they are very useful. Sequential Monte Carlo methods do not require storing all the past data; this is one of its advantages. In the *SMC* setting, it is assumed that there are an *initial distribution, a motion model and an observation model.*

$$p(\boldsymbol{x}_0) \tag{3.18}$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) = p(\boldsymbol{x}_t|\boldsymbol{X}_{0:t-1}, \boldsymbol{Z}_{1:t-1}, \boldsymbol{U}_{0:t}) \tag{3.19}$$

$$f(\boldsymbol{z}_t|\boldsymbol{x}_t) = p(\boldsymbol{z}_t|\boldsymbol{X}_{0:t}, \boldsymbol{Z}_{1:t-1}) \tag{3.20}$$

$\boldsymbol{X}_{0:t} = \{\boldsymbol{x}_0, \ldots\ldots, \boldsymbol{x}_t\}$ and $\boldsymbol{Z}_{1:t} = \{\boldsymbol{z}_1, \ldots\ldots, \boldsymbol{z}_t\}$, denotes the *states* and the *observations* sequence up to time *t*, respectively. Note that 1st *Markov process* and *conditional independence* may be assumed to simplify the process model and measurement model;

$$p(\boldsymbol{x}_t|\boldsymbol{X}_{0:t-1}, \boldsymbol{Z}_{1:t-1}, \boldsymbol{u}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \tag{3.21}$$

$$p(\boldsymbol{z}_t|\boldsymbol{X}_{0:t}, \boldsymbol{Z}_{1:t-1}) = p(\boldsymbol{z}_t|\boldsymbol{x}_t) \tag{3.22}$$

The objective is to estimate in time the posterior *PDF* $p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})$, and its associated features including the marginal distribution of $p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})$, known as the *filtering distribution*, and the expectations, recursively

Multiple Monte Carlo samples which are drawn from a complicated high dimensional distribution, such as $\pi_t(\boldsymbol{x}_t)$ may be used to imitate the distribution itself. Let $S_t = \{\boldsymbol{x}^{(j)}, j = 1,2 \ldots\ldots, m\}$ denote a set of random draws that are weighted by the set of weights $W_t = \{w^{(j)}, j = 1,2 \ldots\ldots, m\}$ with respect to π_t, properly. Let g_{t+1} be a trial distribution. Then the Sequential Importance Sampling (SIS) procedure may be summarized in Table 3.1.

$\boldsymbol{x}_t^{(j)}$ is drawn from the trial distribution $g_t\left(\boldsymbol{x}_t^{(j)}\middle|\, \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{u}_t\right)$ and is attached to $\boldsymbol{x}_{t-1}^{(j)}$ to form the tuple $\left\{\boldsymbol{x}_t^{(j)}, \boldsymbol{x}_{t-1}^{(j)}\right\}$. Then the new weight $w_t^{(j)}$ is computed.

Here $g_t\left(\boldsymbol{x}_t^{(j)}\middle|\, \boldsymbol{x}_{t-1}^{(j)}, \boldsymbol{u}_t\right)$ is the *trial distribution* which extends the *posterior distribution* $p(\boldsymbol{x}_{t-1}|\boldsymbol{Z}_{1:t-1})$ at time $t-1$ to prior *PDF* at time t by an approximate/incomplete way. Note that *SIS* is derived from *importance sampling* and it is r*ecursive* in nature, so it allows simulating a dynamical system as time increase

Table 3.1 SIS Algorithm

SIS Steps: for $j = 1,2 \ldots, m$
a. Draw $\boldsymbol{x_t = x_t^{(j)}}$ from $g_t\left(\boldsymbol{x_t^{(j)}} \middle\| \boldsymbol{x_{t-1}^{(j)}, u_t}\right)$; attach it to $\boldsymbol{x_{t-1}^{(j)}}$ to form $\boldsymbol{x_t^{(j)} = \left\{x_t^{(j)}, x_{t-1}^{(j)}\right\}}$
b. Let $w_t^{(j)} = \frac{p(\boldsymbol{z_t}\|\boldsymbol{x_t})p(\boldsymbol{x_t}\|\boldsymbol{x_{t-1}, u_t})}{g_t(\boldsymbol{x_t}\|\boldsymbol{x_{t-1}, u_t})} w_{t-1}^{(j)}$

At this point there are two options:

1. Do Resampling
2. Proceed with the newly sampled $\left(\boldsymbol{x_{t-1}^{(j)}, x_t}\right)$ with new weights $w(j, \boldsymbol{x_t})$

A well known problem of the *SIS filter* is the *degeneracy problem* which appears after a few iterations. All but one particle will have negligible weights. Apparently, degeneracy is a disadvantage in *SIS*. The easy approach to reducing this effect is to use *very large population*. This is often not practical and so there are two other methods:

- A good choice of *importance density function*
- Use of *Resampling.*

Importance density function may be chosen so as to avoid the degeneracy problem. Although *Resampling* reduces the effects of degeneracy problem by eliminating the particles with negligible weights, it causes other practical problems. The particles with large weights are likely to be selected many times. This leads to *loss of diversity* among the particles as the resultant sample will contain *many repeated particles*. This problem, known as *sample impoverishment* is severe in the case of *small process noise*.

3.4.4 Instances of SIS Algorithm

In this section, two distinct instance of SIS algorithm will be described: the first one is *Sampling Importance Resampling (SIR)* and the other one is *Auxiliary variable*.

3.4.4.1 Sampling Importance Resampling

Sampling Importance Resampling is explained by using (Gordon 1993, Haug 2005). The *SIR algorithm* can be derived from *SIS algorithm* by an appropriate choice of

- The *importance density* $g_t^*(\boldsymbol{x_t}|\boldsymbol{x_{t-1}^{(j)}}, \boldsymbol{u}_t)$ is chosen to be $P\left[\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t\right]$
- *Resampling step* is applied at every time index

The above choice of *importance density* implies that samples are needed from $P\left[\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t\right]$. A sample $\boldsymbol{x}_t^i \sim P\left[\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_t\right]$ can be generated by first generating a process noise sample $v_{t-1}^i \sim P_v(v_{t-1})$ and then setting $\boldsymbol{x}_t^i = f_t\left(\boldsymbol{x}_{t-1}^i, v_{t-1}^i\right)$, where $P_v(v_{t-1})$ is the *PDF* of v_{t-1}. For this particular importance density, it is clear that the weights are given by $w_t^i \; \alpha \; w_{t-1}^i P\left[\boldsymbol{z}_t|\boldsymbol{x}_t^i\right]$

However note that *Resampling* is applied at every time index, then $w_t^i = \frac{1}{N_s}$ and so $w_t^i \; \alpha \; P\left[\boldsymbol{z}_t|\boldsymbol{x}_t^i\right]$. The weights given by the proportionality are normalized before the resampling step.

3.4.4.2 Auxiliary Variable

This section is a summary of (Liu 1998). Objective is to sample from the *joint density* $P(\boldsymbol{x}_{t+1}, k|\boldsymbol{Z}_{0:t+1})$ where k is an index on mixture. Define

$$P(\boldsymbol{x}_{t+1}, k|\boldsymbol{Z}_{0:t+1}) = P(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1})P\left(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^k, \boldsymbol{u}_{t+1}\right)w_t^k \tag{3.23}$$

and draw from this joint density $(\boldsymbol{x}_{t+1}, k)$ and discard the index k to produce a sample from the *empirical filtering density* as required. k is called *an auxiliary variable* as it aids the task of simulation. This type of generic particle filter is

named as *Auxiliary Particle Filters.* $P(\boldsymbol{x}_{t+1}, k|\boldsymbol{Z}_{0:t+1})$ may be approximated as

$$P(\boldsymbol{x}_{t+1}, k|\boldsymbol{Z}_{0:t+1}) = P(\boldsymbol{z}_{t+1}|\mu_{t+1}^{k})P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t}^{k}, \boldsymbol{u}_{t+1})w_{t}^{k} \tag{3.24}$$

where μ_{t+1}^{k} is the *mean, the mode, a draw, or some other likely value associated with the density of* $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t}^{k}, \boldsymbol{u}_{t})$. The probability λ_k which is proportional to $P(k|\mathbf{Z}_{0:t+1})$ may computed as:

$$P(k|\boldsymbol{Z}_{0:t+1}) \; \alpha \\ \int P(\boldsymbol{z}_{t+1}|\mu_{t+1}^{k})P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t}^{k}, \boldsymbol{u}_{t+1})w_{t}^{k} \, d\boldsymbol{x}_{t+1} = w_{t}^{k}P(\boldsymbol{z}_{t+1}|\mu_{t+1}^{k}) \tag{3.25}$$

$$\lambda_k = w_{t}^{k}P(\boldsymbol{z}_{t+1}|\mu_{t+1}^{k}) \tag{3.26}$$

The index with probability λ_k is simulated to sample from $P(k|\boldsymbol{Z}_{0:t+1})$ and then the *transition density* is sampled given the mixture $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t}^{k}, \boldsymbol{u}_{t})$. λ_k is called the *first stage weights*. Having sampled the joint density of $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t}^{k}, \boldsymbol{u}_{t+1})$ R times, reweighting is performed, and $(\boldsymbol{x}_{t+1}^{j}, k^{j})$ are formed; the weights proportional to the so-called *second stage weights* are computed as:

$$w_{t+1}^{j} = \frac{P(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1}^{j})}{P(\boldsymbol{z}_{t+1}|\mu_{t+1}^{kj})} \tag{3.27}$$

Finally, the weights are normalized

$$w_{t+1}^{i} = \frac{w_{t+1}^{i}}{\sum_{j=1}^{R} w_{t+1}^{j}} \tag{3.28}$$

3.5 Kalman Filter and Variants

Kalman filter is a powerful method for sensor fusion and filtering noisy data so as to obtain the optimal representation of a measurement. Kalman filter is derived by using linear process and measurement models and it is assumed that

the process and measurement noise is zero-mean Gaussians. However, most of the dynamical systems are nonlinear. Therefore Extended Kalman filter which uses the linearized process and/or measurement models and assumes the process and measurement noise is zero-mean Gaussian is designed to cope with these situations. Once the linearized process and measurement models are obtained, the Extended Kalman filter runs very similar to Kalman filter. In the following text, only Extended Kalman filter is discussed, because it is relevance to this study, is more striking than Kalman filter.

3.5.1 Extended Kalman Filter

Extended Kalman Filter is explained by using references (Brown 1993, Thrun 2005). In the following text, the short notation is used.

Predicted states

$$\widehat{\boldsymbol{x}}_t^- = P(\boldsymbol{x}_t|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t-1}) \tag{3.29}$$

Updated states

$$\widehat{\boldsymbol{x}}_t^+ = P(\boldsymbol{x}_t|\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}) \tag{3.30}$$

Predicted covariance

$$\boldsymbol{P}_t^- = P(\boldsymbol{P}_t|\boldsymbol{Z}_{0:t-1}, \boldsymbol{U}_{0:t-1}) \tag{3.31}$$

Updated covariance

$$\boldsymbol{P}_t^+ = P(\boldsymbol{P}_t|\boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}) \tag{3.32}$$

The EKF is initialized with the *posterior state estimate* $\widehat{\boldsymbol{x}}_0^+$ and *uncertainty* $\boldsymbol{P}_0^+$ at time step 0. At every time step, the *EKF* propagates the state and *uncertainty of the system* at the previous time step to current time using the *prediction equations.*

States are predicted by using the process model:

$$\widehat{\boldsymbol{x}}_t^- = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{t-1}^+) \tag{3.33}$$

Predicted covariance is computed as follows:

$$\boldsymbol{P}_t^- = \boldsymbol{A}_t \boldsymbol{P}_{t-1}^+ \boldsymbol{A}_t^T + \boldsymbol{Q}_{t-1} \tag{3.34}$$

where the *Jacobian matrix* $\boldsymbol{A}_t$ contains the partial derivatives of *system function* $f(.)$ with respect to state **x**, and evaluated at the *posterior state estimate* $\hat{\boldsymbol{x}}_{t-1}^+$ of the last time step. The *EKF* corrects the *prior state estimate* with a full measurement $\boldsymbol{z}_t$ by means of the *correction equations*

First, the Kalman gain is computed

$$\boldsymbol{K}_t = \boldsymbol{P}_t^- \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_t^- \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1} \tag{3.35}$$

Then, states are updated by a weighted combination of predicted states and measurement innovation. Note that the weight of the measurement innovation is the Kalman gain.

$$\hat{\boldsymbol{x}}_t^+ = \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t [\boldsymbol{z}_t - \boldsymbol{h}(\hat{\boldsymbol{x}}_t^-)] \tag{3.36}$$

Finally, covariance matrix is updated

$$\boldsymbol{P}_t^+ = [\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t] \boldsymbol{P}_t^- \tag{3.37}$$

where the *Jacobian matrix* $\boldsymbol{H}_t$ contains the partial derivatives of the *measurement function* $\boldsymbol{h}(.)$ with respect to the state $\boldsymbol{x}$, evaluated at the *prior state estimate* $\hat{\boldsymbol{x}}_t^-$. $\boldsymbol{Q}_{t-1}$ and $\boldsymbol{R}_t$ are process and measurement noise covariance matrix

3.5.2 Information Filter

Description of information filter is extracted from (Thrun 2004, 2005). In the information filter the estimated covariance and estimated states are replaced by the information matrix an information vector, respectively.

Information matrix

$$\boldsymbol{Y}_t^+ = [\boldsymbol{P}_t^+]^{-1} \tag{3.38}$$

Information vector

$$\hat{\boldsymbol{y}}_t^+ = [\boldsymbol{P}_t^+]^{-1}\hat{\boldsymbol{x}}_t^+ \quad 3.39)$$

Similarly, the predicted covariance and state have equivalent information forms.

Predicted information matrix

$$\boldsymbol{Y}_t^- = [\boldsymbol{P}_t^-]^{-1} \quad (3.40)$$

Predicted information vector

$$\hat{\boldsymbol{y}}_t^- = [\boldsymbol{P}_t^-]^{-1}\hat{\boldsymbol{x}}_t^- \quad (3.41)$$

The measurement covariance and measurement vector are defined as:

Measurement covariance

$$\boldsymbol{I}_t = \boldsymbol{H}_t^T[\boldsymbol{R}_t]^{-1}\boldsymbol{H}_t \quad (3.42)$$

Measurement vector

$$\boldsymbol{i}_t = \boldsymbol{H}_t^T[\boldsymbol{R}_t]^{-1}\boldsymbol{z}_t \quad (3.43)$$

The information update now becomes a trivial sum.

Updated information matrix

$$\boldsymbol{Y}_t^+ = \boldsymbol{Y}_t^- + \boldsymbol{I}_t \quad (3.44)$$

Updated information vector

$$\hat{\boldsymbol{y}}_t^+ = \hat{\boldsymbol{y}}_t^- + \boldsymbol{i}_t \quad (3.45)$$

The main advantage of the information filter is that N measurements can be filtered at each time step simply by summing their information matrices and vectors.

$$\boldsymbol{Y}_t^+ = \boldsymbol{Y}_t^- + \sum_{j=1}^{N} \boldsymbol{I}_{t,j} \quad (3.46)$$

$$\hat{\boldsymbol{y}}_k^+ = \hat{\boldsymbol{y}}_t^- + \sum_{j=1}^{N} \boldsymbol{i}_{t,j} \tag{3.47}$$

To predict the information filter the information matrix and vector can be converted back to their state space equivalents, or alternatively the information space prediction can be used.

$$\boldsymbol{M}_t = [\boldsymbol{F}_t^{-1}]^T \boldsymbol{Y}_{t-1}^+ \boldsymbol{F}_t^{-1} \tag{3.48}$$

$$\boldsymbol{C}_t = \boldsymbol{M}_t [\boldsymbol{M}_t + \boldsymbol{Q}_t^{-1}]^{-1} \tag{3.49}$$

$$\boldsymbol{L}_t = \boldsymbol{I} - \boldsymbol{C}_t \tag{3.50}$$

$$\boldsymbol{Y}_{t-1}^+ = \boldsymbol{L}_t \boldsymbol{M}_t \boldsymbol{L}_t^T + \boldsymbol{C}_t \boldsymbol{Q}_t^{-1} \boldsymbol{C}_t^T \tag{3.51}$$

$$\hat{\boldsymbol{y}}_t^- = \boldsymbol{L}_t [\boldsymbol{F}_t^{-1}]^T \hat{\boldsymbol{y}}_{t-1}^+ \tag{3.52}$$

3.5.3 Unscented Kalman Filter

This section is a summary of (Julier 1995). When the *state transition* and *observation models* are highly nonlinear, the *Extended Kalman Filter* can give particularly poor performance. This is because only the mean is propagated through the non-linearity. The *Unscented Kalman Filter* uses a *deterministic sampling technique* known as the *unscented transform* to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the nonlinear functions and covariance of the estimate is then recovered. The result is a filter which captures the mean and covariance more accurately.

Predict

The estimated state and covariance are augmented with the mean and covariance of the process noise, respectively.

$$[\boldsymbol{x}_{t-1}^+]^a = [[\hat{\boldsymbol{x}}_{t-1}^+]^T \quad E[w_k^T]]^T \tag{3.53}$$

$$[\boldsymbol{P}_{t-1}^{+}]^{a} = \begin{bmatrix} \boldsymbol{P}_{t-1}^{+} & 0 \\ 0 & \boldsymbol{Q}_{t} \end{bmatrix} \tag{3.54}$$

A set of (2L+1) sigma points are derived from the augmented state and covariance, where L is the dimension of augmented state.

$$[\boldsymbol{\chi}_{t-1}^{+}]^{0} = [\boldsymbol{x}_{t-1}^{+}]^{a} \tag{3.55}$$

$$[\boldsymbol{\chi}_{t-1}^{+}]^{i} = [\boldsymbol{x}_{t-1}^{+}]^{a} + \left[\sqrt{(L+\lambda)[P_{t-1}^{+}]^{a}}\right]_{i} \quad i = 1 \dots L \tag{3.56}$$

$$[\boldsymbol{\chi}_{t-1}^{+}]^{i} = [\boldsymbol{x}_{t-1}^{+}]^{a} - \left[\sqrt{(L+\lambda)[\boldsymbol{P}_{t-1}^{+}]^{a}}\right]_{i-L} \quad i = L+1 \dots 2L \tag{3.57}$$

Where $\left[\sqrt{(L+\lambda)[\boldsymbol{P}_{t-1}^{+}]^{a}}\right]_{i}$ is the i^{th} column of the square root of $(L+\lambda)[\boldsymbol{P}_{t-1}^{+}]^{a}$. The sigma points are then propagated through the transition function.

$$[\boldsymbol{\chi}_{t}^{-}]^{i} = f\left([\boldsymbol{\chi}_{t-1}^{+}]^{i}\right) \; i = 1..2L \tag{3.58}$$

The weighted sigma points recombined to produce the predicted state and covariance.

$$\hat{\boldsymbol{x}}_{t}^{-} = \sum_{i=1}^{2L} W_{s}^{i}[\boldsymbol{\chi}_{t}^{-}]^{i} \tag{3.59}$$

$$\boldsymbol{P}_{t}^{-} = \sum_{i=1}^{2L} W_{c}^{i}\left[[\boldsymbol{\chi}_{t}^{-}]^{i} - \hat{\boldsymbol{x}}_{t}^{-}\right]\left[[\boldsymbol{\chi}_{t}^{-}]^{i} - \hat{\boldsymbol{x}}_{t}^{-}\right]^{T} \tag{3.60}$$

Where the weights for the state and covariance are given by

$$W_{s}^{0} = \frac{\lambda}{L+\lambda} \tag{3.61}$$

$$W_c^0 = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{3.62}$$

$$W_s^i = W_c^i = \frac{\lambda}{2(L+\lambda)} \tag{3.63}$$

$$\lambda = \alpha^2(L + \kappa) - L \tag{3.64}$$

Typical values for $\alpha, \beta, and\ \kappa$ are 10^{-3}, 2 and 0 respectively.

Update

The predicted state and covariance are augmented as before, except now with the mean and covariance of the measurement noise, respectively.

$$[\boldsymbol{x}_t^-]^a = [[\hat{\boldsymbol{x}}_t^-]^T \quad E[v_k^T]]^T \tag{3.65}$$

$$[\boldsymbol{P}_t^-]^a = \begin{bmatrix} \boldsymbol{P}_t^- & 0 \\ 0 & \boldsymbol{R}_t \end{bmatrix} \tag{3.66}$$

As before, (2L+1) sigma points are derived from the augmented state and covariance.

$$[\boldsymbol{\chi}_t^-]^0 = [\boldsymbol{x}_t^-]^a \tag{3.67}$$

$$[\boldsymbol{\chi}_t^-]^i = [\boldsymbol{x}_t^-]^a + \left[\sqrt{(L+\lambda)[\boldsymbol{P}_t^-]^a}\right]_i \qquad i = 1 \ldots L \tag{3.68}$$

$$[\boldsymbol{\chi}_t^-]^i = [\boldsymbol{x}_t^-]^a - \left[\sqrt{(L+\lambda)[\boldsymbol{P}_t^-]^a}\right]_{i-L} \quad i = L+1 \ldots 2L \tag{3.69}$$

The sigma points are then propagated through the observation function h.

$$[\boldsymbol{\zeta}_t^-]^i = h\left([\boldsymbol{\chi}_{t-1}^+]^i\right) \ i = 1..2L \tag{3.70}$$

The weighted sigma points are recombined to produce the predicted measurement mean and predicted measurement covariance.

$$\hat{\boldsymbol{x}}_t^- = \sum_{i=1}^{2L} W_s^i [\boldsymbol{\zeta}_t^-]^i \tag{3.71}$$

$$\boldsymbol{P}_t^- = \sum_{i=1}^{2L} W_c^i \left[[\boldsymbol{\zeta}_t^-]^i - \hat{\boldsymbol{z}}_t\right]\left[[\boldsymbol{\zeta}_t^-]^i - \hat{\boldsymbol{z}}_t\right]^T \tag{3.72}$$

The state-measurement cross-covariance matrix

$$\boldsymbol{P}_{x_t z_t} = \sum_{i=1}^{2L} W_c^i \left[[\boldsymbol{\chi}_t^-]^i - \hat{\boldsymbol{x}}_t^-\right]\left[[\boldsymbol{\zeta}_t^-]^i - \hat{\boldsymbol{z}}_t\right]^T \tag{3.73}$$

is used to compute the UKF Kalman Gain

$$\boldsymbol{K}_t = \boldsymbol{P}_{x_t z_t} \boldsymbol{P}_{z_t z_t}^{-1} \tag{3.74}$$

As with the Kalman Filter, the updated state is the predicted state plus the innovation weighted by the Kalman gain.

$$\hat{\boldsymbol{x}}_t^+ = \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t(\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t) \tag{3.75}$$

And the updated covariance is the predicted covariance, minus the predicted measurement covariance, weighted by the Kalman gain.

$$\boldsymbol{P}_t^+ = \boldsymbol{P}_t^+ - \boldsymbol{K}_t \boldsymbol{P}_{z_t z_t} \boldsymbol{K}_t^T \tag{3.76}$$

3.5.4 Gaussian Sum Filter

This section is a summary of (Kotecha 2003). A single Extended Kalman Filter (*EKF*) may approximate a Gaussian *PDF* of states of nonlinear system. However it cannot represent a multimodal *PDF*. In order to cope with this situation multiple *EKF* linearized at different local states may used. So that multiple *EKF* runs in parallel throughout the analysis projected forward in time with the process model and weighted with measurement model. Such an approach may be used to deal with the multimodal nature of the global

localization. First the initial *PDF* representing our knowledge about the system can be represented with multiple Gaussian *PDFs*. This is expressed in Equation

$$P(\boldsymbol{x}_0|\boldsymbol{z}^{-1}) = \sum_{i=1}^{\xi_0'} \alpha_{0i}' \, N(\boldsymbol{x}_0 : \boldsymbol{x}_{0i}', \boldsymbol{P}_{0i}') \tag{3.77}$$

Where α_{0i}' are the weights and $N(\boldsymbol{x}_0 : \boldsymbol{x}_{0i}', \boldsymbol{P}_{0i}')$ is the i^{th} Gaussian *PDF* with mean $\boldsymbol{x}_{0i}'$ and covariance $\boldsymbol{P}_{0i}'$ evaluated at $\boldsymbol{x}_0$ and number of Gaussian *PDF's* are ξ_0'. Positive weights,α_{0i}' are normalized such that their sum is equal to one. This expressed with Equation (3.78).

$$\sum_{i=1}^{\xi_0'} \alpha_{0i}' = 1 \tag{3.78}$$

$$\alpha_{0i}' > 0 \tag{3.79}$$

Each *EKF* are projected in time through the process model. Note that each *EKF* are non-interacting and separate predictive *PDF* are computed for each of them. The predictive approximate *PDF* is expressed in Equation (3.80). The predicted mean of i^{th} *EKF* is computed with the process model and $\widehat{\boldsymbol{x}}_{t,i}$ which is the mean computed in previous time step. The predicted covariance of i^{th} EKF,$\boldsymbol{P}_{t+1,i}'$ is computed with Equation (3.82) where $\boldsymbol{F}(\widehat{\boldsymbol{x}}_{t,i})$ is the Jacobian of the process model $f_t(\widehat{\boldsymbol{x}}_{t,i})$, which is linearized at $\widehat{\boldsymbol{x}}_{t,i}$, $\boldsymbol{Q}_t$ is the Gaussian process model, $\boldsymbol{P}_{t,i}$ is the posterior covariance matrix. The weights and number of EKF are kept unchanged. When a new observation is received, the covariance of i^{th} *EKF*, $\boldsymbol{S}_{t,i}$, is computed with Equation (3.86) where $\boldsymbol{H}(\boldsymbol{x}_{t,i}')$ is the Jacobian of the measurement model linearized at $\boldsymbol{x}_{t,i}'$ and $\boldsymbol{R}_t$ is the Gaussian measurement noise. Kalman Gain of the i^{th} *EKF* $\boldsymbol{K}_{t,i}$, is computed with Equation (3.87) and the updated state, $\widehat{\boldsymbol{x}}_{t,i}$, is computed with Equation (3.88). The covariance of i^{th} EKF, $\boldsymbol{P}_{t,i}$, is computed with Equation (3.89). Then each separate EKF is weighted by the nonnormalized likelihood of

measurements. Finally the weights are normalized such that their sum is equal to one.

The approximate predictive PDF

$$P_A(\boldsymbol{x}_{t+1}|\boldsymbol{z}_t) = \sum_{i=1}^{\xi'_{k+1}} \alpha'_{t+1,i}\, N\left(\boldsymbol{x}_{t+1}: \boldsymbol{x}'_{t+1,i}, \boldsymbol{P}'_{t+1,i}\right) \tag{3.80}$$

$$\boldsymbol{x}'_{t+1,i} = f_t\left(\widehat{\boldsymbol{x}}_{t,i}\right) \tag{3.81}$$

$$\boldsymbol{P}'_{t+1,i} = \boldsymbol{F}\left(\widehat{\boldsymbol{x}}_{t,i}\right)\boldsymbol{P}_{t,i}\boldsymbol{F}^T\left(\widehat{\boldsymbol{x}}_{t,i}\right) + \boldsymbol{Q}_t \tag{3.82}$$

$$\alpha'_{t+1,i} = \alpha_{t,i} \tag{3.83}$$

$$\xi'_{t+1} = \xi_t \tag{3.84}$$

The approximate filtering PDF

$$P_t(\boldsymbol{x}_t|\boldsymbol{z}_t) = \sum_{i=1}^{\xi_k} \alpha_{t,i}\, N\left(\boldsymbol{x}_t: \widehat{\boldsymbol{x}}_{t,i}, \boldsymbol{P}_{t,i}\right) \tag{3.85}$$

$$\boldsymbol{S_{t,i}} = \boldsymbol{H}\left(\boldsymbol{x}'_{t,i}\right)\boldsymbol{P}'_{t,i}\boldsymbol{H}^T\left(\boldsymbol{x}'_{t,i}\right) + \boldsymbol{R}_t \tag{3.86}$$

$$\boldsymbol{K}_{t,i} = \boldsymbol{P}'_{t,i}\boldsymbol{H}^T\left(\boldsymbol{x}'_{t,i}\right)\boldsymbol{S}_{t,i}^{-1} \tag{3.87}$$

$$\widehat{\boldsymbol{x}}_{t,i} = \boldsymbol{x}'_{t,i} + \boldsymbol{K}_{t,i}\left[\boldsymbol{z}_t - \boldsymbol{h}_t(\boldsymbol{x}'_{t,i})\right] \tag{3.88}$$

$$\boldsymbol{P}_{t,i} = \left[\boldsymbol{I} - \boldsymbol{K}_{t,i}\,\boldsymbol{H}\left(\boldsymbol{x}'_{t,i}\right)\right]\boldsymbol{P}'_{t,i} \tag{3.89}$$

$$\zeta_{t,i} = N\left(\boldsymbol{z}_t: \boldsymbol{h}_t\left(\boldsymbol{x}'_{t,i}\right), \boldsymbol{S_{t,i}}\right) \tag{3.90}$$

$$\alpha_{t,i} = \frac{\alpha'_{t,i}\zeta_{t,i}}{\sum_{i=1}^{\xi'_t} \alpha'_{t,i}\zeta_{t,i}} \tag{3.91}$$

$$\xi_t = \xi'_t \tag{3.92}$$

3.6 Localization Problem Illustrated With the State Space Model of a Navigating Mobile Robot

In preceding subchapters, a general approach to tracking problem is investigated. In this subchapter, these general approaches will be elaborated with process model of a mobile robot, equipped with a laser range finder operating 2D environment. The relevant parameters of the mobile robot are illustrated on the Figure 3.2. These are; position of a reference point on mobile robot measured from a global Cartesian coordinate, and the heading of the mobile robot which is measured CCW from x-axis. By using the geometry of the mobile robot and assuming that the robot navigates at constant speed, the kinematic model of a mobile robot can be expressed as

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + T \begin{bmatrix} v\cos\theta_t \\ v\sin\theta_t \\ \left[\frac{1}{L}\right] v\tan\beta \end{bmatrix} + \begin{bmatrix} N(0,\sigma_x) \\ N(0,\sigma_y) \\ N(0,\sigma_y) \end{bmatrix} \tag{3.93}$$

where $\{x, y, \theta\}$ are the coordinates and heading of the mobile robot and $\{v, \beta\}$ are the translational velocity and steering angle of mobile robot, respectively. The robot kinematic model is illustrated in Figure 3.2. The measurement model of the mobile robot is expressed as

$$\begin{bmatrix} O_{11} \\ O_{12} \\ \vdots \end{bmatrix} = \begin{bmatrix} \sqrt{[x_r - x_L^1]^2 + [y_r - y_L^1]^2} \\ atan\left(\frac{y_L^1 - y_r}{x_L^1 - x_r}\right) + \theta \\ \vdots \end{bmatrix} + \begin{bmatrix} N(0,\sigma_{11}) \\ N(0,\sigma_{12}) \\ \vdots \end{bmatrix} \tag{3.94}$$

where $\{x_r, y_r, \theta\}$ are the states of mobile robot, $O_{n1,2}$ and $\{x_L^n, y_L^n\}$ are the measurement to the n^{th} landmark, and the coordinates of the considered landmark, respectively. σ denotes the standard deviation of Gaussians.

In order to reduce the computational complexity, the robot's heading is measured by a compass which changes within certain (absolute) bounds. Throughout this study, the nominal value of the compass measurement is used as heading angle data.

$$\begin{bmatrix} O_1 \\ \vdots \\ O_n \end{bmatrix} = \begin{bmatrix} \sqrt{[x - x_L^1]^2 + [y - y_L^1]^2} \\ \vdots \\ \sqrt{[x - x_L^n]^2 + [y - y_L^n]^2} \end{bmatrix} + \begin{bmatrix} N(0, \sigma_1) \\ \vdots \\ N(0, \sigma_n) \end{bmatrix} \tag{3.95}$$

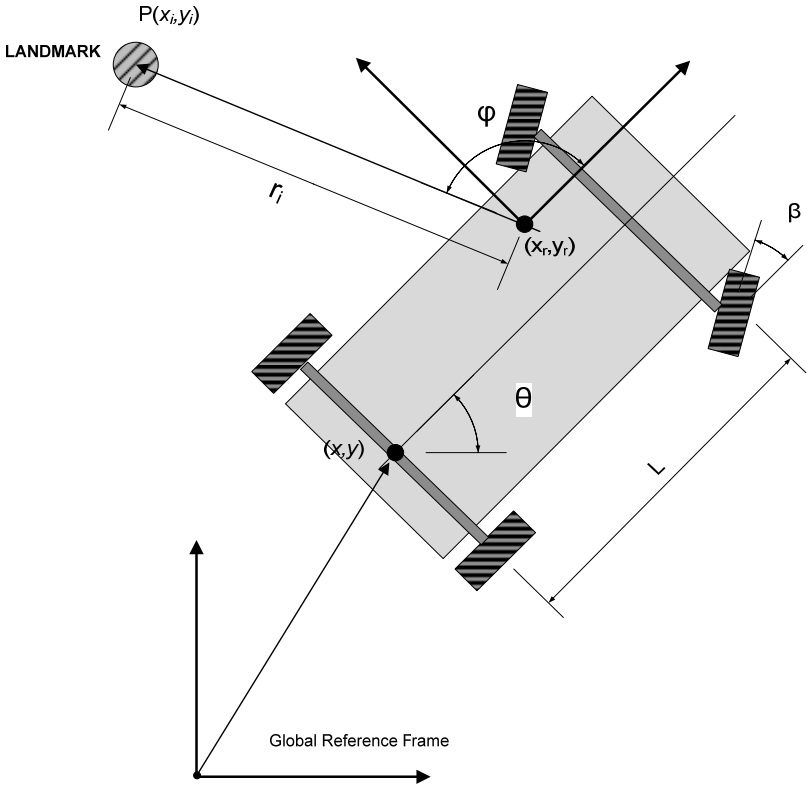


Figure 3.2 Vehicle model

$$\boldsymbol{x}_{t+1} = f_t(\boldsymbol{x}_t, w_t) \tag{3.96}$$

$$\boldsymbol{y}_{t+1} = h_t(\boldsymbol{x}_{t+1}, v_t) \tag{3.97}$$

Suppose one is interested in *estimating state of a dynamical system* on line. Thus the *dynamic system* or the *state space model* is

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{Z}_{0:t+1}) \propto \prod_{s=1}^{t} P(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1})P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_{t+1}) \tag{3.98}$$

Where $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$ can be obtained with Equation (3.96) and $P(\boldsymbol{z}_t|\boldsymbol{x}_t)$ can be obtained with Equation (3.97).

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) \propto \frac{P(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t)}{P(\boldsymbol{x}_t|\boldsymbol{Z}_{0:t})} = P(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1})P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{3.99}$$

Similarly with $g^*_{t+1}(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) = P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)$ the *importance Resampling procedure* becomes:

- Draw $J = j$ with probability proportional to $w_t^{(j)}$
- Draw $\boldsymbol{x}_{t+1}$ from $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_{t+1})$
- Assign weights to the sample $(J, \boldsymbol{x}_{t+1})$ as $w_t^{(j)} P(\boldsymbol{z}_{t+1}|\boldsymbol{x}_{t+1})$

The procedure of *Sampling Importance Resampling* is exactly the above with an additional step of Resampling from the obtained sample using the assigned weight. *Auxiliary Variable* incorporates an adjustment multiplier

$$\boldsymbol{x}_{t+1}^j = P(\boldsymbol{z}_{t+1}|\mu_{t+1}^j) \tag{3.100}$$

Where μ_{t+1}^j can be *mode, mean or more likely value of* $\boldsymbol{x}_{t+1}^{(j)}$ Thus the resulting weight for the obtained sample is

$$w(j, \boldsymbol{x}_{t+1}) = \frac{P(\boldsymbol{z}_{t+1}|x_{t+1}^j)}{P(\boldsymbol{z}_{t+1}|\mu_{t+1}^j)} \tag{3.101}$$

The *Jacobian matrix* $\boldsymbol{A}_t$ and *Jacobian matrix* $\boldsymbol{H}_t$ in *Kalman Filter* can be computed as follows

$$\boldsymbol{A}_t = \frac{\partial \boldsymbol{f}_t}{\partial x}\Big| \boldsymbol{x} = \hat{\boldsymbol{x}}_{t-1}^{+} \tag{3.102}$$

The Jacobian of process model is evaluated at the $\boldsymbol{x} = \hat{\boldsymbol{x}}_{t-1}^{+}$. Note that this best value available for the states

$$\boldsymbol{H}_t = \frac{\partial \boldsymbol{h}_t}{\partial x}\Big| \boldsymbol{x} = \hat{\boldsymbol{x}}_{t}^{-} \tag{3.103}$$

The Jacobian of measurement model is evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}_{t}^{-}$. Again note that this the best value for the states.

3.7 Conclusion and Discussion

In this chapter, a general approach to tracking problem is investigated. The following methods are elaborated:

- Markov localization
- Markov Chain Monte Carlo methods
    - Importance sampling method
    - Sequential Monte Carlo Methods
- Kalman filter
    - Extended Kalman filter
    - Information filter
    - Unscented Kalman filter
    - Gaussian sum filter

Note that the first two groups are sample based and Kalman filter and derivatives offer a closed form analytical solution to the problem, so they are faster.

The problem of position tracking of a mobile robot is certainly nonlinear due to nonlinear process and measurement models. The process model includes trigonometric functions and the measurement model includes Euclidian distances which are certainly nonlinear. Therefore any attempt to solve this

problem must be capable of coping with nonlinear, non-Gaussian problems. In addition, the global localization problem which is a generalization of tracking problem requires that the solution techniques must hold a multi-modal PDF in position.

The tracking problem may be solved by Extended Kalman filter which offers a closed form solution for the problem considered here. The derivatives of Kalman filter; the information filter and unscented Kalman filter may also solve the tracking problem. Information filter is the dual of Kalman filter; in Kalman filter the update stage demands the most computation power whereas in Information filter demands the most computation power in prediction stage, so information filter offers an advantage when multiple observations are made. Actually the advantage of information filter appears in SLAM problem as the information matrix is sparse and accelerates SLAM algorithms (Thrun 2004). Here it is given as an alternative representation of probabilistic state estimation. Unscented Kalman filter offers more flexibility to account for the nonlinear process and measurement models. Likewise Gaussian Sum filter offers great flexibility to account these nonlinear terms.

Nevertheless, the global localization problem which is a more difficult problem than position tracking, demands more computational power and more advanced techniques. Note that solution of global localization may be achieved by tracking multi hypothesis at the start of simulation. This multi hypothesis is required because several candidate locations may arise due to geometrically similar locations in the environment. Besides process and measurement noise may delay the determination of the location precisely. It is obvious that multimodal PDF is realty in global localization.

Multimodal PDFs obscure analytical solution. It is not possible to sample complex PDFs. Therefore the following advanced techniques are introduced:

- Markov localization

- Sequential Monte Carlo methods
    - Importance sampling method

Sequential Monte Carlo methods and Markov localization are numerical sample based methods, so they are capable of imitating complex PDFs with a high computational cost. They are flexible and may exploit the available computational power so as to optimize system resources. These methods allow dealing with complex PDFs. The main idea behind these methods is to represent the complex PDFs with a finite number of points/grids. Once the complex PDFs are represented numerically, the Bayes Filter may be simulated. However numerical simulation of Bayes filter is not practical due to integration on high dimensional spaces. Markov localization which approximates complex PDFs on fixed size deterministic grid is the least cost effective. Therefore more efficient techniques like Importance sampling method must be introduced so as to place the grid points at positions whose likelihood is high. All these numerical techniques may solve the problem considered here.

In this study, the importance sampling i.e. Monte Carlo Localization has been selected as the primary localization technique since it is a common tool in mobile robotics community. In general Kalman filters are not useful for the objective of this study, since they hold a unimodal belief in states. However, Gaussian sum filter may be used to track multiple hypotheses. Hence, it may be used to solve global localization. Gaussian sum filter may be composed of *Extended Kalman Filter* or *Unscented Kalman Filter*. Both approaches are possible, in this study *Extended Kalman Filter* is chosen as the technical papers on *Extended Kalman Filter* application in localization are in vast amount.

CHAPTER 4

MAP GENERATION USING SATELLITE IMAGES

4.1 Introduction

In this chapter, the methods that are used to process the satellite images to obtain map are investigated. These images are downloaded from the Internet i.e. Google Earth Map website (http://maps.google.com). These maps which may be used for navigation and localization contain the geometrical and structural information, which explains clearly the content of map. Neural networks are used in satellite image interpretation frequently. Hence, in this chapter, neural networks are employed for mapping purposes and a novel method which is a hierarchical method, composed of Fuzzy C-Means and Adaptive Neuro- Fuzzy Inference System is developed.

Here, mapping is considered only and subject is elaborated. In order to evaluate the result numerically Kappa coefficient and Error matrix representation are used.

In the rest of this chapter, first the performance of SLAM in mapping is discussed. Then, potential use of satellite images for mapping is investigated. Next, a novel method, which may be used with satellite images for mapping purposes, is described. Finally, the soft-computing methods, including the novel approach developed in this study, are compared for their performance. Kappa coefficient and error matrix are used frequently in comparison. This chapter is concluded with a conclusion and discussion part.

4.2 A Discussion about SLAM Performance in Mapping

The most popular online algorithm in mobile robotics research is the SLAM algorithm. The original solution to SLAM problem, which is proposed by (Randall 1990), is based on Kalman Filter. It is well known that the complexity

of Kalman Filter based SLAM is $O(N^2)$ because at every update stage the filter has to update every element of the covariance matrix which is very costly. During the last decade, several optimized approaches have appeared. Compressed Extended Kalman Filter (CEKF) (Guivant 2001), Sparse Extended Information Filter (SEIF) (Thrun 2004), Relative Filter (Csorba 1997), FastSLAM (Montemerlo 2002, Montemerlo 2007) and DP-SLAM (Eliazar 2003, 2004) has reduced the computational cost in great deal. These filters have solved many practical problems. However, they are not suitable for the scenarios considered here. First, the original Kalman Filter based solution, CEKF and SEIF can hold only single hypothesis as they represent the solution to SLAM with a multidimensional unimodal Gaussian distribution which in turn poses a big problem. In long run, the filter diverges due to the linearization error and at some point; it cannot perform the right data association. As a consequence, a catastrophic filter failure could be observed. It is impossible for the filter to recover thereafter. Extended Kalman Filter is based on the Jacobian of the nonlinear process and measurement models. The convergence properties of this linearization are still unknown.

FastSLAM and DP—SLAM can be applied to the problem considered here but the problem is far more complex than the ones reported in the literature. Therefore, here the complex problem of SLAM is decomposed into a mapping problem and a localization problem with the map obtained at the first stage. This will certainly simplify the problem and make it tractable.

4.3 Potential Use of Satellite Images for Map Generation

The remote sensing technology (RST), which is a powerful tool to capture the outlook for large pieces of land across large distances through the use of various spectrum of light, has been successfully applied to a broad range of applications. In fact, RST has been used to monitor the environment and human activities i.e. urban development and the natural resources, for years. In (Isaev 2002, Park 1991) the remote sensing technique is used to monitor the forest activities and marine pollution. The estimation of the urban population is

critical as urban development has major impact on nature, water resources, and climate. Such effects cannot be detected from ground; therefore, RST is being used extensively to estimate the extent of urban development and the degree of damage to the nature (Shin 2005, Weng 2006). Agriculture is yet another field where remote sensing finds its use such as water requirement predictions (Gayan 2005) and yield estimation (Patel 2006). Remote sensing is also being used to monitor the structure of the earth (Sarup 2006, Stramondo 2006). Archeological inheritance dispersed in vast areas cannot be assessed from the ground therefore; it is being used to study archeological sites (Barnes 2003, Harrower 2002). Natural hazards are one of the hottest topics of remote sensing; remote sensing can be used to identify the damage level (Khan 2003). It is also used to identify the quality and quantity of water resources, (Pietroniro 2002).

Satellite images are becoming freely available on the Internet, and the quality of these images is increasing in time. Hence, this seems to be a rich source of information for robots as well as humans. Although various aspects of mapping problem in the mobile robotics domain is extensively studied in the literature, map generation from freely available satellite images for mobile robot navigation in urban environment problem has not been fully explored so far.

The robot navigation problem can be best solved if localization and mapping are taken into consideration simultaneously. This approach is widely known as Simultaneous Localization and Mapping (SLAM) in literature. Unfortunately, it is difficult to solve the SLAM problem in real time. In this study, a novel technique is proposed to solve the mapping problem within the context of robot localization and navigation. The outcome of this study will provide a map that can be fed to various available localization techniques like Markov Localization to localize the mobile robot.

A kidnapped robot that operates in outdoor environment can localize itself by using satellite images with appropriate localization techniques. A robot which

has access to the internet, can download the satellite images from the internet and can make use of these images for localization. Although the raw form of the satellite images are useless for this purpose, these images can be converted into a map with certain image processing techniques. This chapter is a milestone which attempt to solve the scenario described in scope of dissertation section. It proposes several techniques to convert the satellite images in a form that can be used with certain localization techniques.

Next, a novel soft computing technique is described and then it is compared with available standard soft algorithms.

4.4 A Novel Soft-Computing Technique to Segment Satellite Images for Mobile Robot Localization and Navigation

4.4.1 Introduction

In this study, the mapping is considered as a separate issue. Hence, the coupled problem of localization and mapping is reduced to a localization problem over a pre-constructed map. In the proposed technique, the satellite image of an operational urban area is processed into a meaningful form that can be used by the robot for navigation purposes.

The main goal of this subchapter is to develop a novel method that can convert (freely available) satellite images into maps

Among all of the studies listed preceding section, none of them focused on creating maps that are suitable for mobile robot navigation in urban environments. For this purpose, a mixed approach that makes use of the FCM and ANFIS is proposed. Next, these methods are introduced and their relevance to the current goal is explained.

The novel hybrid method is a combination of unsupervised and supervised techniques; the satellite images are first clustered in feature space by using an unsupervised clustering technique, namely, Fuzzy C-Means (FCM) then boundaries between different fuzzy cluster regions are drawn by an Adaptive

Network-Fuzzy Inference System (ANFIS) thus both of them constructs the structural regions in the environment useful for localization purpose

4.4.2 Classification Methods

There are several methods for segmentation. The basic segmentation methods can be grouped into two (Richards 2006), supervised and unsupervised; an unsupervised technique uses a given set of unclassified data points, whereas a supervised technique uses a data set with known classification. The classification commonly used in the literature (Richards 2006) can be listed as follows:

Supervised Classification Techniques

- Maximum Likelihood Classification
- Bayes Classification
- Minimum Distance Classification
- Mahalonobis Classifier
- Table Look Up Classification
- The kNN (Nearest Neighbor) Classifier
- Context Classification
- Spatial Context by Markov Random Fields
- Nonparametric Classification: Geometric Approach
- Linear Discriminator
- Neural Network
- **Clustering and Unsupervised Classification**
- K-means Classification
- Fuzzy C-means classification
- Hybrid Approaches

4.4.3 Proposed Technique

The methods proposed in (Lin 2000, Knoll 1997) are the starting point but the novelty of this study is that applies image classification techniques to convert

satellite images into maps for outdoor mobile robot applications. In (Lin 2000, Knoll 1997) cascaded architectures are used to segment satellite images, but there are major differences between architectures described in those studies and the one proposed here. In (Lin 2000), at the first stage of the cascaded architecture, Kohonen's self-organizing feature map is used whereas in this dissertation FCM is used to cluster the pixels. Thus the fuzzy information is not lost. In (Lin 2000) statistical features computed from co-occurrence matrix, three gray values and spectral features are used as the inputs, but in here, HSV color spectrum is used as inputs to the first stage of the cascaded architecture. In (Lin 2000) a self-constructing fuzzy neural network structure is implemented to extract and tune fuzzy rules and membership functions, here the fuzzy rules are extracted by an expert based on the clusters which are found by FCM at the first stage, then the parameters of the FIS is tuned by ANFIS. Hence the structures of the neural networks at the second stage are different.

In (Knoll 1997), the flow of the information is totally different, three fuzzy rule bases are used in segmenting the image; one of them is called *expert fuzzy rule base*, this is generated by an expert and this fuzzy rule base is not modified anymore by means of any mathematical procedure. The second one is called *automatically generated fuzzy rule base*, it is generated by a modified neural network, and this neural network structure is different from the fuzzy neural network, ANFIS, used in this study. Nevertheless, there is no explicit clustering before the raw pixels data is fed to this modified neural network structure. The final fuzzy rule base is the *fusion fuzzy rule base*, which combines the fuzzy output of the two fuzzy inferences to improve results. The structure of the single fuzzy rule base in this dissertation and the way three fuzzy rule bases are generated and implemented in (Knoll 1997) are totally different. On the other hand, the clustering is implemented in this dissertation and the one in (Knoll 1997) is different; in (Knoll 1997) clustering is used to find the cluster of interest, later which is analyzed by the three fuzzy rule base, whereas in this dissertation FCM is used to minimize the rules in the FIS and help the expert in constructing the coarse fuzzy rules

In this dissertation (Dogruer 2007), a combination of supervised and unsupervised methods is employed. The benefit of unsupervised classification method can be justified by its data compression characteristics. Hence, the supervised system running on a compact data set reduces the training time dramatically. Consequently, an Artificial Neural Network (ANN) can be trained to perform nonlinear classification efficiently. Below, the satellite image processing is discussed and a roadmap for this study is drawn.

The FCM algorithm (Richards 2006, Hertz 1991) generalizes the hard k-clustering algorithm by letting a point belong partially to multiple clusters. Therefore, it produces a soft partition for a given data set. The main advantage of this method lies in the fact that it compresses the bulk number of pixels and forms the initial clusters. Note that these clusters serve as the rough guesses for the prototypes of ANFIS membership functions which in turn reduce the number of fuzzy rules in ANFIS.

The ANFIS is constructed from Sugeno-Takagi fuzzy model (Czogala 2000, Yen 1999). It is commonly trained by hybrid learning scheme that is a combination of Least Squares and Gradient Descent Methods.

In satellite image processing a specific area has diverse characteristics and each pixel cannot be easily associated with one class with full confidence. Hence, a fuzzy logic based approach can resolve these sorts of ambiguities, so fuzzy approach is assumed.

Secondly, mass number of the pixels with diverse characteristics makes the classification problem more difficult. Therefore, an unsupervised clustering algorithm revealing the meaningful data groups autonomously is the obvious choice to compress the large data. Nevertheless, this method keeps the useful prototypes existing in the data mass. The main problem is that the clusters found in the previous stage are fuzzy. A Fuzzy Inference System (FIS) is constructed from the prototypes found in the preceding step. However, FIS

cannot be trained. At this stage, FIS must be tuned properly so an ANN may be constructed and trained. This neural network can draw the boundaries between fuzzy clusters. With the use of the trained ANN, FIS performs the classification automatically.

In the light of previous discussion, a method composed of unsupervised and supervised classifiers, namely, FCM and ANFIS is proposed. First fuzzy clustering is used to generate the meaningful fuzzy clusters where the initial clustering serves as a rough guess for the membership's functions in a FIS. In order to estimate the membership function's parameters of the FIS precisely, an ANFIS is constructed / trained by a hybrid learning scheme. The flowchart of this cascaded system is shown in Figure 4.1.

The main structure of segmentation procedure implemented here can be summarized as follows:

- A number of satellite images are collected
- The digital images are divided into two groups:
    - Training images
    - Test images
- The training images are classified by FCM clustering algorithm
- Rough FIS is constructed from the clusters found by FCM
- The ANFIS is constructed for the FIS and will be trained to compute the parameters of the FIS
- The trained system ANFIS/FIS is applied on the test images

The RGB, color plot of the satellite image pixels are classified by FCM algorithm. The clustering results show that the cluster centers are on the diagonal of the color space which suggests that the two of three color band is redundant. Therefore, HSV color spectrum feature space is used for classification and segmentation. The pseudo code map generation by remote sensing technology is given in Table 4.1. This pseudo code may be used with

all techniques used in this chapter. The pseudo code of hybrid classifier is given Table 4.2.

Note that in this dissertation, the colors are used to identify objects. Indeed, the texture of objects would improve the classification results in great deal.

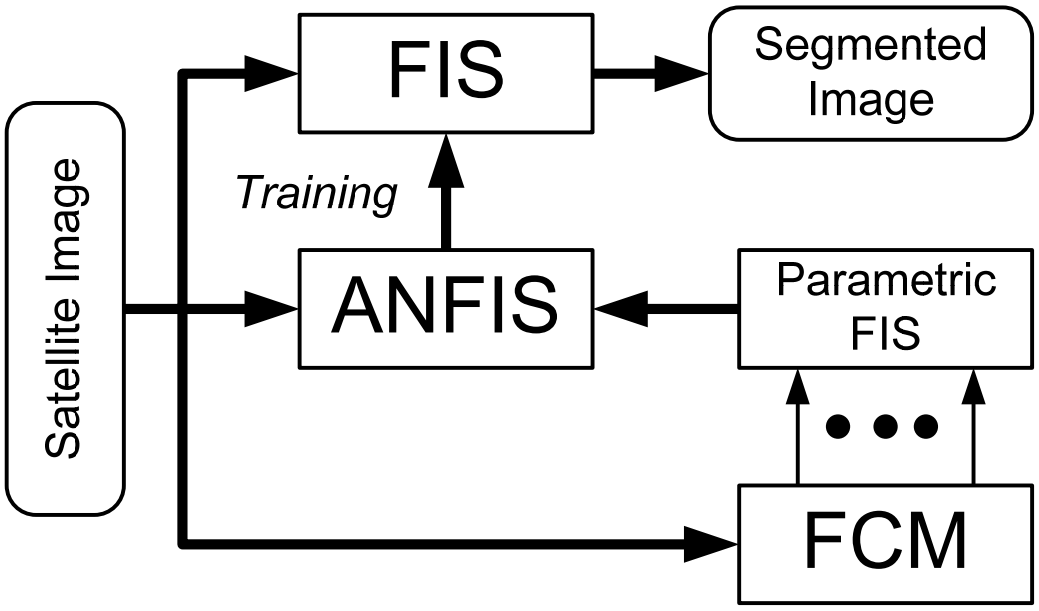


Figure 4.1 Block diagram of the cascaded system (Dogruer 2007).

However, the quality of satellite images is not good enough to display the surface textures, so surface texture is not used in this study.

4.4.4 The Structure of Hybrid Classifier and Training Scheme

First a satellite image is downloaded and analyzed by Fuzzy C-Means method so as to determine the data prototypes in the image. It is assumed that the information classes that represent the objects in the scene are fixed by a human operator. However the nonlinear mapping between the image pixel values and the information classes are not known yet. For instance, the satellite image shown in Figure 4.2 reveals that there are buildings, roads, forest, ground/soil

exist in the image. First, Fuzzy C-Means is started and the number of cluster is fixed by the human operator, note that this will certainly affect the number of Fuzzy rules in the fuzzy rule base. A fine fuzzy rule base will perform better than others but a coarse fuzzy rule base will accelerate training and image processing phases. So the number of clusters is kept minimal within the limits of process accuracy. Figure 4.3 shows the histogram of images cropped from the satellite image. Figure 4.4 shows these cropped images for different information classes: road, forest, ground, building. Examining this picture, it is concluded that some of the information class may be represented by more than one cluster: road information class has certainly two distinct pixel value range

Table 4.1 Pseudo code for map generation using satellite images

1. *Select a number of satellite images which contain the typical objects in a robot world i.e. buildings, road, vegetation, ground etc. Note that these objects are guessed by the human operator*
2. *Crop several sample images from these satellite images, these samples are selected considering the fact that their colors are typical of those distinct objects i.e. The color of a building may red or gray, vegetation is green etc.*
3. *Then a particular Neural network is trained by reinforcement learning*
    a. *The pixel value of a specific sample image in HSV color space is fed to neural network*
    b. *The information class of that specific object which is represented by a number is given to neural network as target output in a synchronized manner*
    c. *Neural network learns the nonlinear relation between the pixel value and the information class number*
4. *Determine local region by using Google Earth software running on a mobile phone.*
    a. *This may also be called crude localization*
5. *Download the relevant satellite images of the region which is determined in step 4 approximately*
6. *The satellite image is processed by the trained neural network in step (1-3)*

Figure 4.2 Typical satellite image used for training i.e. METU campus

Table 4.2 Pseudo code for Hybrid classifier

1. *The objects in sample image is plotted and the number of clusters are determined by an expert by ad hoc techniques*
2. *The satellite image is clustered by using an unsupervised method e.g. Fuzzy C-Means*
3. *The clusters are examined by an expert so as to construct the fuzzy rules. Note that at this stage the clusters are attached to information classes by the same expert.*
4. *The fuzzy rule base is converted into ANFIS*
5. *ANFIS is trained using reinforcement learning.*
    a. *ANFIS is trained by Gradient descent and Least square optimization*
        i. *The premise parameters are identified/tuned by gradient descent*
        ii. *The consequent parameters are identified/tune by least square optimization*
6. *Satellite image is processed by trained ANFIS*
7. *The output of ANFIS is a number which corresponds to a particular information class*
8. *Finally, a different color in grayscale is appointed to each number and a gray scale map is generated*

Table 4.3 Cluster centers of cropped image in HSV space

	H	S	V
Cluster 1	0.3213	0.0404	0.86683
Cluster 2	0.85916	0.062829	0.34107
Cluster 3	0.45611	0.069451	0.35763
Cluster 4	0.090153	0.097033	0.35436
Cluster 5	0.094077	0.11913	0.56598
Cluster 6	0.58175	0.1542	0.22558

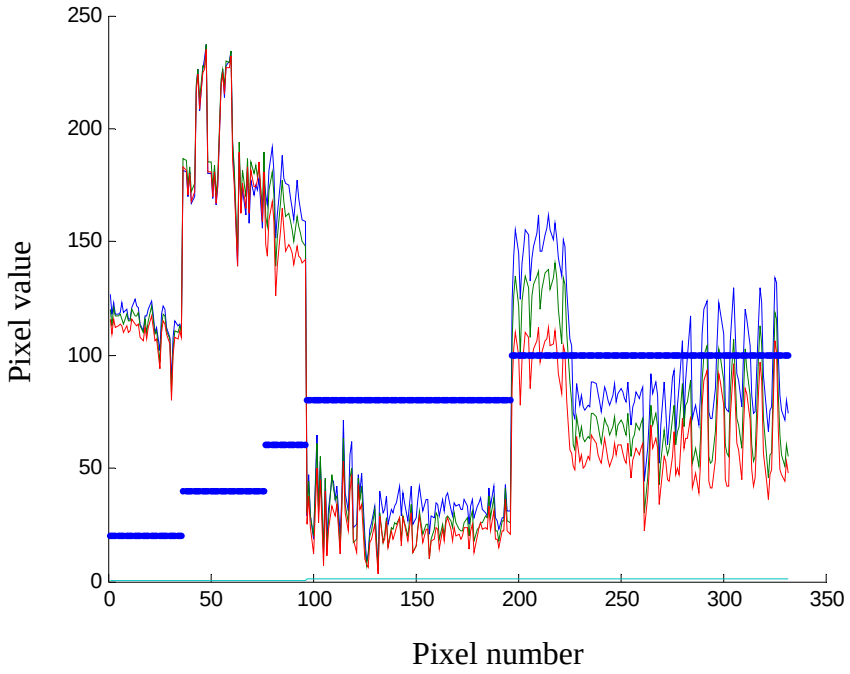


Figure 4.3 Classes: 0.2=road; 0.4=building; 0.6=road; 0.8=forest; 1.0=ground

ground information class has two distinct pixel value range. Therefore it may be reasoned that the cluster number is six. Then, FCM is run to obtain the cluster centers which are shown in Table 4.3. By using these cluster centers in HSV color space, the fuzzy rule base is constructed by a human operator. The fuzzy rule base used for this picture is shown in Table 4.4 and the corresponding membership functions are shown in Figure 4.5-4.8. Since the fuzzy inference system cannot be trained and the fuzzy rules are very coarse,

fuzzy rule base is converted into ANFIS. ANFIS can be trained, the corresponding ANFIS is shown in Figure 4.9. Note that this is not a complete diagram not to confuse the reader with junk of lines, only the first and third fuzzy rules are shown on the figure. In that figure, H S V denote channels of HSV color space; MF denotes membership function i.e. H MF1 represents the first membership function of H variable; VS denotes very small; S denotes small and M denotes medium; P denotes product; N denotes normalize; FN denotes fuzzy neuron.

Once the ANFIS which represent the fuzzy rule base is constructed, next it is fine tuned by sample images. The training images are fed to ANFIS and the known information class is introduced to network in a synchronized manner. The training of ANFIS is done by gradient descent and least square optimization; gradient descent is used to determine the premise parameters of the fuzzy rules and the least square optimization is used to optimize the consequent parameters of the fuzzy rules. After the training has been completed, the ANFIS or the fine tuned fuzzy inference system may be applied to the test image to convert them into maps whose contents are clearly identified.

Throughout this study, the information classes are represented with numbers for instance, road information class is denoted by 0.2, building information class is denoted by 0.4, forest information class is denoted by 0.8 and finally ground information class is denoted by 1. So it could be said that actually the ANFIS/FIS attaches a number to each input vector (pixel value in HSV color space). For instance FIS attaches a number very close to 0.4 if the pixel belongs to a building or it attaches a number very close to 0.8 if the pixel belongs to vegetation. Once, all image pixels are classified in this manner. A matrix whose dimension is the same as the original satellite picture is obtained; this matrix is full of numbers. Then distinct colors are allocated for each number and the matrix may be displayed in grayscale color. Note that the selection of these colors is important because it needs to be consistent the way

the laser scanner operates in the field. Laser scanner detects tree and buildings but it is insensitive to ground and road.

4.4.5 Adaptive Network Fuzzy Inference System

As an example to highlight the key points of ANFIS, a Sugeno-Takagi fuzzy model consisting of two rules only is examined. This example is taken from (Czogala 2000). This example will elaborate how a fuzzy model is converted to ANFIS and later how this network is trained. Note that the fuzzy models cannot be trained whereas ANFIS type networks may be trained by hybrid technique composed of gradient descent and least square optimization. The gradient descent is used to optimize the premise parameters of fuzzy models and the least square optimization is used to optimize the consequent parameters of fuzzy model.

Table 4.4 Fuzzy Rule base[1]

IF H is SMALL & S is VERYSMALL & V is MEDIUM
THEN Pixel belongs to BUILDING Information Class
IF H is SMALL & S is MEDIUM & V is VERYSMALL
THEN Pixel belongs to FOREST Information Class
IF H is VERYSMALL & S is MEDIUM & V is SMALL
THEN Pixel belongs to GROUND Information Class
IF H is VERYSMALL & S is VERYSMALL & V is SMALL
THEN Pixel belongs to ROAD Information Class
IF H is MEDIUM & S is SMALL & V is VERYSMALL
THEN Pixel belongs to GROUND Information Class
IF H is VERYSMALL & S is VERYSMALL & V is MEDIUM
THEN Pixel belongs to ROAD Information Class

[1] H, S, and V are the components of HSV color space.

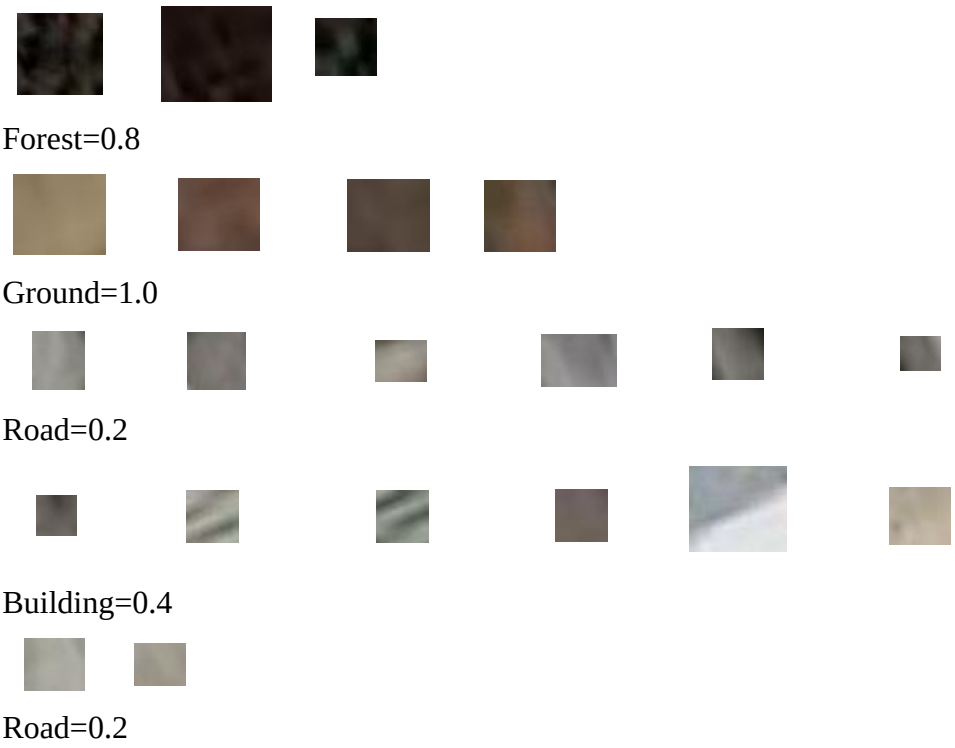


Figure 4.4 Typical training sample images for neural networks

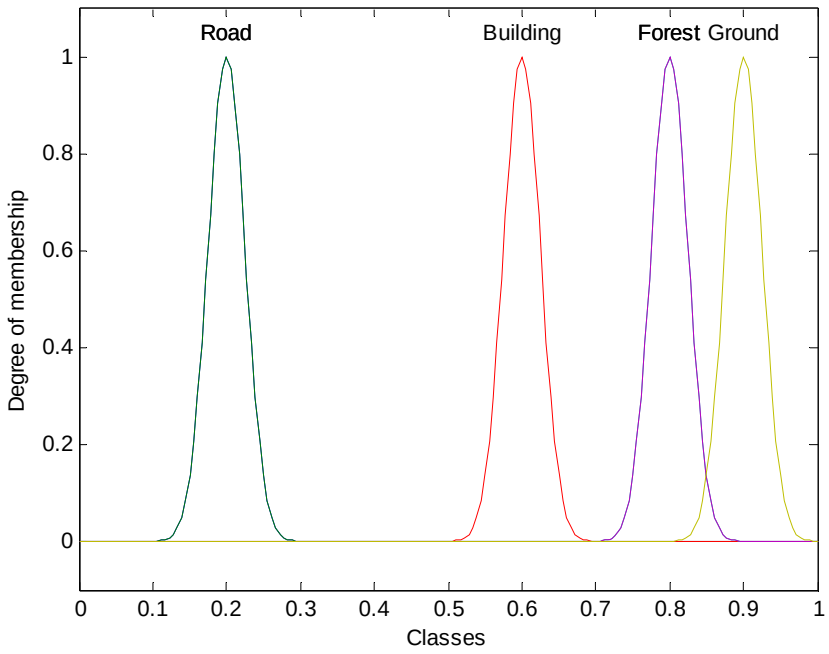


Figure 4.5 Gaussian Membership functions for information class in fuzzy model

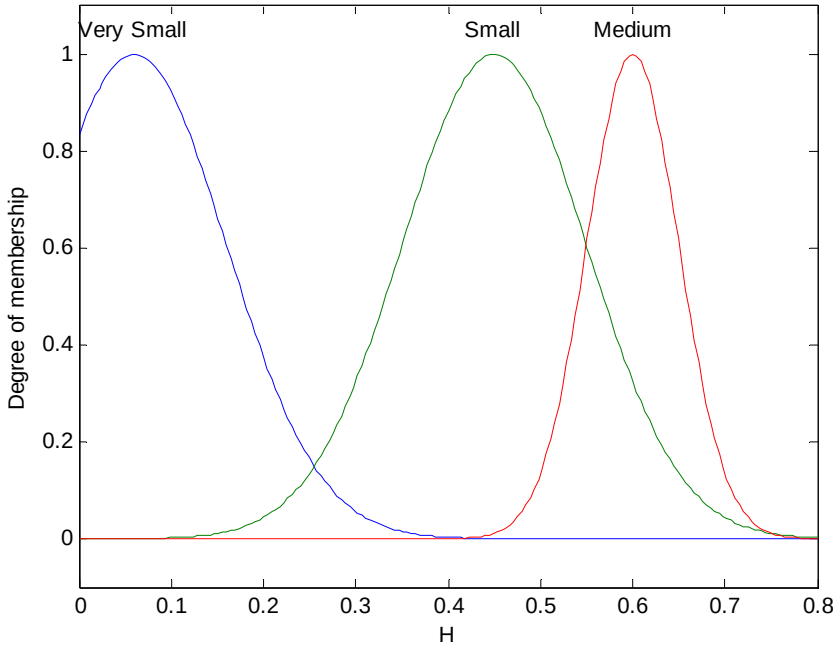


Figure 4.6 Gaussian Membership functions for H variable in fuzzy model

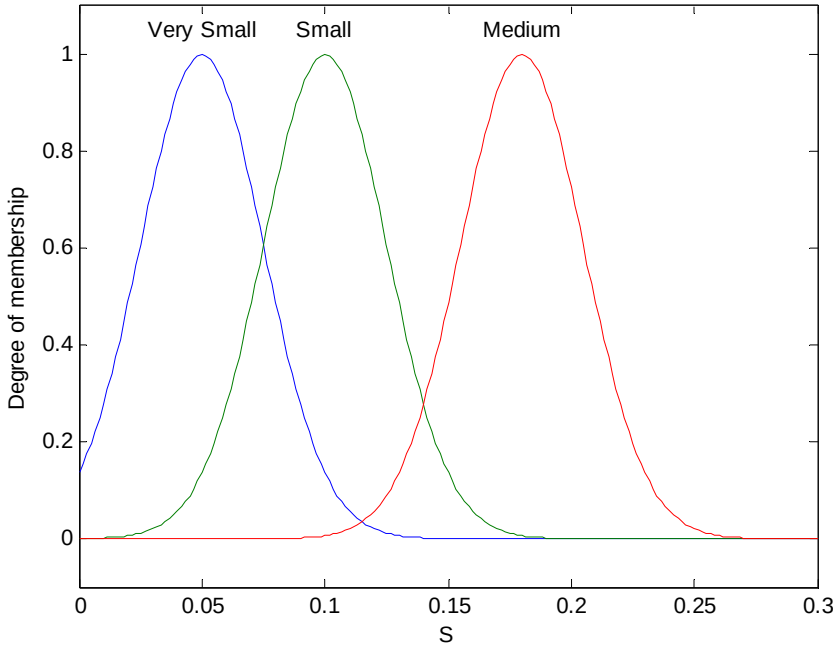


Figure 4.7 Gaussian Membership function for S variable in fuzzy model

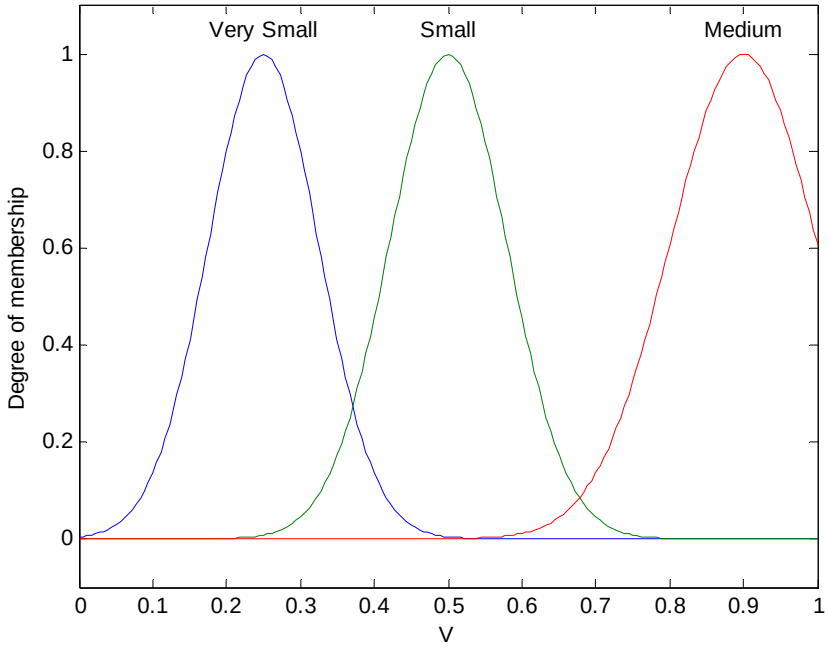


Figure 4.8 Gaussian Membership functions for V variable in fuzzy model

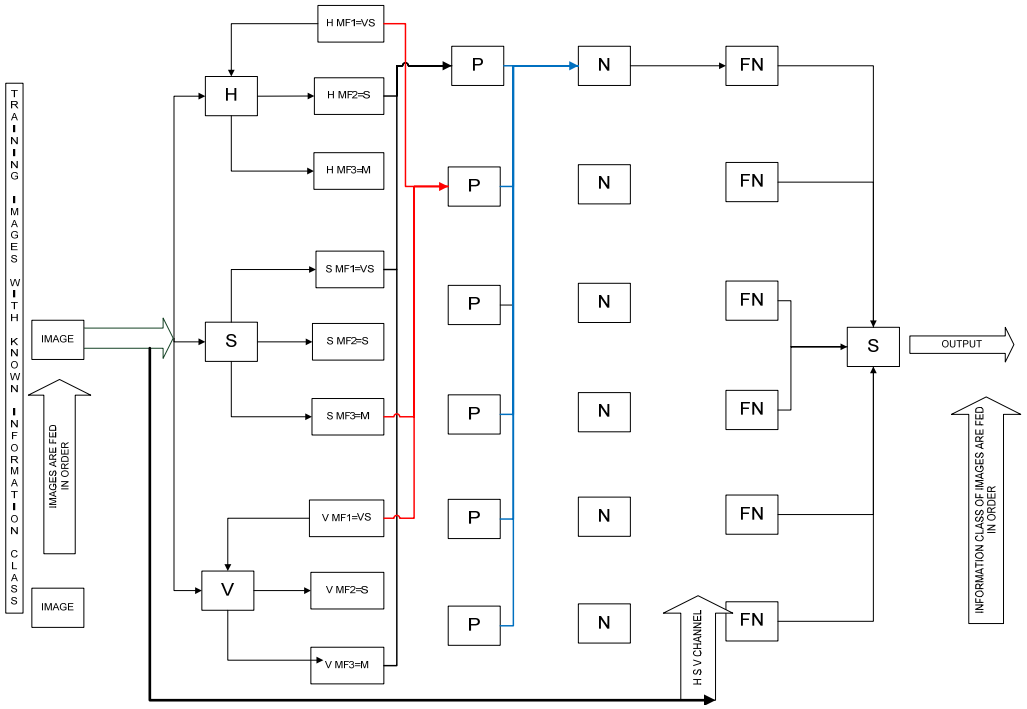


Figure 4.9 ANFIS that represents the fuzzy rule base given Table 4.4. Note that only first and third fuzzy rules are shown.

A simple two fuzzy rule is described in terms of generic membership functions A_i and B_i. The consequent part is defined in terms of $f_i(x)$ where $x = \{x_1, x_2\}$ is the vector of inputs.

R1: If x_1 *is* A_1 *and* x_2 *is* B_1 *then* $y = f_1(x)$

R2 If x_1 *is* A_2 *and* x_2 *is* B_2 *then* $y = f_2(x)$

A_i *and* B_i are fuzzy sets.

The consequent part is a linear combination of input vector $x = \{x_1, x_2\}$ and the parameters of this linear of function are $\{z_{i1}, z_{i2}, z_{i3}\}_{i=1}^{2}$

$$f_1(x) = z_{11} + z_{12}x_1 + z_{13}x_2 \tag{4.1}$$

$$f_2(x) = z_{21} + z_{22}x_1 + z_{23}x_2 \tag{4.2}$$

Note that when numerical input $x = \{x_1, x_2\}$ is presented, the fuzzy model will compute the numerical output

$$y^* = \frac{A_1(x_1)B_1(x_2)f_1(x) + A_2(x_1)B_2(x_2)f_2(x)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \tag{4.3}$$

Next a fuzzy neural network is constructed that mimics these fuzzy model, this network is shown in Figure 4.10. The observed input $x = \{x_1, x_2\}$ is presented into Layer 1 by input layer 0. The outputs of layer 1 are

$$(O_{11}, O_{12}, O_{13}, O_{14}) = (A_1(x_1), A_2(x_1), B_1(x_2), B_2(x_2)) \tag{4.4}$$

The membership functions may be selected from a family of membership functions i.e. triangular or Gaussian.

Layer 2 consists of fuzzy neurons with an aggregation operator being some t-norm. In this example the product t-norm selected in way that product is used in Sugeno-Takagi's inference procedure. The output of layer 2 are expressed

$$(O_{21}, O_{22}) = \big(A_1(x_1)B_1(x_2), A_2(x_1)B_2(x_2)\big) \tag{4.5}$$

Layer 3 normalizes the output of layer 2. The outputs of layer 3 are expressed

$$(O_{31}, O_{32}) = \left(\frac{O_{21}}{O_{21} + O_{22}}, \frac{O_{22}}{O_{21} + O_{22}}\right) \tag{4.6}$$

$$O_{31} = \frac{A_1(x_1)B_1(x_2)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \tag{4.7}$$

$$O_{32} = \frac{A_2(x_1)B_2(x_2)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \tag{4.8}$$

$$O_{3i} = \overline{w}_{\mathrm{i}} = \frac{w_{\mathrm{i}}}{w_1 + w_2} \; for \; i = 1{,}2 \tag{4.9}$$

These outputs, O_{3i} are called as normalized firing strengths

The neurons in layer 4 are fuzzy and output the following values

$$O_{41} = \left(\frac{A_1(x_1)B_1(x_2)[z_{11} + z_{12}x_1 + z_{13}x_2]}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)}\right) \tag{4.10}$$

$$O_{42} = \left(\frac{A_2(x_1)B_2(x_2)[z_{21} + z_{22}x_1 + z_{23}x_2]}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)}\right) \tag{4.11}$$

$$O_{4i} = \overline{w}_i f_i = \overline{w}_i[z_{i1} + z_{i2}x_1 + z_{i3}x_2] \; for \; i = 1{,}2 \tag{4.12}$$

Parameters in this layer are referred to as consequent parameters.

Finally the output layer computes the control action by summing the output of layer 4.

$$y^* = O_{41} + O_{42} \tag{4.13}$$

$$overall\ ouput = O_5 = \sum_i \overline{w}_i f_i \tag{4.14}$$

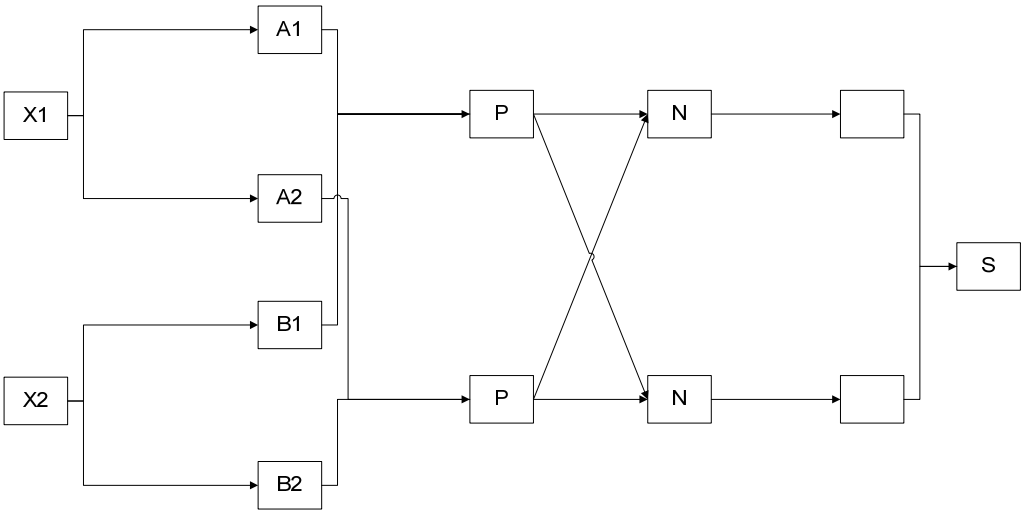


Figure 4.10 First order Sugeno fuzzy model with two rules

In order to train this network first error measure for the k^{th} training pattern is defined as

$$E^k = \frac{1}{2}(y^k - y^{*k})^2 \tag{4.15}$$

where y^{*k} is the output from the fuzzy system corresponding to the input pattern x^k, and y^k is the desired output, $k = 1,2, \dots, K$.

Now that the network is constructed and the error measure is defined, next the training of this network will be described

Hybrid Learning Algorithm

When the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters.

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{4.16}$$

$$f = \bar{w}_1[z_{11} + z_{12}x_1 + z_{13}x_2] + \bar{w}_2[z_{21} + z_{22}x_1 + z_{23}x_2] \tag{4.17}$$

$$\begin{aligned} f = {} & [\bar{w}_1 z_{11} + \bar{w}_1 z_{12}x_1 + \bar{w}_1 z_{13}x_2] \\ & + [\bar{w}_2 z_{21} + \bar{w}_2 z_{22}x_1 + \bar{w}_2 z_{23}x_2] \end{aligned} \tag{4.18}$$

which is linear in the consequent parameters

$$\boldsymbol{A} = \begin{bmatrix} \bar{w}_{11} & x_{11}\bar{w}_{11} & x_{21}\bar{w}_{11} & \bar{w}_{21} & & x_{21}\bar{w}_{21} \\ \bar{w}_{12} & x_{12}\bar{w}_{12} & x_{22}\bar{w}_{12} & \bar{w}_{22} & & x_{22}\bar{w}_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{w}_{1n} & x_{1n}\bar{w}_{1n} & x_{2n}\bar{w}_{1n} & \bar{w}_{2n} & & x_{2n}\bar{w}_{2n} \end{bmatrix} \tag{4.19}$$

$$\boldsymbol{x} = [z_{11} \quad z_{12} \quad z_{13} \quad z_{21} \quad z_{22} \quad z_{23}]^T \tag{4.20}$$

$$\boldsymbol{d} = [y_1 \quad y_2 \qquad\qquad y_n]^T \tag{4.21}$$

Where $[(x_{1k}, x_{2k}), y_k]$ are the k^{th} training pair $k = 1,2, \dots, n$ and $\bar{w}_{1k}$ and $\bar{w}_{2k}$ are the outputs of layer 3 associated with inputs (x_{1k}, x_{2k})
This matrix can be represented in matrix-vector form

$$\boldsymbol{Ax} = \boldsymbol{d} \tag{4.22}$$

In the forward pass of the hybrid learning algorithm, nodes output go forward until layer 4 and the consequent parameters are identified by the least squares method explained above. In the backward pass, the signals that propagate backwards are the error signals and the premise parameters are updated by gradient descent method.

Note that this is a simple example to illustrate how an ANFIS is constructed from fuzzy model and how this network is trained. As it was discussed before, in this dissertation the fuzzy model has six rules and three variables. The membership functions are Gaussian whose centers are roughly identified from the clusters found by an unsupervised clustering method i.e. Fuzzy C-Means.

The fuzzy rule base is first constructed by examining these clusters. The ANFIS is constructed by using MATLAB 7.0 and it is trained so as to fine tune the membership function a single example is given, the satellite image of METU campus and the relevant training images used to train the network. It should be pointed out that these training images are selected such that they represent the corresponding information class truly.

4.4.6 Simulations and Results

The satellite image of Middle East Technical University Campus is downloaded from the Web by using Google Earth program. The sample image is shown in Figure 4.11 As it is discussed before, the RGB color space is not a good choice to discriminate among different segments of the picture by comparing color codes. Therefore, the image is converted into HSV color space. The FIS optimized for training images by cascaded system is then applied to the image. The segmented image is shown in Figure 4.12. First, a number of samples are cropped from the image as shown in Figure 4.13. These samples are plotted in HSV color dimensions where the distinct cluster centers are found by FCM Technique. The structure of the fuzzy rule base is also constructed from the clusters found by FCM. This feature helps minimizing the fuzzy rules number in the knowledge base. The ANFIS is constructed for the FIS in order to optimize the parameters of the Gaussian membership functions. Lastly the ANFIS is trained by hybrid learning. The ANFIS learns the classification which is shown in Figure 4.14. After training period, the parameters of the ANFIS representing a parametric form for FIS are fixed.

The whole image is ultimately processed by the trained ANFIS/FIS and is segmented into four regions: buildings, roads, forest and ground. The four classes exist in the thematic map and the FIS results are presented in Figure 4.14 for comparison. The classified image is shown in Figure 4.15. The classification of the sample satellite image is found successful; however there are some misclassifications owing to the fact that the color codes of some objects are very difficult to distinguish from each other. This is especially true

for the color codes of the roads and the roofs of some building which are almost identical in some parts of the picture. This misclassification of the color codes may depend on the particular region being analyzed.

In the next classification, the satellite image of a more compact region, Figure 4.15 (a) is classified as shown in Figure 4.15 (b). In this part of the study, the classification done by the computer is compared to that of a human expert. Thus, the results are presented in the form of a matrix in Table 4.5. As can be seen, the results are quite satisfactory since the Kappa Coefficient is computed as 0.5448. Although there are some misclassifications, both classification jobs are considered as successes since the major components of the map (such as the roads and the large buildings) are identified clearly. However, the weak side of the proposed paradigm is that it depends heavily on the color spectrum of the objects in the image. Indeed, this is not sufficient to extract the maximum information out of the image alone since some objects may share similar color characteristics. That is, there may have been some places where these feature space elements are almost the same for different objects. For such cases, fine granular fuzzy rule base may be needed for better results.

In this section, a novel technique is proposed to generate digital map for mobile robot localization and navigation, from satellite images. The technique is a combination of unsupervised, Fuzzy C-Means, and supervised, ANFIS, methods.

The sample satellite image downloaded from web is segmented thru a cascaded system of Fuzzy C-Means and ANFIS. If compared to the similar techniques employed in image analysis/segmentation; the performance of the segmentation is quite acceptable. For instance, in (Wilkinson 2005), a large number of satellite image classification publications are surveyed and the average Kappa coefficient and the standard deviation for these applications are calculated as 0.6448 and 0.1980, respectively. Considering that the Kappa coefficient for the particular segmentation shown in Figure 4.15(b) is 0.5448;

Figure 4.11 Comparison of the digital map and the satellite image, Original satellite image in RGB color space

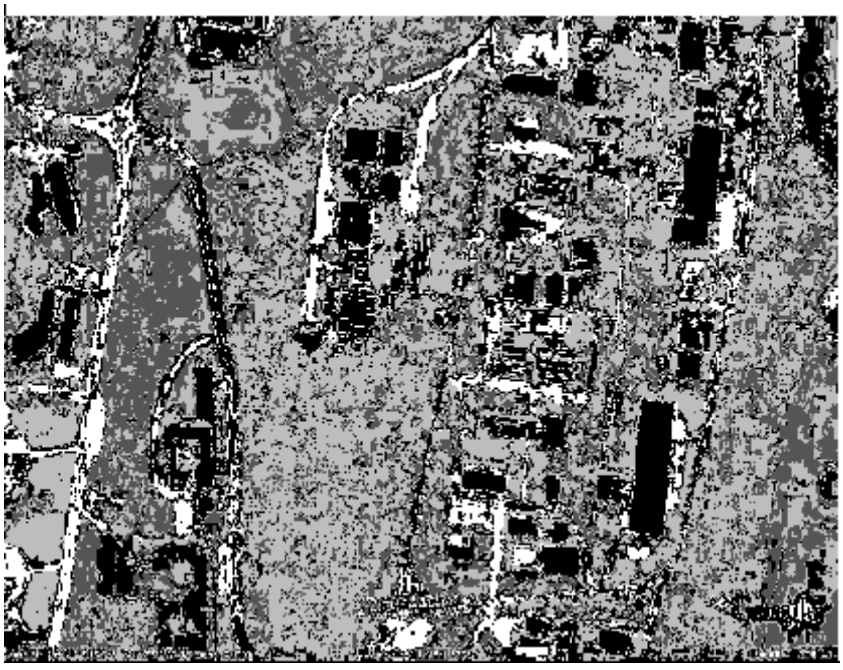


Figure 4.12 Comparison of the digital map and the satellite image; the digital map segmented by FIS

Figure 4.13 Few samples cropped from the original satellite image to train the proposed technique The sampled part of the satellite image are shown by white rectangles

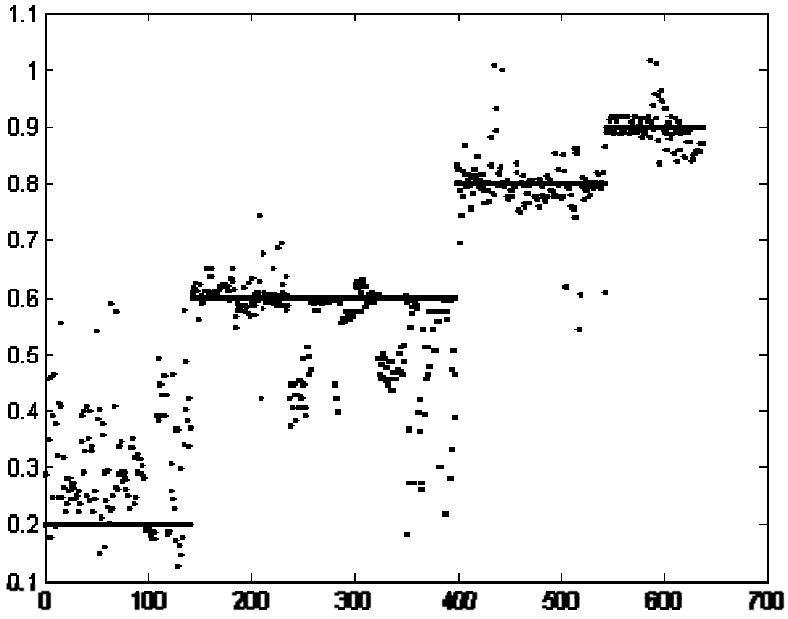


Figure 4.14 Classes exist in the image and the approximation of these classes by FIS

the proposed method is found to be promising. Note that the use of digital maps shouldn't be evaluated quantitatively alone; the quality of map used in localization must be rigorously evaluated thru detailed simulations. That is, the Kappa value for a particular segmentation could be theoretically high. On the other hand, the generated map might turn out to be too "noisy" to yield meaningful results in localization.

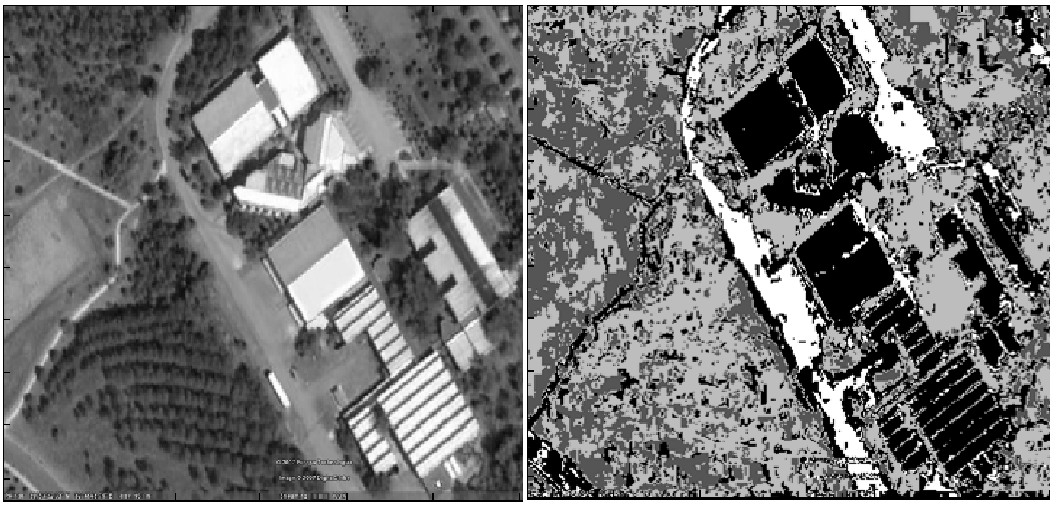
4.5 A Comparison of Alternative Soft-Computing Methods to Segment Satellite Images for Outdoor Mobile Robot Mapping

4.5.1 Introduction

In previous section, a solution is proposed to create digital maps from freely available satellite images. That is, the satellite image of an urban area is processed into a meaningful form that can be used by the robot for navigation purposes. The analysis of satellite images is done by a combination of unsupervised and supervised techniques; namely, Fuzzy C-Means (FCM) and Adaptive Network-Fuzzy Inference System (ANFIS). Here, this novel technique is compared with available soft-computing techniques for their performance in creating maps; Feed-Forward Neural Network (FNN), Fuzzy-C

Table 4.5 Error Matrix of the Classification

		Ground Truth Classes			
Thematic Map Classes		Building	Road	Ground	Forest
	Building	0.9746	0.2819	0.1669	0.1287
	Road	0.0211	0.7012	0.0244	0.0027
	Ground	0.0002	0.0046	0.4651	0.1172
	Forest	0.0041	0.0123	0.3436	0.7514
	# pixels	14717	17020	38448	63884



(a) Original satellite image (b) Segmented satellite image

Figure 4.15 Segmentation of a local region, used in the performance metric

Means (FCM), and Probabilistic Neural Network (PNN). Kappa coefficient, which is a common metric in remote sensing field, and error matrix, are used in performance comparison.

4.5.2 Background

Maps can be created from satellite images by classifying them into various meaningful components with the utilization of various supervised/unsupervised segmentation techniques. In this subsection, the novel method (Dogruer 2007) (a.k.a. “hybrid method”), whose functional diagram is illustrated in Figure 4.1, is considered for mapping purposes and its performance is compared to those of the popular segmentation techniques in the literature. That is,

- Supervised Classifications
    - **a.** Feed-forward Neural Network (FNN)
    - **b.** Probabilistic Neural Network (PNN)
- Unsupervised Classifications
    - **a.** Fuzzy C-Means (FCM)

The supervised and unsupervised classification methods used in this study are described in detail (Richards 2006, Hertz 1991, Czogala 2000). In fact, the FNNs are successful in approximating nonlinear functions (Hertz 1991) while FCM (Richards 2006, Czogala 2000) can both handle noisy data and identify the patterns in bulk data. On the other hand, the PNN can be employed in classification problems as it boils down to a Bayesian classifier when sufficient training data are provided. Hence, the aforementioned methods can be all used to approximate any nonlinear relation between a pixel value at a specific point in the image and its corresponding class/segment number, i.e. an integer number assigned to a particular class.

In supervised classifications, the system is trained by samples cropped from multiple fields whose attributes are diverse in nature. Four types of classes are fixed by an expert and for each class, the sample images are cropped from a satellite image as demonstrated in Figure 4.13. These four classes are buildings, roads, forest / vegetation, and ground.

Unsupervised classification method is applied to an image so as to extract the distinct patterns autonomously. A human expert usually determines the number of the classes by adhoc techniques. Then, the whole test image is processed by unsupervised classification method to establish the cluster centers and the corresponding memberships. Since the FCM method finds the degree of membership to each existing fuzzy cluster; the input pixel is associated with the nearest fuzzy cluster using a Euclidean distance metric.

4.5.3 Training of Supervised and Unsupervised Artificial Neural Networks

Fuzzy C-Means

In pattern recognition and image processing, unsupervised clustering is often used to perform the task of segmenting the images (i.e. partioning pixels on an image into regions that corresponds to different objects or different faces of an object in the images).

Conventional clustering techniques find hard partition of a given dataset based on a certain criteria evaluate the goodness of partition. By hard partition it is meant that each datum belongs to exactly one cluster of partition.

A soft clustering algorithm finds a soft partition of a given set of data set based on a certain criteria. In a soft partition, a datum can partially belong to multiple clusters. The Fuzzy C-Means, which is best known fuzzy clustering algorithm, produces a constrained soft partition.

The Fuzzy C-Means algorithm (FCM) generalizes the hard c-means algorithm to allow a point to partially belong to multiple clusters. Therefore it produces a soft partition for a given dataset. In fact it produces a constrained soft partition. To do this the objective function of the hard C-Means algorithm is modified in two ways:

1. the fuzzy membership degrees in clusters were incorporated into the formula
2. and additional parameter m was introduced as a weight exponent in the fuzzy membership

The extended objective function denoted by J_m, is

$$J_m(P,V) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \left[\mu_{ci}(x_j)\right]^m \left\|x_j - v_i\right\|^2 \tag{4.23}$$

where P is a fuzzy partition of the dataset X formed by $C_1, C_2, \ldots\ldots, C_k$. The parameter m is a weight that determines the degree to which partial members of a cluster affect the clustering result.

Like hard c-means, Fuzzy C-Means also tries to find a good partition by searching for prototypes v_i that minimizes the objective function J_m. Unlike hard c-means, however, the fuzzy c-means algorithm also needs to search for membership functions μ_{ci} that minimizes J_m. To accomplish these two objectives, a necessary condition for local minimum of J_m was derived from

J_m. This condition which is formally stated below serves as the foundation of the fuzzy c-means algorithm. The pseudo code of FCM is given in Table 4.6.

A constrained fuzzy partition $\{C_1, C_2, \ldots\ldots, C_k\}$ can be local minimum of the objective function J_m only if the following conditions are satisfied.

$$\mu_{Ci}(x) = \frac{1}{\sum_{j=1}^{k} \left[\frac{\|x - v_i\|^2}{\|x - v_j\|^2} \right]^{\frac{1}{m-1}}} \tag{4.24}$$

$1 \leq i \leq k, x \in X$

$$v_i = \frac{\sum_{x \in X} [\mu_{Ci}(x)]^m x}{\sum_{x \in X} [\mu_{Ci}(x)]^m} \tag{4.25}$$

$1 \leq i \leq k$

Feed-Forward Neural Network

Feed-Forward Neural Network is known for their power in function approximation, so they may be safely used for nonlinear mapping of pixel values to some discrete output values which represents the numerical codes of different information classes. For instance, the information classes may be coded as Building=0.2, Road=0.4, Forest=0.8 and Ground=1.0. Then the training images are fed one by one and the network is forced to output the corresponding discrete numbers for them. The feed-forward neural network topology used in this dissertation is shown in Figure 4.16. There are three hidden layers and one input and one output layers in this topology. The input layer has three components of the HSV color of pixels. The first hidden layer has 6 neurons, the second and third hidden layer has twelve neurons and the output layer has a single neuron. The transfer function of neurons in hidden layers are tangent sigmoid function, the output layer has a linear transfer function. The feed-forward neural networks are trained by back propagation. In this dissertation Levenberg-Marquet method is used to optimize the neural

Table 4.6 Pseudo code for Fuzzy C-Means (FCM) algorithm

$FCM(X, c, m, \varepsilon)$

X: *an unlabeled data set*

c: *the number of clusters to be formed*

m: *the parameter of objective function*

ε: *a threshold for the convergence*

Initialize prototype $V = \{v_1, v_2, \ldots, v_c\}$

Repeat

$V^{previous} \leftarrow V$

Compute the membership function

Update the prototype

Until $\sum_{i=1}^{c} \left\| v_i^{previous} - v_i \right\| < \varepsilon$

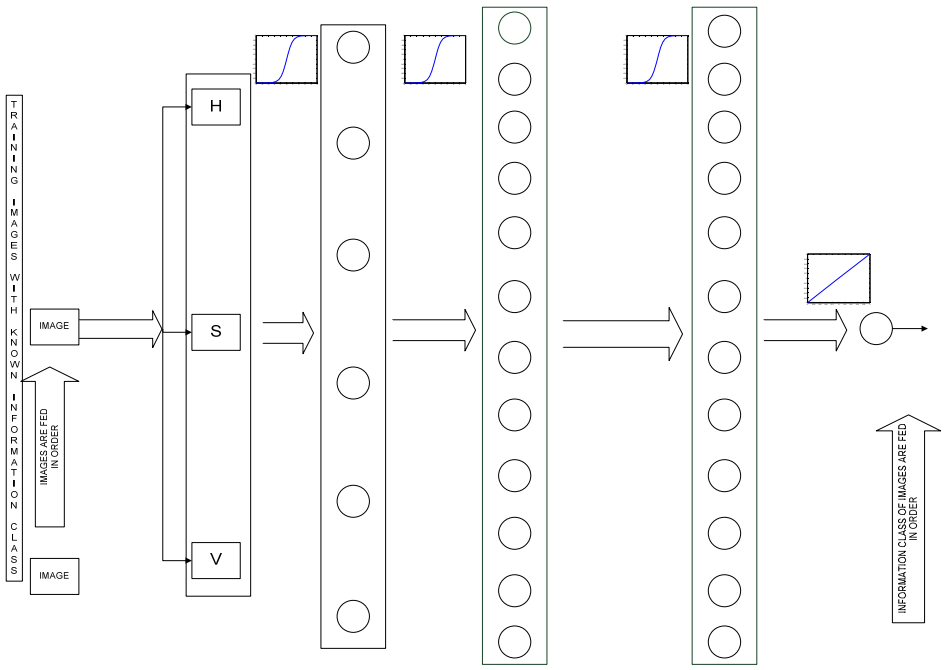


Figure 4.16 Topology of Feed-Forward Neural Network used in image processing

network weights. The network topology is shown in Figure 4.16; note that all neurons in successive layer are fully connected. However, only a small number of connections are shown not to confuse the reader and make the graph unnecessarily complicated.

A typical generic feed-forward network topology is presented in Figure 4.17 to illustrate the back-propagation graphically. First signals are propagated through the network, see Figure 4.17. Finally the delta for the network output is computed and then deltas for the preceding layer are computed by back propagating the errors backwards, see Figure 4.17-19.

Then weights are updated as follows

$$\Delta w_{ij} = \eta \delta_i^m V_j^m \tag{4.26}$$

where δ_i^m is delta of i^{th} neuron in m^{th} layer and V_j^m is the output of j^{th} neuron in $(m\text{-}1)^{th}$ layer and Δw_{ij} is change in the weight between j^{th} neuron and i^{th} neuron

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \tag{4.27}$$

Probabilistic Neural Network

Probabilistic Neural Network (PNN) is classifier which converges to optimal Bayes classifier when enough number training data is supplied. In classification theory an unknown x may classified into a particular class i, if the PDF of each population is known. For instance

$$f_i(X) > f_j(X), all\ j \neq i \tag{4.28}$$

Where

$f_k\ is\ the\ PDF\ of class\ k$

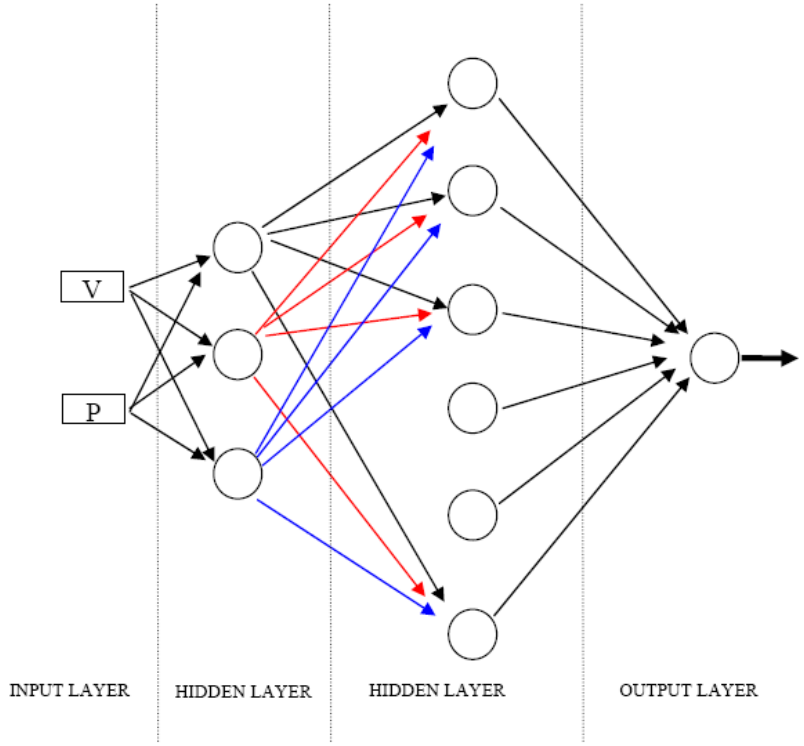


Figure 4.17 Input signals are propagated forwards

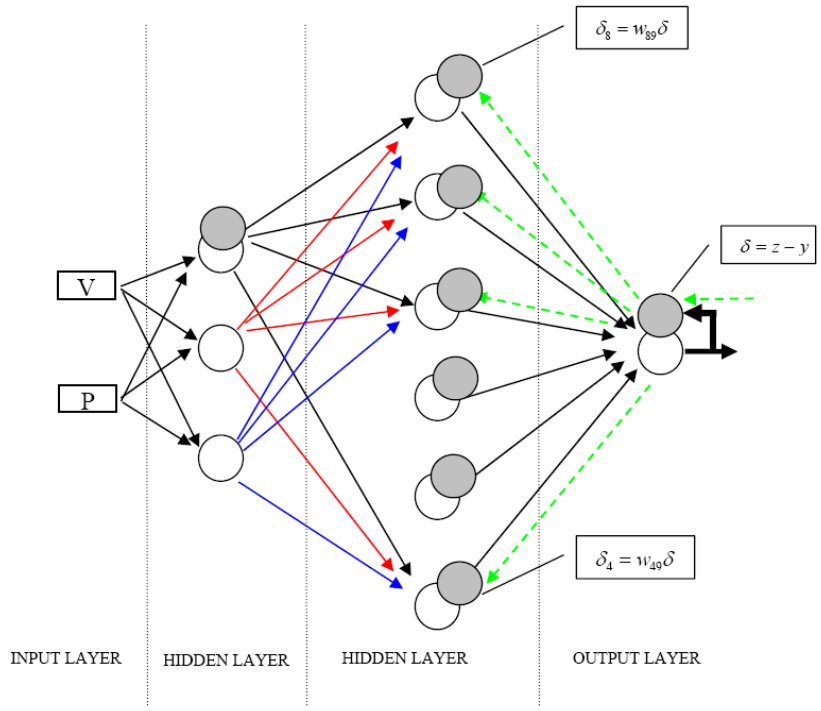


Figure 4.18 Error signals are propagated backward to the second hidden layer

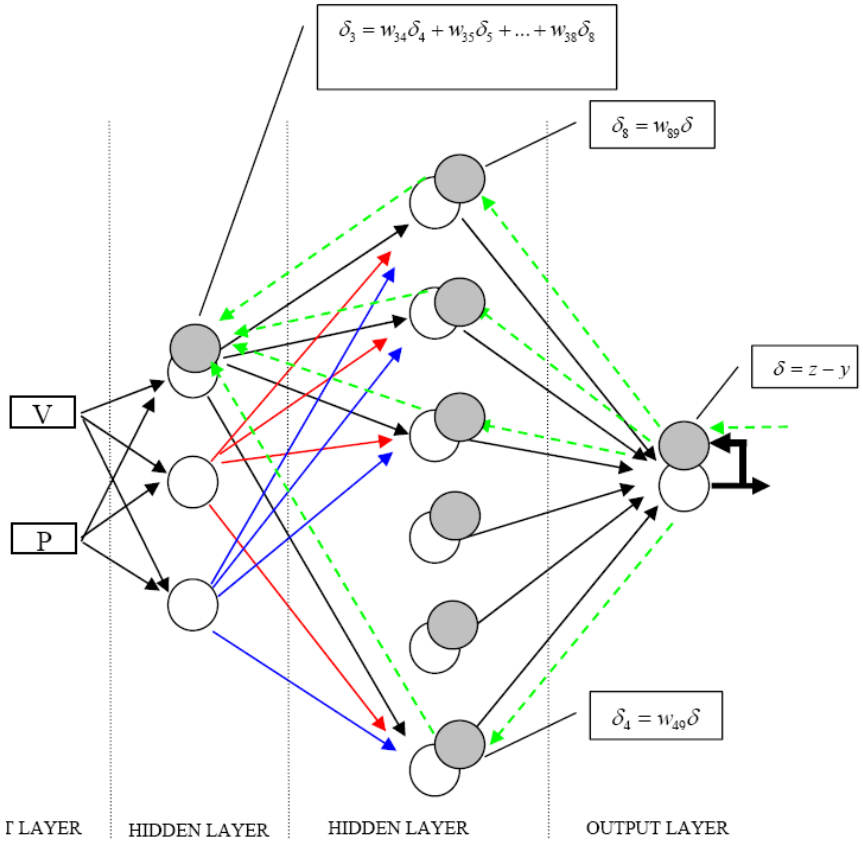


Figure 4.19 Error signals are propagated backward to the first hidden layer

For multivariate inputs the following PDF is used

$$f_i(X) = \frac{1}{n_i}\sum_{k=1}^{n_i}\frac{\|X - X_{ik}\|}{\sigma^2} \tag{4.29}$$

$$f_i(X) > f_j(X), all\ j \neq i$$

In PNN the pattern layer computes the degree of membership of input vector to each pattern then the output of pattern layer are pass on to summation layer where average of all patterns connected to that category is computed. Note that the pattern layer neurons are not fully connected to neurons in summation layer see Figure 4.20 for the topology of PNN network. Particular set of neurons

whose transfer function is a multivariate Gaussian PDFs are connected to a certain category. In this way an average of patterns are computed. The neurons in summation layer compete with each other so as to select best category that is assigned to input vector. The winning neuron publishes its category as output. A generic topology of PNN is shown in Figure 4.20. The first layer is the input layer where the inputs are the HSV colors of each pixel; the summation layer has four categories as Building (B), Road (R), Ground (G) and Forest (F). In the actual case each category is assigned an integer number in the range 1-4. The pattern layer has 163 neurons with Gaussian transfer functions and analysis of weight matrix also proves that four set of neurons in pattern layer each of them are connected only a single category. No single neuron is connected to more than one category. The training of PNN is a supervised training where the training images and the corresponding information class are supplied to the network simultaneously. Finally the PNN learns how to map the pixel into a particular category by processing the pixel colors. In this dissertation the PNN is constructed by using MATLAB 7.0, the *newpnn* command is used for this purpose.

4.5.4 Methodology

Various supervised and unsupervised classification techniques as discussed briefly in the previous subsection are taken into consideration. To test the above-mentioned algorithms, the satellite images of various locations in Ankara City are downloaded from the web by using Google Earth program. One of these images is illustrated in Fig 4.24.

The classification results are compared by using a matrix representation. It is customary to present the classification results in the form of error matrices. This error matrix may later be used to compute a single scalar coefficient. For instance, (Richards 2006) suggests the use of a metric called Kappa coefficient to evaluate the degree of conformity of a classified image to a reference one representing the ground truth. That is, the Kappa coefficient may be expressed as:

$$\kappa = \frac{P\sum_k x_{kk} - \sum_k x_{k+}x_{+k}}{P^2 - \sum_k x_{k+}x_{+k}} \tag{4.30}$$

$$x_{i+} = \sum_j x_{ij} \tag{4.31}$$

$$x_{+j} = \sum_i x_{ij} \tag{4.32}$$

where P is the total number of observations in the error matrix, and x_{ij} are the element of this matrix. Note that as Kappa takes higher values closer to one, the segmentation is better whereas Kappa takes lower values, segmentation is worse. The diagonal elements of error matrix denote the number of pixels classified into true Ground Truth Class and the off-diagonal elements in a particular column show the number of pixels, which are misclassified into wrong Ground Truth Classes. For instance, in Table 4.7, the first column shows that 60.45% of building pixels are classified into building Ground Truth Class correctly and second column shows 3.56% of road pixels are misclassified into ground Ground Truth Class. Note that in this study each column of Error Matrix is normalized with respect to the total number of pixels in that Ground Truth Class, so elements of each column is a percentage of total number of pixels in that particular Ground Truth Class.

Lastly, the Kappa Coefficient computed here is compared to the one computed in (Wilkinson 2005) where large number of publications on satellite image classification, published in the last decade, are surveyed.

4.5.5 Benchmarking

In this part of the study, the classification done by the computer is compared to that of a human expert. The performance of hybrid method is investigated by comparing the results to other soft-computing methods in the form of error matrices and Kappa coefficients. Two local sites in Ankara City are used for benchmarking. The first sample image that illustrates a small portion of the city

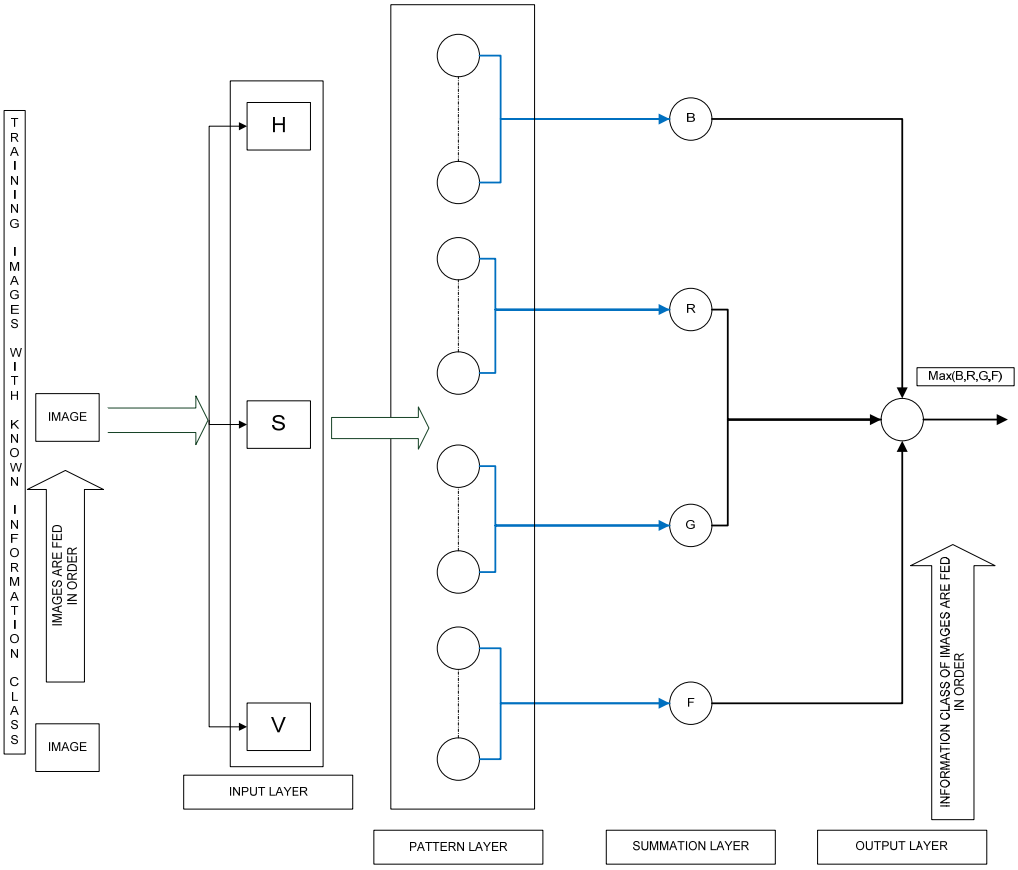


Figure 4.20 Topology of PNN, only limited number of connections are drawn

is shown in Figure 4.21. Since the RGB color space is not a good choice to discriminate amongst different segments of the picture, the image is converted into the HSV color space and the rest of the training/segmentation analyses are done in this color space.

Similarly, the supervised methods are trained by the sample images cropped from the satellite image of the city district, shown in Figure 4.21. The white rectangle areas in Figure 4.21 demonstrate the cropped training samples. During segmentation, the performances of the trained algorithms are tested on the sampled image whereas the unsupervised method is applied to image to extract the distinct data groups autonomously. The results of the automatic segmentation are compared to the manual segmentation done by the Adobe

Photoshop (Figure 4.22), Here, Thematic Map Classes are the ones found by automatic segmentation and the Ground Truth Classes refer to the correct segmentation made manually

Table 4.7 Error Matrix for FNN ($\kappa = 0.3731$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.6045	0.1157	0.1453	0.1858
	Road	0.1739	0.7475	0.3594	0.2481
	Forest	0.1409	0.1012	0.4111	0.2436
	Ground	0.0807	0.0356	0.0842	0.3226

As a baseline case highlighted in Table 4.7, a human expert classifies the test image pixels into 4 categories: Building (54124), road (85094), forest (66542), and ground (34232) where the numbers in parenthesis refer to the number of pixels belonging to that category. The resulting image is illustrated in Figure 4.23(b). Then, each column of Error Matrix is normalized with respect to the total number of pixels in that Ground Truth Class, so elements of each column is a percentage of total number of pixels in that particular Ground Truth Class.

First, the test image shown in Figure 4.21 is processed by the aforementioned soft-computing methods. The test image, which is classified by FNN, is illustrated in Figure 4.23 and the related error matrix is presented in Table 4.7. The Kappa coefficient which shows the overall correlation between automatic segmentation and manual segmentation / ground truth image is 0.3731. Raw classification by FNN is filtered with median filter to remove the noise in image; the filtered image is shown in Figure 4.26(b). As can be seen, all the distinct features (buildings, roads, ground and forest) are identified, but there

are some misclassifications due to the similarity of the color spectrum of different thematic map classes. Close examination of the related error matrix reveals some facts about the classification success. 17% of the building's pixels are misclassified for road Ground Truth Class, 14% of buildings pixels are misclassified for forest Ground Truth Class and 8% of buildings are misclassified as ground Ground Truth Class. 36% of road pixels are misclassified for ground Ground Truth Class, 36% forest pixels are misclassified for road Ground Truth Class. Ground pixel classification is the least successful one; almost 70% of the ground is misclassified for remaining three Ground Truth Classes, evenly.

Secondly, the same test image is classified by PNN, the segmented image is illustrated in Figure 4.24 and the results are given in Table 4.8. The Kappa coefficient is 0.4810. 24% of building pixels are misclassified for ground Ground Truth Class, 24% of building pixels are misclassified for forest Ground Truth Class, 24% of road pixels are misclassified for the ground Ground Truth Class, 20% of forest pixels are misclassified for road Ground Truth Class, %14 of forest pixels are misclassified for ground Ground Truth Class. The highest misclassification is associated with ground class, 22% of ground pixels are misclassified into the building Ground Truth Class.

Next, FCM classification method is taken into consideration. As stated before, it is directly applied onto the test image to extract the distinct clusters autonomously. The optimum number of clusters is found to be eight, by trial and error. There are one building, three road, two ground, two forest clusters. The unsupervised clustering technique finds the patterns in the data group automatically but the data clusters and the information classes has to be edited by a human expert. FCM is applied to the image; the results are tabulated in Table 4.9 where the Kappa coefficient is 0.5791. The analysis of the error matrix reveals that the FCM is more successful in detecting building and road thematic map classes than FNN and PNN. However, it cannot outperform FNN

and PNN in detecting the ground class since 47% of the ground pixels are misclassified for building Ground Truth Class.

Last method is the hybrid method proposed by (Dogruer 2007). The test image, which is classified by this method, is illustrated in Fig 4.25 and the related error matrix is presented in Table 4.10. Figure 4.28 indicates that all the distinct features (buildings, roads, ground and forest) are identified with $\kappa = 0.5403$. This value is higher than those of FNN and PNN and is very close the highest Kappa value computed so far. The elements of the related error matrix highlight some facts about the segmentation. 22% of building pixels are misclassified for road Ground Truth Class, 13% of roads are misclassified for forest Ground Truth Class, 25% of forest pixels are misclassified for road Ground Truth Class. Ground classification is the least successful one in this case: 27% of the ground is misclassified for road and 33% of ground pixels are misclassified for forest Ground Truth Class.

Finally, it is observed that the most problematic thematic map class for this image is ground class. There are high rates of misclassification of ground for buildings. This can be best explained with color spectrum of a typical area. It is seen that the color spectrum of sampled building and the ground are almost the same. Actually, this method is likely to fail when the color spectrum of different Ground Truth Class are similar.

The second test image is illustrated in Figure 4.26. This image is segmented with the same supervised and unsupervised methods. As a visual example, the image segmented with novel method is illustrated in Figure 4.27. The segmentation results for FNN, PNN and FCM are presented only in the form of error matrices and Kappa coefficients. Table 4.11, Table 4.12 and Table 4.13 illustrates the results in the form of error matrix for FNN, PNN and FCM respectively while the error matrix for novel hybrid method is given in Table 4.14. The four ground truth image classes for second image which are created

Figure 4.21 Satellite image of Ankara city district and the ground truth image, white rectangles shows the cropped samples.

Figure 4.22 Satellite image of Ankara city district and the ground truth image

by an expert: Building (146680), road (105925), forest (54595), and ground (36009)where the figures in parenthesis refer to the number of pixels belonging to that category.

The Kappa coefficient for FNN is the highest which is 0.5942. The highest misclassification for this method is associated with ground truth class where 23% ground pixels are misclassified for forest Ground Truth Class. The Kappa coefficient for PNN is 0.5107. For PNN, the followings can be deduced from error matrix; 30% of forest pixels are misclassified for ground Ground Truth Class and 20% of ground pixels are misclassified for building Ground Truth Class. The Kappa coefficient for FCM is 0.4716. For FCM, the followings can be observed: 22% of forest pixels are misclassified for building Ground Truth Class and 40% ground pixel are misclassified for forest Ground Truth Class. Table 4.14 shows that the performance of novel method is close the FNN which is the most successful method considered for this benchmark problem so far. The Kappa coefficient for novel method is 0.5108.

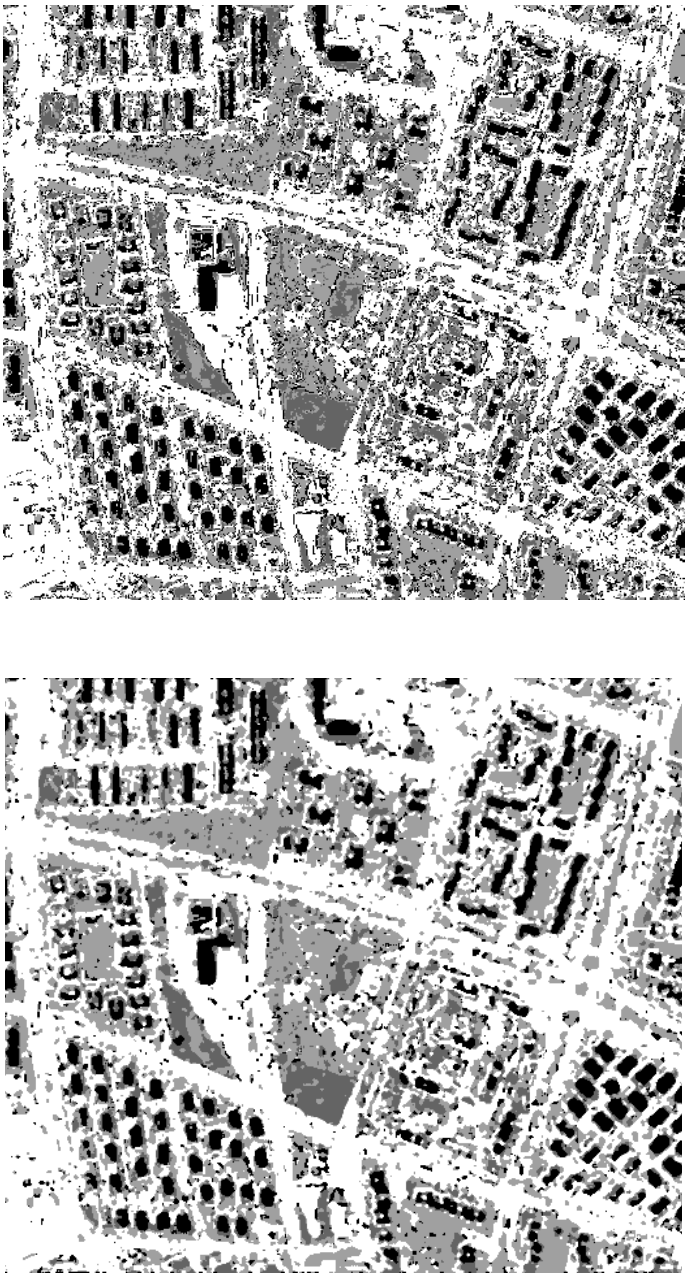
The Kappa coefficients for the tested classification techniques used on the first image are; FNN (0.3731); PNN (0.4810; FCM (0.5791), Hybrid Method (0.5403), for second test, the FNN (0.5942); PNN (0.5107); FCM (0.4716) and Novel Method (0.5108). The computed coefficients Kappa coefficients are comparable to the Kappa coefficient found in (Wilkinson 2005) whose mean is 0.6557. So the preliminary results are not in contradiction with previous works in the literature, but these results must be elaborated in the field with a mobile robot and mobile robots models operating virtual environments to improve their success. The challenge for these segmentations is that they are useful for localization and navigation purposes. However the Kappa coefficient used so far gives information about the overall performance. The existing noise terms may lead to serious errors and it does not provide any insight about the success of a digital map constructed to be used in mobile robot applications. The error matrix is more useful in such an evaluation. Lastly, Table 4.15 shows the total segmentation time for automatic methods and manual classification. The

manual segmentation is very tedious and lengthy. The supervised method spend most of the time for sampling the image, the training and segmentation does not need an operator, nevertheless their contribution to the total segmentation time is negligible when compared to the manual classification. Unsupervised classification is definitely the fastest.

Table 4.8 Error Matrix for PNN ($\kappa = 0.4810$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.6643	0.0619	0.0114	0.2204
	Road	0.0684	0.6081	0.1997	0.1045
	Forest	0.0235	0.0904	0.6477	0.1235
	Ground	0.2439	0.2396	0.1412	0.5515

Satellite images can be classified with many techniques among which the following are considered for evaluation in this work: hybrid technique, FNN, PNN, and FCM. Their performances are rigorously investigated through a number of images. In fact, these methods have the potential to convert satellite images into maps that are suitable for mobile robot applications. Sampling/test procedures for these techniques definitely play a key role in their performances. Supervised techniques are trained by samples cropped from a large area whose characteristics are thought to be distinctive. On the other hand, the unsupervised technique is directly applied to the test image in order to extract the distinct patterns autonomously. The segmentation success rate of these methods depends heavily on the color spectrum of thematic map classes in the test image. Varying success rates are possible; however, initial attempts are found to be encouraging. The performance of hybrid method introduced by (Dogruer 2007) is found to be satisfactory. The Kappa coefficient of the



a) Segmented image by FNN

b) FNN segmentation filtered with median filter

Figure 4.23 Segmented satellite image of Ankara city district

Figure 4.24 Segmented image of Ankara city district with PNN

Figure 4.25 Segmented satellite image of Ankara city district with hybrid method

Table 4.9 Error Matrix for FCM ($\kappa = 0.5791$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.8198	0.0395	0.0287	0.4609
	Road	0.1007	0.7068	0.1032	0.0699
	Forest	0.0475	0.0286	0.6923	0.0569
	Ground	0.0320	0.2252	0.1758	0.4123

Table 4.10 Error Matrix for Hybrid Method ($\kappa = 0.5403$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.6495	0.0342	0.0126	0.1986
	Road	0.2231	0.7910	0.2465	0.2761
	Forest	0.0968	0.1261	0.6993	0.3247
	Ground	0.0306	0.0486	0.0416	0.2006

Table 4.11 Error Matrix for FNN ($\kappa = 0.5942$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.6749	0.0019	0.0120	0.1698
	Road	0.0713	0.7331	0.0451	0.0731
	Forest	0.0731	0.1271	0.8433	0.2310
	Ground	0.1807	0.1380	0.0997	0.5261

Table 4.12 Error Matrix for PNN ($\kappa = 0.5107$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.7104	0.0436	0.0361	0.2075
	Road	0.0626	0.5864	0.0811	0.0822
	Forest	0.0297	0.0656	0.5882	0.0778
	Ground	0.1973	0.3045	0.2946	0.6326

Table 4.13 Error Matrix for FCM ($\kappa = 0.4716$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.7286	0.1510	0.2181	0.0437
	Road	0.1167	0.6188	0.1501	0.1687
	Forest	0.1514	0.1837	0.6143	0.4073
	Ground	0.0033	0.0465	0.0175	0.3803

Table 4.14 Error Matrix for Hybrid Method ($\kappa = 0.5108$)

		Ground Truth Classes			
		Building	Road	Forest	Ground
Thematic Map Classes	Building	0.6674	0.1286	0.0847	0.1916
	Road	0.0533	0.6319	0.0353	0.0512
	Forest	0.1454	0.1150	0.7406	0.2817
	Ground	0.1339	0.1245	0.1394	0.4754

Figure 4.26 Satellite image of Ankara city district

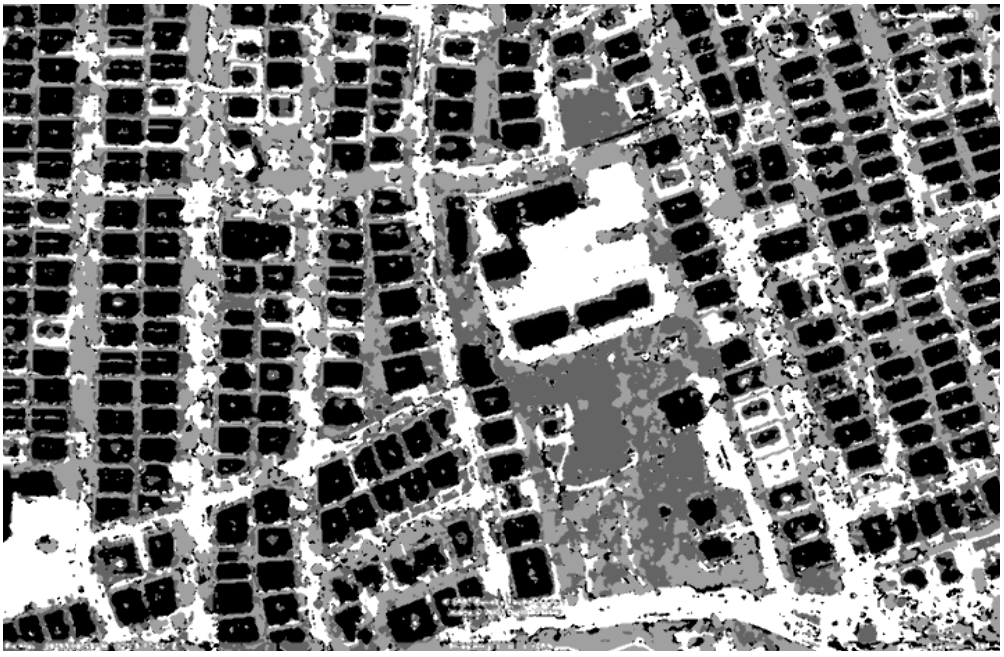


Figure 4.27 Segmented satellite image of Ankara city district with hybrid method

Table 4.15 Approximate segmentation time (seconds)

	Hybrid	FNN	FCM	PNN	MANUAL
Sampling	~300	~300	N/A	~300	N/A
Training	~40	~71	~172	~2	N/A
Segmentation	~34	~13	~35	~118	~1200
Total Time	~374	~384	~207	~420	~1200

method for the first test image is 0.5403 and 0.5108 for the second test image. These figures are in the second place among the Kappa coefficients computed for soft-computing methods. With respect to the computation time, the automatic segmentation demands less time if compared to the manual segmentation and it is more precise when small details are present in the image. However, improved automatic segmentation methods are superior to the manual segmentation.

4.6 Conclusion

In this chapter various soft-computing methods are evaluated for mapping purpose. These are

- FF ANN
- PNN
- FCM
- A novel hybrid classifier

All these methods give satisfactory results under certain conditions. FF ANN and PNN which are supervised techniques require proper training samples. It is observed that selection of the training samples has a big impact on the results. FCM is an unsupervised technique, here the fixing the number of classes is a challenge and the cluster must be attached to the correct information classes by

a human operator. The hybrid classifier which is developed in this study is successful in segmentation when it is compared with other techniques. In the rest of this dissertation, the mapping is done with this novel hybrid classifier.

It is observed there is noise in crafted maps; the influence of this noise term may be fatal. The Kappa coefficient and Error Matrix can evaluate the characteristics of noise left. This must be interpreted with simulations.

CHAPTER 5

A NOVEL APPROACH TO OUTDOOR MOBILE ROBOT LOCALIZATION

5.1 Introduction

So far the objective of dissertation is proposed and the elementary problems whose solution constitutes the solution of global urban localization, **GUL** problem, are considered one by one. However the complete solution has not appeared yet. This chapter introduces the complete solution to global urban localization problem based on the information given on previous chapters. First the mapping problem then the localization problem is studied by considering the specific details of the problem at hand. Finally the solution is proposed. A single simulation result is given to demonstrate the effectiveness, but the detailed experimental results and simulations are postponed to the subsequent chapter.

5.2 Localization

As it is discussed before, Bayes recursive filter is the general framework for tracking problem. However this simple looking filter does not lend itself for closed form solution, in most problems. Analytical solution is possible for few cases, under severe conditions. Kalman filter is the most famous one. Kalman filter assumes that the process and measurement models are linear and process and measurement noises are additive, zero mean, white Gaussians. But mobile robot kinematic model is certainly nonlinear due rotations and related trigonometric functions. EKF appears to be the right option to deal with such nonlinearities. Other means of dealing nonlinear process and measurement models and noises are SMC methods. EKF is a valid tool for tracking but it cannot cope with global localization, so this option is dropped at first sight. SMC methods can deal with multi modal distributions so SIR, auxiliary particle filter and SMC methods can be used for global localization. These

points are discussed in chapter three and remarks are made about these methods. The most popular SMC method, the SIR filter is selected as the principle for simulating Bayes recursive formula and used in the rest of this study.

5.3 Map Generation

In the rest of this work, mapping is done by the hybrid classifier. In previous chapter several soft-computing methods are evaluated for mapping purpose. These soft-computing methods prove to be powerful for mapping. In addition to the standard techniques, a novel hybrid classifier composed of FCM and ANFIS is proposed and compared with them. The performance of this novel classifier keeps up with the performance of standard methods and it manages to generate maps whose noise content is less. So, this novel technique which is developed in this study is selected as the principle tool for mapping.

5.4 Proposed Solution

5.4.1 Introduction

In this chapter a novel solution to GUL problem is presented. The novel solution comprises two disjoint operations:

- Mapping using Remote Sensing Technology
- Localization on a priori map

Mapping with RST is discussed in previous chapter and it is an essential part of the solution, so it will be mentioned briefly here to describe road map of novel solution. The localization with a priori map is studied with Monte Carlo Localization; this method may be classified under Sequential Monte Carlo methods

As mentioned earlier; despite successful applications, the SLAM algorithms are not practical in large-scale environments such as the ones considered in this paper. Note that another important limitation of SLAM is that only the maps of the previously explored regions are available to the robot. Whereas, the maps

obtained from satellite images may enable the robot to devise mission plans on a much larger scale. Similar efforts in literature, which make use of satellite images, can be found in (Booker 2007, Guivant 2007). (Booker 2007) correlated millimeter radar data with aerial images and (Guivant 2007) computed the path of a mobile robot by matching inertial navigation system data to road networks found on satellite images.

Remote sensing promises an alternative solution to outdoor mapping. The remote sensing technology (RST), which is a powerful tool to capture the outlook of large pieces of land across large distances through the use of various spectrum of light, has been successfully applied to a broad range of applications.

Among all of these studies listed for RST above, none of them focused on creating maps that are suitable for outdoor mobile robot navigation. For this purpose, a mixed (a.k.a. “hybrid”) approach that makes good use of the FCM and ANFIS is proposed to identify various objects present in satellite image describing a scene, in which the robot is to operate.

5.4.2 Map Generation Using Satellite Images

In this work, the mapping is considered as a separate step independent of the localization process. Hence, the coupled problem of localization and mapping is reduced to a localization problem over a pre-constructed map. In the proposed technique, satellite image(s) of an operational urban area is processed into a meaningful form that can be used by a robot for navigation purposes. During mapping, geometric information as well as the structural information associated with the segmented parts is included in this representation.

Analysis of satellite image is performed by a combination of unsupervised and supervised techniques; the satellite images are first clustered in feature space by using an unsupervised clustering technique, namely, Fuzzy C-Means (FCM) then boundaries between different fuzzy cluster regions are drawn by an Adaptive Network-Fuzzy Inference System (ANFIS) thus both of them

constructs the structural regions in the environment useful for localization purpose (Richards 2006, Czogala 2000, Yen 2000, Dogruer 2007).

The main structure of segmentation procedure implemented here can be summarized as: A number of satellite images are collected from the region of interest and converted to HSV color spectrum. The images are divided into two groups: Training images, Test images. The pixels of training images are classified by FCM clustering algorithm. Rough FIS is constructed from the clusters found by FCM. The ANFIS is constructed for the FIS and will be trained to compute the parameters of the FIS. The trained system ANFIS/FIS is applied on the test images to label the pixels.

5.4.3 Localization

Localization of a mobile robot, that is the estimation of the pose of robot, $\{x, y, \theta\}$ coordinates of the robot and its heading, respectively, is best explained as a Dynamic Bayesian Network (DBN) where the robot pose/state is the hidden variable. This hidden variable can be represented with a first order Markov Process. A complete description of a Markov process requires specification of three probability measures, namely; state transition model $P[\boldsymbol{x}_{t+1}|\boldsymbol{x}_t]$, observation model $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$, and a priori distribution $P[\boldsymbol{x}_0]$. Given the Markov process, the question to be answered is: Given an observation sequence, $\boldsymbol{Z}_{1:t}$, how a state sequence, $\boldsymbol{X}_{o:t}$ which explains the observations best is found?

Bayesian models can be used to estimate hidden states from given observations. Thus, posterior distributions of states can be obtained from Bayes theorem and inference can be made about the states (i.e. maximum likelihood). The expression for the marginal distribution $p[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t}]$ satisfies the following recursion.

Prediction:

$$P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}] = \int P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t+1}]P[\boldsymbol{x}_{t-1}|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x}_{t-1} \quad (5.1)$$

Updating:

$$P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t}] = \frac{P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]}{\int P[\boldsymbol{z}_t|\boldsymbol{x}_t]P[\boldsymbol{x}_t|\boldsymbol{Z}_{1:t-1}]d\boldsymbol{x}_t} \tag{5.2}$$

Next the discrete time process and measurement models and state transition and observation models of the mobile platform and sensors used in this study are developed considering DBN. The kinematic model of a mobile robot is illustrated in Figure 3.2 where the process and measurement models are given in Equation (3.93) and Equation (3.94), respectively.

Next the computation of $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}]$ and $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$ is explained. Here the motion model $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}]$, which is a First-Order Markov Process, is defined by the process model $f(\boldsymbol{x}_{t-1}, w_{t-1})$ and the known statistics of w_{t-1}(Gordon 1993)

$$P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t+1}] = \int \delta[\boldsymbol{x}_t - f(\boldsymbol{x}_{t-1}, \boldsymbol{w}_{t-1})]\ P[\boldsymbol{w}_{t-1}]d\boldsymbol{w}_{t-1} \tag{5.3}$$

Conditional observation model $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$ are defined by the measurement model $h(\boldsymbol{x}_t, \boldsymbol{v}_t)$ and known statistics of $\boldsymbol{v}_t$. Both process, $\boldsymbol{w}_{t-1}$ and measurement $\boldsymbol{v}_t$, noises are zero mean, temporally uncorrelated white Gaussians (Gordon 1993).

$$P[\boldsymbol{z}_t|\boldsymbol{x}_t] = \int \delta[\boldsymbol{z}_t - h(\boldsymbol{x}_t, \boldsymbol{v}_t)]P[\boldsymbol{v}_t]d\boldsymbol{v}_t \tag{5.4}$$

Both equations require that process and measurement noise are sampled.

The necessary probabilistic models to simulate DBN modeling the mobile robot localization problem are explained next. Since the lost robot case is assumed, there is no prior information about the initial pose of the robot, and the initial distribution is assumed to be uniform.

The motion model is assumed as a Gaussian distribution about the nominal incremental move measured by the odometer. It can be expressed as:

$$P[\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_{t+1}] = \frac{1}{\sqrt{2\pi\sigma_{nom}}} e^{-\frac{[d_{nom}-d]^2}{2\sigma_{nom}{}^2}} \tag{5.5}$$

where $\boldsymbol{u}$ is the action command; $\boldsymbol{x}_t$, is the pose of the robot prior to the execution of the action and $\boldsymbol{x}_{t+1}$ is pose of the robot pose after action command is executed. d_{nom} is the nominal distance traveled by the robot as it is commanded by the action $\boldsymbol{u}$, and σ_{nom} is the standard deviation of uncertainty in the motion. Since the heading is assumed to be known, 1-D process model in polar coordinates is sufficient to develop the motion model of the robot and Equation (5.5) is a direct consequence of Equation (5.3) and Equation (3.93).

The robot scans the environment at constant angular sectors. The measurement model is computed by a joint probability of these single measurements along each angular direction. Thus, the measurement model can be expressed mathematically as:

$$P[\boldsymbol{z}_t|\boldsymbol{x}_t] = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi\sigma_{lms}}} e^{-\frac{[R_k-V_k]^2}{2\sigma_{lms}^2}} \tag{5.6}$$

where K is the total number of angular sectors which sums up to gross angular scan, σ_{lms} is the standard deviation of the uncertainty in the laser scanner measurement, R_k, is the k_{th} measurement in gross measurement $\boldsymbol{z}_t$ observation, made by real laser scanner, V_k is the k_{th} measurement in gross measurement $\boldsymbol{z}_t$ observation, made by virtual laser scanner located on the map whose pose is $\boldsymbol{x}_t$. .Since the scale of environment map obtained from satellite images and the real environment scale does not match, the virtual scans made by virtual laser scanner on the map is calibrated before, such that unit distance measured by the real laser scanner on the robot and the virtual laser scanner on the hypothetical robot on the map, is the same. The joint conditional probability Equation (5.6) can be computed from Equation (5.4) and (3.94). In this study, buildings and dense vegetation are used as landmarks. It has been observed that dense tree groups and man-made objects (e.g. buildings) are

detected by the laser scanner in the field and they can be safely used for localization purpose. Thus, a satellite image is segmented accordingly into four categories: building, forest, ground, and roads. The color codes of four categories are arranged such that the virtual laser scanner is sensitive to the first two categories, building and dense tree groups and is insensitive to last two categories in the map, i.e. ground and asphalt (Dogruer 2007).

Once $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}]$ and $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$ are obtained from the process and measurement models, MCL can be implemented to run the Bayes Filter. Indeed it is not required to evaluate $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}]$ and $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$ exactly, because the MCL is a sample based filter. So it is only required to sample from the measurement and process noises and propagate the particles representing the prior PDF to approximate the PDF in the next time steps. Equation (5.3) and Equation (5.4) can be used for this purpose.

5.4.4 Experimental Setup

To test the performance of the proposed method, a mobile platform is designed to collect data outdoors. The mobile platform is equipped with a SICK LMS 291-S05 laser scanner, a notebook, and a battery pack. Custom software is specifically developed to communicate with this laser scanner and to record the 180-degree polar scan for the scanner. In the experiments, these polar plots are to be recorded at equal distances of travel. Note that the direction of the laser scanner is aligned with the motion of the mobile platform. 180-degree angular sector is scanned with increments of half degrees however the data are then under-sampled with increments of 3 degrees to speed up the simulation process, see Figure 5.1.

During the experiments, the mobile platform is moved along a certain path. Man-made objects (like buildings and walls) allow precise measurements whereas certain elements of nature (vegetation, trees, bushes) may pose difficulty. For instance, the laser beam scattered by the vegetation has considerable noise content in the measurements so the acquired data may not

be directly used for localization. To overcome this difficulty, a custom filter, which compares the derivative (i.e. the first order difference) of the range signal to a selected threshold, is designed, which estimates the new range signal by considering the minima of that region.

5.4.5 Simulation Result

To assess the performance, the (filtered) range data recorded in the field are fed into the simulation software (M-script for Matlab 7.1) as if the robot were actually moving around the track where the data were originally collected.

The standard deviation of process noise is set to 0.4 pixels and in this particular simulation the correlation coefficient is used to compare the similarity of real measurement and virtual measurement. There are 1000 particles.

To speed up the simulation process, the satellite images have been preprocessed to obtain the digital map of the environment. As illustrated in Figure 5.2, the satellite images were segmented into four elements: buildings (in black color), forest (light gray), ground (dark gray), and asphalt road (white) by following the procedure introduced by [Dogruer 2007]. Figure 5.3-5.4 demonstrates the particle distribution (as red marks) on track at various simulation steps. Notice that since the mobile robot is to travel on the roads, the algorithm distributes the particles on this medium only. As can be seen, the simulation results indicate that after the 10^{th} move (step), which corresponds to a 40-meter of travel on the track; the robot localizes itself correctly inside the designated environment. Hence, the localization process thereafter boils down to position tracking.

To assess the performances of the MCL technique on track quantitatively, maximum entropy metric is defined:

$$E = -\sum_{i} p_i log p_i \tag{5.7}$$

where p_i is the value of the PDF at discrete locations. This metric may be conveniently used to compare the performance of multiple approaches or trials with different paths in the same environment or totally different paths and/or environments. Figure 5.4 illustrates the entropies for track at each simulation step. As can be seen from Figure 5.4, the entropy computed for the track drops steadily up to the 10^{th} step and then it oscillates around 0.5. This sudden drop observed at the early phase points out that the algorithm has not settled yet while the steady behavior after this step indicates that the "simulated" mobile robot has been localized and that no further improvements can be accomplished. Note that entropy metrics for the track is consistent with the particle distribution observed in previous figures.

5.5 Conclusion

In the last decade, the use of internet has increased and the digital data being created by various sources have grown dramatically. Not surprisingly, the satellite images of most urban areas have become publicly available on the Internet. Hence, this study has focused on a novel localization method for mobile robots that make good use of such images.

In this chapter, The MCL technique in conjunction with the digital maps has been successfully utilized to localize a mobile robot in outdoor environment. It is shown that a 2D laser scanner and a simple odometer system can be used with digital maps created from satellite images to localize mobile robot in large scale-environment, safely. The mobile robot is localized in a large-scale landscape which is larger than most of the environments used in previous studies.

In fact, the satellite images have several drawbacks. They may be distorted by the camera geometry, atmosphere, and daylight conditions. The orthogonality between top view visible on the satellite images and side view exposed to the laser scanner poses a limitation to implement the method proposed here. Hence, the following conditions, which have significant influences on the results, must be satisfied to apply the method effectively:

- buildings in the images must appear orthogonal and the contour of the top view should be in accordance with the laser scanner plots
- single tree or vegetation, which scatters the laser beam, must be omitted by selecting a suitable field that guarantees to yield a planar world
- satellite images should be (quasi-)static over long time periods and must be further processed such that no objects like moving- or parked cars appear in them.

In the experiments, these conditions are fulfilled as the laser scanner is placed on the top of a vehicle (emulating a mobile robot) whose rooftop is far above the height of an average passenger car on the road. Hence, the non-quasi-static elements in the scene are effectively eliminated. Note that there are some events whose period extends well beyond the capture frequency of satellite images like construction sites. However, the experiments in this study are conducted in a well developed urban setting to avoid such instances. The following section discusses the localization performance of the proposed technique.

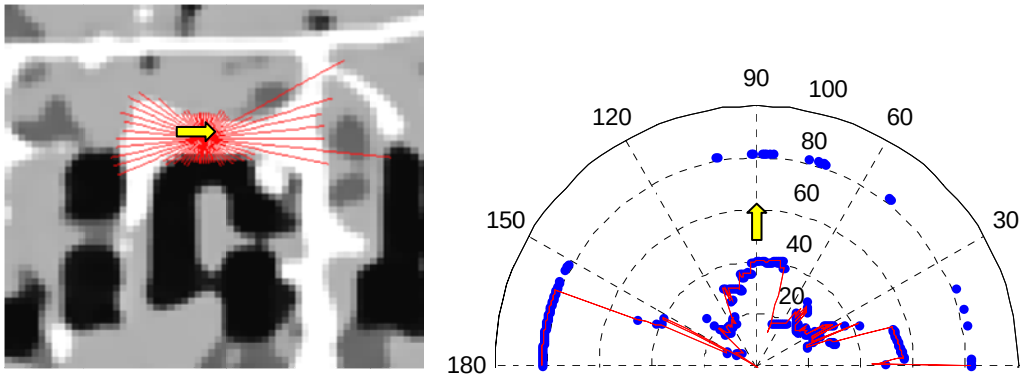


Figure 5.1 Virtual scan made by the virtual laser scanner on hypothetical robot placed on the digital map (Left Figure). Real scan made by the real laser

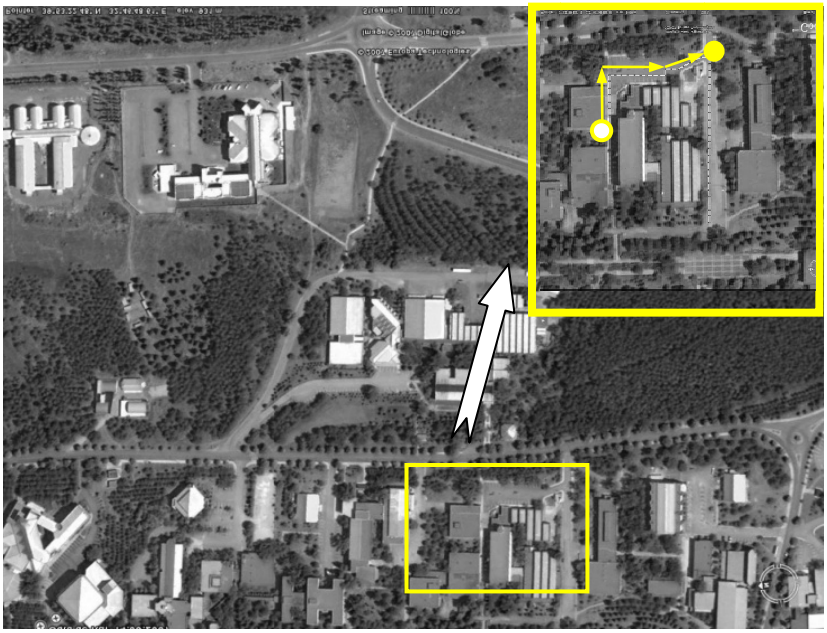


Figure 5.2 Satellite image illustrating the region of the track that is shown by yellow arrows, the start of the track is shown by hollow yellow circle; the full yellow circle shows the end of the track.

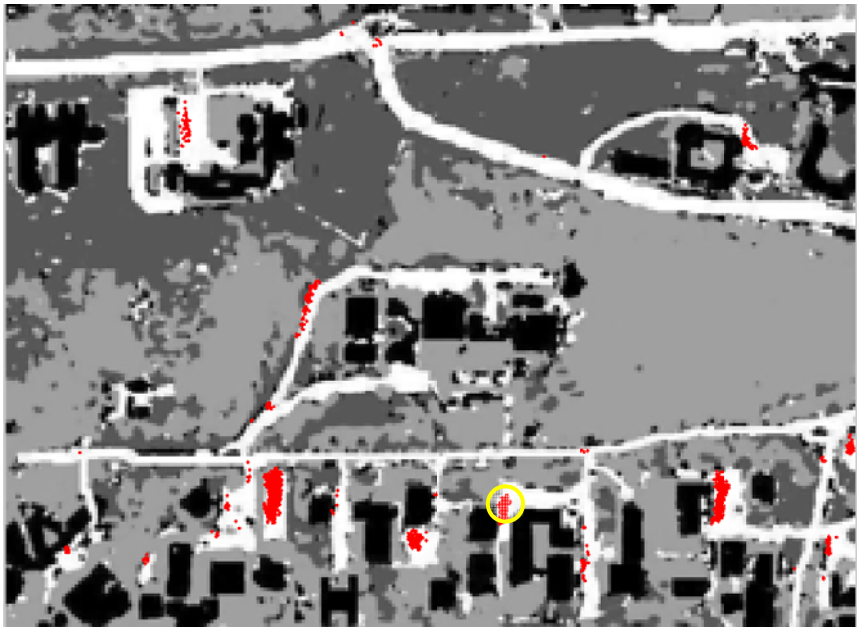
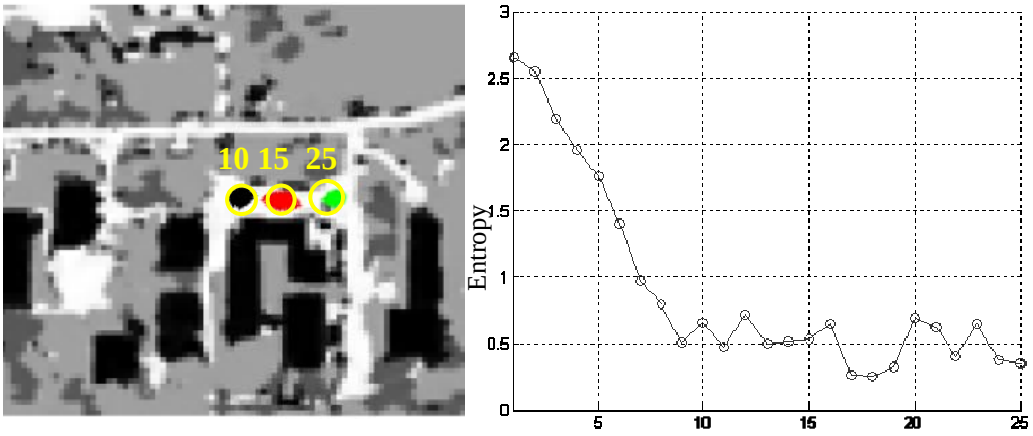


Figure 5.3 Particle distribution is shown by red dots and the real robot pose is shown with yellow circle at simulation step 5.



a) b)

Figure 5.4 a) Particle distribution is by red dots and the real robot pose is shown with yellow circles at simulation steps 10, 15, and 25. b) The entropy metric computed with Equation 5.7

The localization results are found quite satisfactory for all intensive purposes. It is expected that the widespread use of this approach will reduce the setup time of outdoor mobile robotics applications in large-scale urban environments. It will eliminate the map crafting totally. Lastly, the presented approach is the first attempt of solving a novel problem defined in this paper, namely, *Global Urban Localization* (GUL).

CHAPTER 6

NOVEL SOLUTIONS TO GLOBAL URBAN LOCALIZATION

6.1 Introduction

Global Urban Localization problem, which is initially defined in Chapter two, is a mobile robot localization problem solved with the particle filter method. In this chapter, several novel solutions for this problem are proposed and evaluated: Particle Filter, Gaussian Sum Filter, Viterbi Algorithm, Path Matching and Genetic Algorithms.

Global localization has been addressed in mobile robotics literature and the scope of this study sounds very similar to the classic *global localization* problem, but the title and the aim of this study need some extra attention. Here the *Global Urban Localization* is different from *global localization* which is restricted to a given well defined local region. **GUL** aims to localize a mobile robot on planet Earth, not in the local region. Nevertheless, the techniques used to solve global localization are valid tools for **GUL** problem as well. **GUL** is achieved as the robot is localized on the local map and later this local map is registered to telephone communication tower landmark map, because Google Earth can locate a person who uses mobile phone within a circle of one kilometer diameter, without **GPS**. Hence if large enough maps are used, later it is possible to register these maps on the tower landmark maps. Thus a mobile robot localized on the map is said to be localized on Earth.

GUL problem could be considered as an alternative to *Simultaneous Localization And Mapping* (**SLAM**) problem in a large-scale outdoor environment. However, it is apparent that the complexity of **GUL** problem is significantly less than that of **SLAM**. Although there are improvements to the original solution of **SLAM** problem, the computational complexity of methods, data association and dense representation of the environment still remains a

challenge for future studies. Besides solution of **SLAM** requires an exploration strategy which poses other difficulties. Recently, the satellite images of most urban sites are available on the internet and this source, whose information content is rich enough to describe a scene, is thought as an alternative to outdoor mapping. This idea is elaborated in a series of papers (Dogruer et. al 2007, 2008) and successful initial results are obtained. Here other possible solutions are evaluated for the solution of **GUL** problem and new field tests are introduced, these new field tests proves the efficiency and unique side of **GUL** as a *virtual* **GPS**.

The solution of **GUL** may act as a *virtual* **GPS**. However, what is the point in designing a *virtual* **GPS,** as the *real* **GPS** functions well? .**GUL** has a clear advantage over **GPS** technology when it is used in conjunction with a mobile robot. This advantage can be best understood by referring to the philosophy of autonomous navigation. *Global Positioning System* (**GPS**) provides accurate position information, but main disadvantage is that a mobile platform that relies on GPS is dependent on a central unit. Such a mobile platform is not said to be fully autonomous because it is capabilities are not self-contained. Therefore it is required to find other means of localization and internal representation for the environment. With this perspective in mind, the solution of **GUL** problem may provide an answer to the need of autonomous navigation in outdoor settings (Dogruer et al 2007, 2008). **GUL** uses the satellite images for mapping (Dogruer et al 2007) and several other mathematical tools for localizing the mobile robot on pre-constructed maps in previous steps [(Dogruer 2008), thus operation of **GUL** can be summarized in two disjoint consecutive steps; mapping and localization.

It is argued that mathematical tools which are valid for global localization may be safely used to solve **GUL** problem too. The nature of the **GUL** problem and capabilities of mathematical tools are discussed briefly, here. The most apparent side of global localization is that mathematical tools must be capable of tracking multiple hypotheses. Keeping this important piece of information in

mind, the followings may be argued. Particle Filter and Gaussian Sum Filter can approximate complex probability density function (PDFs) which are not necessarily unimodal. Therefore these methods are suitable for multiple hypotheses tracking right at the start of global localization. Since the aim is to localize a mobile robot globally, the methods which are only suitable for position tracking cannot be used e.g. *Kalman Filter* (**KF**) and *Extended Kalman Filter* (**EKF**), because these filters hold a *unimodal* PDF for the state of the system. They cannot cope with *multimodal distributions*, necessary for *multi-hypothesis tracking* at the early stage of localization. However, the Gaussian Sum Filter is made of multiple *Extended Kalman Filters* which start at different positions and run in parallel throughout the analysis so it is suitable for global localization. Particle Filter may approximate complex PDF with a set of random sample and associated weights and it has been used to simulate non-linear, non-Gaussian systems in the past. Thus it may be used to cope with situation where multiple hypotheses arise. *Viterbi algorithm* is a sort of dynamic programming method that computes the best sequence of states which explains a set of observation sequence. Finding the best sequence of states which explains the sequence of observations may be cast as an optimization problem, so Genetic Algorithm and Viterbi algorithm may be used to solve this optimization problem. Lastly, P*ath Matching* is a discreet approach that compares the robot path with road network so that it selects the best road network patch explaining the robot path.

In this chapter, Particle Filter, Gaussian Sum Filter, Viterbi Algorithm, Genetic Algorithm and Path Matching are proposed for the solution of **GUL** problem. New field tests are presented to demonstrate the effectiveness of **GUL** problem as a *virtual* **GPS**. The rest of the chapter is organized as follows: in the next section the theoretical background is laid, Gaussian Sum Filter, Viterbi Algorithm, Genetic Algorithm and Path Matching are explained and finally section two concludes the paper with conclusions. Experimental verification is postponed to next chapter

6.2 Theoretical Background

In this section the methods; Viterbi Algorithm, Particle Filter, Gaussian Sum Filter, Path Registration and Genetic Algorithm will be described and examined. These methods which are described both graphically and literally will constitute the backbone of this chapter. Note that the ultimate aim is to find a state sequence that explains a sequence of observations. It is assumed that the underlying hidden process which emits the observations is a discrete time first order Markov process. So at every discrete time step an observation is emitted by the states but the cause of this observation, the state cannot be observed due to the measurement model or observations which are corrupted by noise do not represent the hidden process correctly. In short the navigation of the mobile is studied and the states/pose of the robot is sought which emits the observations. In this chapter the observations are the relative distance between the mobile robot and landmarks around. In its naïve form these relative distances do not reveal the hidden states directly e.g. the robot pose but the states may be inferred by using these relative distances. For this purpose an observation model and a state transition model are needed. They explain the probability of emission of a certain observation from a particular state and the probability of move from a certain state to a target state, respectively. The Dynamic Bayesian Network which represents a navigating mobile robot is shown in Figure 2.2. In this Figure, the pose of the robot is hidden state and may only be sensed through the observations attached to a particular state, e.g. $\boldsymbol{z}_{t-1}$ attached to $\boldsymbol{x}_{t-1}$. A particular case of Dynamic Bayesian Network (DBN) is the Hidden Markov Model (HMM). HMM is a discrete time process where the discrete states and observations may be selected from finite set. The hidden states evolves in time according to first order Markov process model and the next state only depends on the previous state and do not depend on the whole state history. Therefore the history of states may be forgotten safely without affecting the analysis. The current observations depend only on the current state. These two assumptions simplify the computations in great deal. DBN shown in Figure 2.2 explains these assumptions graphically.

6.2.1 Constrained Viterbi Algorithm

First, the Viterbi algorithm (Forney 1973) is employed to reveal the hidden state sequences, e.g. robot pose explaining the sequence of observation (the relative distance between the robot and landmarks around) best. In fact, the Viterbi algorithm is a kind of dynamic programming method which finds the most likely state sequence. It is assumed that the cost of move from current state to the next one can be represented by a number increasing in time.

The probability of state sequence which describes the sequence of observations best may be cast as maximum a posterior (MAP), given by $P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})$. Using Bayes' rule (Doucet 2001, Moreno 2006, 2007), one can obtain

$$\mathrm{P}(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) = \frac{P(\boldsymbol{Z}_{1:t}|\boldsymbol{X}_{0:t})P(\boldsymbol{X}_{0:t})}{P(\boldsymbol{Z}_{1:t})} \tag{6.1}$$

Since $\mathrm{P}(\mathbf{Z}_{1:\mathrm{t}})$ is just a constant scale factor which ensures a proper PDF, it can safely be dropped out:

$$P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \propto P(\boldsymbol{Z}_{1:t}|\boldsymbol{X}_{0:t})P(\boldsymbol{X}_{0:t}) \tag{6.2}$$

With the knowledge of finite set of observations at hand and conditional independence among these observations stating that each observation originates from a single state; $\mathrm{P}(\mathbf{X}_{0:\mathrm{t}}|\mathbf{Z}_{1:\mathrm{t}})$ can be expressed as

$$P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \propto P(\boldsymbol{z}_1|\boldsymbol{x}_1)P(\boldsymbol{z}_2|\boldsymbol{x}_2) \ldots P(\boldsymbol{z}_n|\boldsymbol{x}_n)P(\boldsymbol{X}_{0:t}) \tag{6.3}$$

This expression may be further simplified with first order Markov process model:

$$\begin{aligned} &P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \\ &\propto P(\boldsymbol{z}_1|\boldsymbol{x}_1)P(\boldsymbol{z}_2|\boldsymbol{x}_2)..P(\boldsymbol{z}_n|\boldsymbol{x}_n)P(\boldsymbol{x}_2|\boldsymbol{x}_1)P(\boldsymbol{x}_3|\boldsymbol{x}_2)..P(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) \end{aligned} \tag{6.4}$$

Taking the logarithm of this expression yields

$$log[P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})] = \sum_{i=1}^{n} log(\boldsymbol{z}_i|\boldsymbol{x}_i) + log[P(\boldsymbol{x}_1|\boldsymbol{x}_0) + \cdots + (\boldsymbol{x}_n|\boldsymbol{x}_{n-1})] \tag{6.5}$$

The MAP problem may be then considered as the determination of the shortest path on graph. It is critical to note that the path representing the MAP is recursive in nature. Hence, the minimization of the term

$$-log[P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})] = -log[P(\boldsymbol{Z}_{1:t}|\boldsymbol{X}_{0:t}) - P(\boldsymbol{X}_{0:t})] \tag{6.6}$$

is equivalent to maximization of the argument for the rightmost term in Equation (6.6).

In its naïve form the problem may expressed as:

$$P(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \propto P(\boldsymbol{Z}_{1:t}|\boldsymbol{X}_{0:t})P(\boldsymbol{X}_{0:t}) \tag{6.7}$$

With

$$\delta_t(i) = \underset{\boldsymbol{x}_1\boldsymbol{x}_2 \ldots \boldsymbol{x}_{t-1}}{argmax} P[\boldsymbol{x}_1\boldsymbol{x}_2..\boldsymbol{x}_{t-1}, \boldsymbol{x}_t = i|\boldsymbol{z}_1\boldsymbol{z}_2..\boldsymbol{z}_t, \lambda] \tag{6.8}$$

where $\delta_t(i)$ is the best score (highest probability) along a single path at time *t* which account for the first *t* observations and ends in state S_i, by induction:

$$\delta_{t+1}(j) = \left[\underset{i}{max}\, \delta_t(i)\left[P\left(\boldsymbol{x}_j|\boldsymbol{x}_i\right)\right]\right] P\left(\boldsymbol{z}_{t+1}|\boldsymbol{x}_j\right) \tag{6.9}$$

This is equivalent to

$$L_{t+1}(j) = \left[\underset{i}{min}\, L_t(i) + l_t(i,j)\right] \tag{6.10}$$

where

$$l_t(i,j) = -log\left[P\left(\boldsymbol{x}_j|\boldsymbol{x}_i\right)\right] - log\left[P\left(\boldsymbol{z}_{t+1}|\boldsymbol{x}_j\right)\right] \tag{6.11}$$

Hence, the algorithm may be summarized in Table 6.1 (Forney 1973). The pseudo code is given in Table 6.2.

Note that motion model of the mobile robot limits the number states that can be reached from a particular state; this is illustrated in Figure 6.1. In this Figure, motion model imposes constraints on the next states; the robot is represented by a Δ. Note that there is an elliptical region represented by symbol O, described by the covariance of states. Lines represent the relative displacement vector. The 2D Gaussian PDF is computed with the process model. Outside this region the likelihood of the robot being there is negligible so they may safely be ignored. This modified version of original Viterbi algorithm is much faster. Since the likely states are selected from a regular grid as elliptic region, the cost of this operation may be safely ignored. Note that the motion model is reversible, that is to say $P[i|j,-a] = P[j|i,a]$. Thus the cost of one step of recursion in Viterbi algorithm is MN addition and selection among MN paths leading to M states, where M is the total number of grid points and N is the grid points selected by using the motion model. Note that $M \gg N$, and this will reduce the overall cost of the algorithm in great deal.

As it is shown in Figure 6.1 with a black region, sometimes multiple path may lead to a single state, then only the survivor of those links must be stored, e.g. the link with minimum cost. Figure 6.1, illustrate the same problem on 2D matrix representation of trellis structure, it is clear that not all states are connected to each other as the motion model imposes constraints on the next states. Figure 6.2 illustrates the same problem in 1D ordered trellis, the cost move from one state is shown on the link connecting both states and the survivor link is shown as bold and the cost of being at a particular state is shown in the circle representing the states. The optimal path is shown with red bold lines. Since logarithmic function is monotonic, the search in the next time step can be bounded with a certain number of frontiers as the survivor. This will further reduce the cost of search for optimal path.

Table 6.1 Viterbi Algorithm

Initialization

$L_1(j) = 0$

$\psi_1(i) = 0$

$1 \leq i \ll N$

Recursion

$L_{t+1}(j) = \underset{i \in I}{min}[L_t(i) + l_t(i,j)]$

$\psi_t(j) = \left[\underset{j}{argmin} L_{t+1}(j)\right]$

$I = (i; P[i|j, -a] \geq T_m) \qquad P[i|j, -a] = P[j|i, a]$

Termination

$P^* = \underset{1 \leq i \leq N}{min}[L_T(i)]$

$q_T^* = \underset{1 \leq i \leq N}{argmin}[L_T(i)]$

Path (state sequence) backtracking

$q_t^* = \psi_{t+1}(q_{t+1}^*)$

$t = T-1, T-2, \ldots, 1$

In order to select the next set of reachable states first, the current set of states which have survived is projected forward in time. Thus a set of child states whose likelihood is high is determined. Then, these child states are projected backward in time to determine the set of parent states with high likelihood. Finally the search is started from these parent states so as to minimize the cost of move which is defined by Equation (6.11).

$$J = (j; P[j|i \in I_{surv}, a] \geq T_m) \tag{6.12}$$

$$I = (i; P[i|j \in J, -a] \geq T_m) \cap I_{surv} \tag{6.13}$$

$$I_{surv} = select\ p\%\ of\ next\ states\ J\ as\ survivors$$

Table 6.2 Pseudo code for constrained Viterbi algorithm

1. *N search from N states is started*
2. *For each state*
    a. *The cost of moving from a state that is being considered to next state is computed by*
    $l_t(i,j) = -log\left[P\left(\boldsymbol{x}_j|\boldsymbol{x}_i\right)\right] - log\left[P\left(\boldsymbol{z}_{t+1}|\boldsymbol{x}_j\right)\right]$
    b. *Each next state may be reached from N state in previous time step. For each state the cost of being there is computed as the cost of previous state plus the cost of move to the next state. Then optimal path with minimum cost is selected i.e. the most likely path as it will maximize motion model* $P\left(\boldsymbol{x}_j|\boldsymbol{x}_i,\boldsymbol{u}\right)$ *and observation model* $P\left(\boldsymbol{z}_{t+1}|\boldsymbol{x}_j\right)$*considering the cost of starting state*
3. *This recursion is repeated for each state*
4. *The same recursion is repeated for the next time step*
5. *When the simulation ends*
    a. *The state with minimum cost is selected and marked as the optimal final pose of the mobile robot*
    b. *Starting from the final state, the optimal links are traced back to the start of simulation.*
    c. *Hence the whole path of the mobile robot is revealed.*

6.2.2 Monte Carlo Localization

Monte Carlo Localization can be derived from Bayes Filters which address the problem of estimating the posterior distribution of state conditioned on all data (Gordon 1993, Thrun 2001, Doucet 2001). In a static environment; only state of robot, x_t, changes in time and the posterior distribution of robot's state can be expressed as

$$Bel(\boldsymbol{x}_t) = P(\boldsymbol{x}_t|\boldsymbol{d}_{0:t}) \tag{6.14}$$

$$\boldsymbol{d}_{0:t} = \{\boldsymbol{z}_t, \boldsymbol{u}_{t-1}, \boldsymbol{z_{t-1}}, \boldsymbol{u}_{t-2}, \ldots, \boldsymbol{z}_0, \boldsymbol{u}_0\} \tag{6.15}$$

Bayes Filters estimates the posterior distribution recursively and can be expressed as

$$Bel(\boldsymbol{x}_t) = \eta P(\boldsymbol{z}_t|\boldsymbol{x}_t) \int P(\boldsymbol{x}_t|\boldsymbol{a}_{t-1}, \boldsymbol{x}_{t-1}) Bel(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \int P(\boldsymbol{x}_t|\boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1}) Bel(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1} \tag{6.16}$$

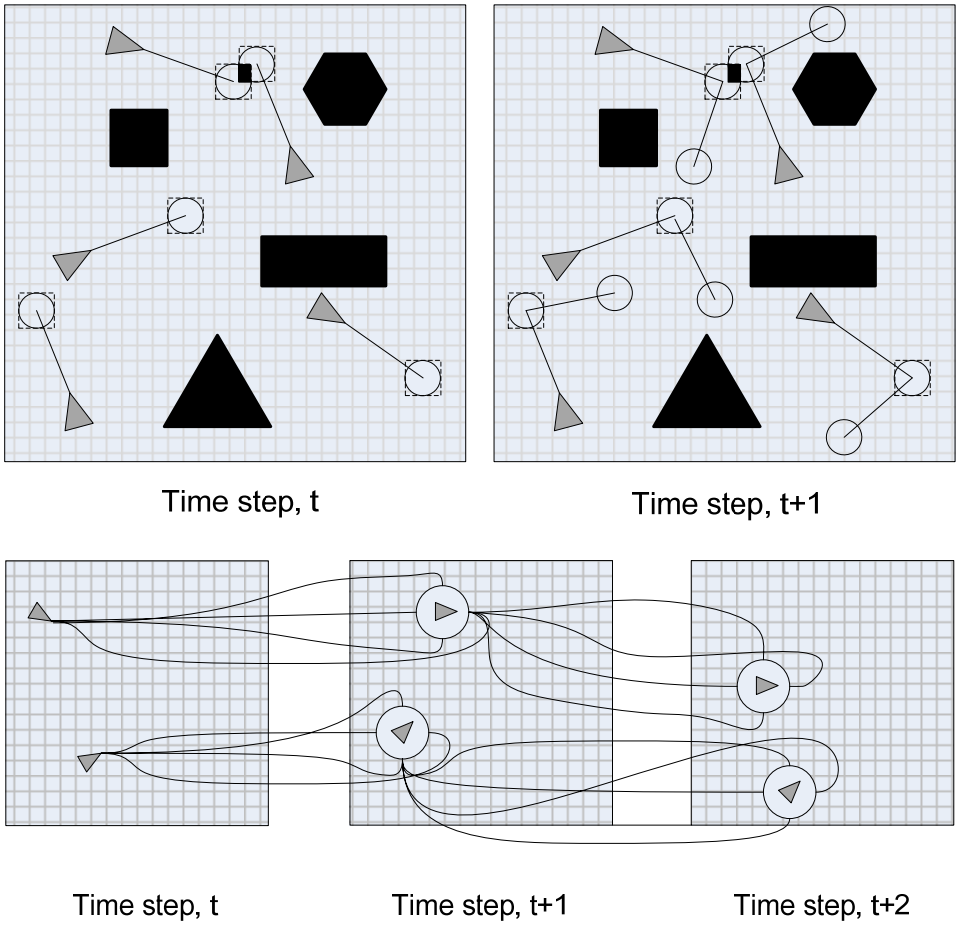


Figure 6.1 First figure shows constrained motion of a mobile robot. Second figure shows the constrained motion on a 2D trellis

where η is a normalizing factor.

While implementing the MCL, the belief $Bel(\boldsymbol{x}_t)$ is represented by a set of weighted particles $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$. In this set, $\boldsymbol{x}_t^i$ are particles and w_t^i are the weights of particles which are determined according to $Bel(\boldsymbol{x}_t)$. The complete steps of MCL algorithm pseudo code are given in Table 6.3 and the algorithm is depicted in Table 6.4. First, a particle $\boldsymbol{x}_{t-1}^{i*}$ is sampled from $Bel(\boldsymbol{x}_{t-1})$ which is represented with discrete set of particles and weights $\{\boldsymbol{x}_{t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}$. Then, $\boldsymbol{x}_t^i$ is sampled from $\boldsymbol{x}_t^i$motion models, $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{i*}, \boldsymbol{a}_{t-1})(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{i*}, \boldsymbol{u}_{t-1})$ by using $\{\boldsymbol{x}_{t-1}^{i*}, \boldsymbol{a}_{t-1}\}\{\boldsymbol{x}_{t-1}^{i*}, \boldsymbol{u}_{t-1}\}$. Thus prior PDF of $\boldsymbol{x}_t^i$ is equivalent to $P(\boldsymbol{x}_t|\boldsymbol{a}_{t-1}, \boldsymbol{x}_{t-1})(\boldsymbol{x}_t|\boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1})Bel(\boldsymbol{x}_{t-1})$. Finally, the particles are weighted

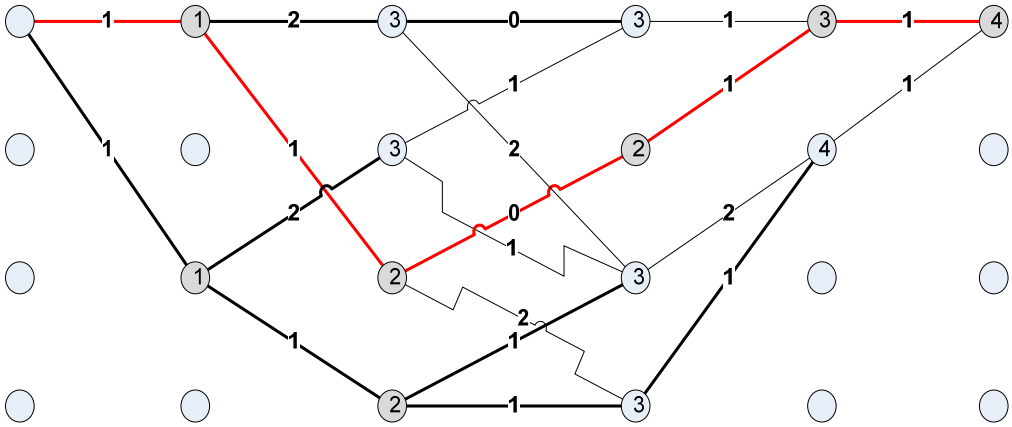


Figure 6.2 1D ordered state trellis. The optimal path is shown with red color line.

by non-normalized $P(\boldsymbol{z}_t|\boldsymbol{x}_t)$. When N_s particles are sampled, their weights are normalized so that sum of weights is equal to one. Thus the posterior PDF of $\boldsymbol{x}_t^i$ is represented with $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$. The degeneracy of particles may be avoided by using a Resampling step.

6.2.3 Gaussian Sum Filter

A single Extended Kalman Filter (EKF) may hold an approximate a Gaussian PDF of states of a nonlinear system. However, it cannot represent a multimodal PDF. In order to cope with this situation, EKFs linearized at multiple local states could be utilized. Thus, multiple EKFs armed with the process- and weighted measurement models run in parallel throughout the analysis that is projected forward in time. Such an approach may be used to deal with the multimodal nature of the global localization. First, the initial PDF characterizing the knowledge about the system can be represented by multiple Gaussian PDFs. This can be expressed by

$$p(\boldsymbol{x}_0|\boldsymbol{Z}^{-1}) = \sum_{i=1}^{\xi_0'} \alpha_{0i}' \, N(\boldsymbol{x}_0 : \boldsymbol{x}_{0i}', \boldsymbol{P}_{0i}') \tag{6.17}$$

where α_{0i}' are the weights and $N(\boldsymbol{x}_0 : \boldsymbol{x}_{0i}', \boldsymbol{P}_{0i}')$ is the i^{th} Gaussian PDF with mean $\boldsymbol{x}_{0i}'$ and covariance $\boldsymbol{P}_{0i}'$ evaluated at $\boldsymbol{x}_0$ and number of Gaussian PDF's are ξ_0'. Non-negative weights,α_{0i}' are normalized such that their sum is equal to one. This is expressed with Equation (6.19).

$$\sum_{i=1}^{\xi_0'} \alpha_{0i}' = 1 \tag{6.18}$$

$$\alpha_{0i}' > 0 \tag{6.19}$$

Each EKF are projected in time through the process model. Note that each EKF are non-interacting and separate predictive and posterior PDF are computed for each of them. Then each separate EKF is weighted by the nonnormalized likelihood of measurements. Finally the weights are normalized such that their sum is equal to one. The complete steps of Gaussian Sum Filter are given below. The EKF prediction and update equations are not shown so as to keep the formulation brief.

Table 6.3 Pseudo code for Monte Carlo localization

1. *Determine the local region by using Google Earth software running on a mobile phone. This step may also be called crude localization*
2. *Download the satellite images of the local region from internet by using Google Earth software*
3. *Convert the satellite images into maps using ANN methods proposed in chapter. Note that in this study, Hybrid structure is employed for this purpose*
4. *Spread N particles on the map. At this stage the structure of the environment is taken into account so particles are only placed on roads and ground where a mobile robot may operate*
5. *Run particle filter*
    a. *The robot scans the environment and records polar plots*
    b. *For every particle representing the robot pose, generate hypothetical polar scans using the map and assuming that robot is the particle being considered*
    c. *Compute conditional probability of observation model* $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$
    d. *Assign each particle a weight equal to* $P[\boldsymbol{z}_t|\boldsymbol{x}_t]$
    e. *Normalize the weights such that their sum is equal to one*
    f. *Predict the states at next time step by using the current state which is assumed to be equal to the particle being considered and the motion model* $P[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t]$.
        i. *First select a particle and set* $\boldsymbol{x}_{t-1}$ *equal to that particle*
        ii. *Using* $\boldsymbol{x}_{t-1}$ *and the action command* $\boldsymbol{u}_t$, *sample the motion model*
    g. *Apply Resampling to eliminate particles whose weights are insignificant*
    h. *Go to step (a)*

Table 6.4 Monte Carlo Localization Algorithm

$$\left[\left\{\boldsymbol{x}_t^i, w_t^i\right\}_{i=1}^{N_s}\right] = MCL\left[\left[\left\{\boldsymbol{x}_{t-1}^i, w_{t-1}^i\right\}_{i=1}^{N_s}, \boldsymbol{z}_t\right]\right]$$

$FOR\ i = 1{:}\ N_s$

$Draw\ \boldsymbol{x}_t^i \sim p\left(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i\right)$

$Assign\ the\ particle\ a\ weight\ w_t^i = p\left(\boldsymbol{z}_t | \boldsymbol{x}_t^i\right)$

$END\ FOR$

$Calculate\ the\ total\ weight{:}\ t = SUM\left[\left\{w_t^i\right\}_{i=1}^{N_s}\right]$

$FOR\ i = 1{:}\ N_s$

$Normalize{:}\ w_t^i = w_t^i t^{-1}$

$END\ FOR$

$$Calculate\ \widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s}\left(w_t^i\right)^2}$$

$IF\ \widehat{N}_{eff} < N_T$

$Resample$

$$\left[\left\{\boldsymbol{x}_t^i, w_t^i, -\right\}_{i=1}^{N_s}\right] = RESAMPLE\left[\left[\left\{\boldsymbol{x}_t^i, w_t^i\right\}_{i=1}^{N_s}, \boldsymbol{z}_t\right]\right]$$

$END\ IF$

The approximate predictive PDF:

$$p_A(\boldsymbol{x}_{t+1}|\boldsymbol{Z}_{0:t}) = \sum_{i=1}^{\xi'_{k+1}} \alpha'_{t+1,i}\, N\left(\boldsymbol{x}_{t+1}{:}\, \boldsymbol{x}'_{t+1,i}, \boldsymbol{P}'_{t+1,i}\right) \tag{6.20}$$

The approximate filtering PDF:

$$p_k(\boldsymbol{x}_k|\boldsymbol{Z}_{0:t}) = \sum_{i=1}^{\xi_k} \alpha_{k,i}\, N\left(\boldsymbol{x}_t{:}\, \widehat{\boldsymbol{x}}_{t,i}, \boldsymbol{P}_{t,i}\right) \tag{6.21}$$

$$\zeta_{t,i} = N\left(\boldsymbol{z}_t{:}\, \boldsymbol{h}_t\left(\boldsymbol{x}'_{t,i}\right), \boldsymbol{S}_{\boldsymbol{t},\boldsymbol{i}}\right) \tag{6.22}$$

$$\alpha_{t,i} = \frac{\alpha'_{t,i}\zeta_{t,i}}{\sum_{i=1}^{\xi'_t}\alpha'_{t,i}\zeta_{t,i}} \tag{6.23}$$

The pseudo code for Gaussian sum filter is given in Table 6.5.

6.2.4 Path Matching

Here, the particle filter explained in previous pages is implemented with different strategy (Gordon 1993, Thrun 2001, Doucet 2001, Guivant 2007). First the road network is extracted from the satellite images and particles representing the robot states are spread on the road patches. Then particle filter is implemented with the following observation model. In Figure 6.3, the

Table 6.5 Pseudo code for Gaussian sum filter

1. *Filtering density is approximated by a number of Gaussian PDF*
2. *Each Gaussian is tracked by a single EKF*
3. *Each hypothesis is tracked by a single EKF*
4. *Each EKF are non-interacting*
5. *They operate separately*
    a. *Prediction stage*
    b. *Update stage*
    c. *Weight of each Gaussian is computed by*
       $\zeta_{t,i} = N\left(\boldsymbol{z}_t : \boldsymbol{h}_t\left(\boldsymbol{x}'_{t,i}\right), \boldsymbol{S_{t,i}}\right)$
    d. *Weights are normalized so their sum is one*
    e. *Go to step (a)*

displacement is represented with black rectangle and the part of the road is shown with blue rectangle. Every time the robot moves, particles representing the robot is projected forward in time and weighted by intersection of a rectangle, whose orientation and center is the same as the robot states, with the

road network. Note that this intersection area will favor particles which are aligned with the road network.

Mathematically the observation model can be expressed as:

$$P(\boldsymbol{o}_t|\boldsymbol{x}_t) = A \cap Road\ Network \tag{6.24}$$

Where A is a rectangular patch located around the particle representing path traversed so far. In this way, the inaccurate path computed by the odometer is projected onto the road network and as the robot moves around the most successful candidates will survive. This is illustrated in Figure 6.3. The road network is shown with black thin lines and the robot path computed with odometer is shown with blue line and the most likely path computed with particle filter is shown with red lines. Indeed this is nothing but the most likely projection of the path, which is

Table 6.6 Pseudo code for path matching

1. *Path matching is an implementation of particle filter*
2. *However its observation function is modified*
    a. $P(\boldsymbol{o}_t|\boldsymbol{x}_t) = A \cap Road\ Network$
    b. *This observation function computes the intersection of past trajectory with the road network*
    c. *Robot pose is represented with particle spread over road network*
    d. *Since road networks are abstract representation extracted from satellite image. The motion model is modified so as to allow a limited angular search in state space*
    e. *This angular search allows particles and past trajectory to align themselves with the abstract road network representation*

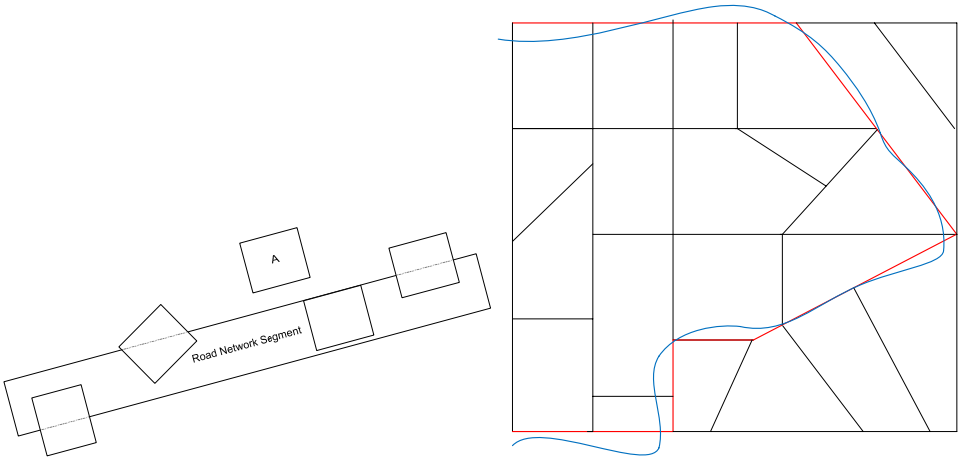


Figure 6.3 Path matching: The road network (black lines), imprecise path (blue lines) and the optimal path (red line)

computed with odometer, onto the road network. The pseudo code for path matching is given in Table 6.6.

6.2.5 A Novel Solution Based on Differential Evolution

In order to solve the global localization, the evolutionary filter developed in (Moreno 2006, 2007) is adopted with a major modification. This modification introduces the island model that is described in terms of the covariance matrix of the posterior of robot pose/states. Hence, the main advantage of the genetic algorithm in this problem (global optimization) is replaced with multiple local minima. This idea is inspired from the simulated annealing and the properties of Kalman Filter. The island model incorporated into genetic algorithm is illustrated in Figure 6.4. The uncertainty of the robot state increases over time, because of the inaccurate process model and observation model; this is illustrated in Figure 6.4. In the absence of absolute measurements, this uncertainty is unbounded. This uncertainty ellipse may describe the local region in which the local parameter vectors may interact. Outside this uncertainty ellipse the likelihood of interaction between parameter vectors is very low. Hence, the continuous parameter space may be divided into elliptical

region around the parameter vector that is considered at the time. Indeed several local searches are initiated at multiple points, overlapped elliptical regions in sequence is possible which leads to mixture between different island regions. Hence it maintains the global optimization tool embedded in a series of island chain, thus an implicit interaction is possible between distant parameter vectors. Size of island region grows in time because the uncertainty ellipse and covariance of the system state grows as the robot navigates around. Thus, all parameter vectors will interact eventually and the global optimization is achieved.

In the next paragraphs, the evolutionary filter will be derived and explained with island model. This derivation is a natural extension of the approach proposed in (Moreno 2006, 2007). The evolution filter utilizes 3-dimensional parameter vectors, for instance $\boldsymbol{x}_i^k$ is the i^{th} solution candidate/vector at k^{th} simulation step:

$$\boldsymbol{x}_i^k = \left[x_i^k, y_i^k, \theta_i^k\right]^T \tag{6.25}$$

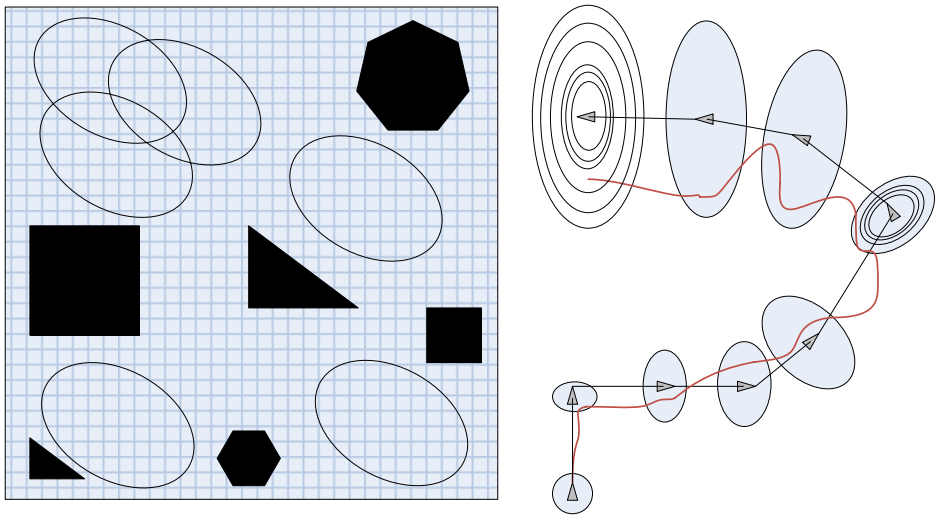


Figure 6.4 Covariance of states grows as the robot moves around (right figure). This covariance is used to describe the dynamic islands in which the states may interact (left figure).

N parameter vectors $\{\boldsymbol{x}_i^k, i = 0,1,..,N\}$ constitutes the generation of the optimization process. The initial population is chosen as randomly to cover entire state space. The differential evolution filter generates a new parameter vector by summing the weighted difference of the best parameter vector and the parameter vector being considered to itself, the result of this reproduction is further changed by the weighted difference of two randomly chosen member of the population. If the performance of resulting vector is better than a selected population member, the generated offspring replaces the vector which it is compared; otherwise the old member is kept.

The generated offspring is disturbed according to the following scheme.

$$v = \boldsymbol{x}_i^k + W(\boldsymbol{x}_b^k - \boldsymbol{x}_i^k) + F(\boldsymbol{x}_{r3}^k - \boldsymbol{x}_{r2}^k) \tag{6.26}$$

where $\boldsymbol{x}_i^k$ is the member of population to be perturbed at iteration *k;* $\boldsymbol{x}_b^k$ is the best member in the population at iteration k, and $\boldsymbol{x}_{r2}^k$ and $\boldsymbol{x}_{r3}^k$ refer to parameter vectors chosen randomly from the population and are different for every *i* index.

Crossover is used to change new generation of parameter vectors further. Denote by

$$\boldsymbol{u}_i^k = [x_i^k, y_i^k, \theta_i^k]^T \tag{6.27}$$

The new parameter vector with

$$u_{i,j}^k = \begin{cases} v_{i,j}^k & if\ p_{i,j}^k < \delta \\ x_{i,j}^k & otherwise \end{cases} \tag{6.28}$$

Where $p_{i,j}^k$ is a randomly chosen variable from the interval [0,1], for each parameter *j* of the population member *i* at step *k* and δ is the crossover probability. The decision about the new vector $\boldsymbol{u}_i^k$ is made: $\boldsymbol{u}_i^k$ is compared with $\boldsymbol{x}_i^k$. If the vector $\boldsymbol{u}_i^k$ performs better than $\boldsymbol{x}_i^k$, then it is replaced with $\boldsymbol{u}_i^k$.

Table 6.7 Pseudo code for novel method based on differential evolution

1. *N chromosomes encoding the pose of robot are spread over the state space of mobile robot*
2. *New offspring are generated*
3. *The covariance matrix of robot state is used create dynamic island*
    a. *An ellipse is drawn by using covariance matrix*
    b. *A single chromosome is selected from population*
    c. *Ellipse is centered at the chromosome being considered*
    d. *The chromosomes which are member of the elliptical region are selected*
    e. *Differential evolution runs with this subpopulation e.g. chromosomes which are member of the temporary ellipse*
    f. *Repeat this loop N times; Go to step (b)*
    g. Repeat (b)-(f) k times
4. *Predict the next generation*
    a. Draw an ellipse around each chromosome
    b. Choose best chromosome
    c. Use this chromosome and motion model to predict the member of next population
    d. Keep a log of parent and child chromosome
    e. Repeat this for every chromosome in the population; Go to (a)
5. *Computation of fitness*
    a. Compute fitness
    b. Use real measurements
    c. Use virtual measurements made by offspring
    d. Use new offspring as state
    e. Use predicted state for that chromosome in (4). Note that a log of parent chromosome and predicted state for that chromosome is kept in (4-d)

$\boldsymbol{x}_{i,t}^{k}$ which is defined as a possible robot pose solution *i* obtained in the last iteration of the algorithm at step *t*, is used with the robot motion model and odometer information, to compute the estimated position at step *t+1*. This is applied to all population members. Next, the cost function which is used to compute the fitness of population member is defined in terms of measurement and motion innovation. The cost function is defined as

$$\begin{aligned} L\left(\hat{\boldsymbol{x}}_{i,t+1} - \boldsymbol{x}_{i,t+1}^{k}\right) \\ = \left[\boldsymbol{z}_{i,t+1} - \hat{\boldsymbol{z}}_{i,t+1}\right]^{T} Q^{-1} \left[\boldsymbol{z}_{i,t+1} - \hat{\boldsymbol{z}}_{i,t+1}\right] \\ + k_b \left[\hat{\boldsymbol{x}}_{i,t+1} - \boldsymbol{x}_{i,t+1}^{k}\right]^{T} R^{-1} \left[\hat{\boldsymbol{x}}_{i,t+1} - \boldsymbol{x}_{i,t+1}^{k}\right] \end{aligned} \tag{6.29}$$

where Q and R are the measurement and process noise covariance matrices. The first term is the measurement innovation and the second term is motion innovation. Special attention must be paid how to compute the predicted state of the system, $\hat{\boldsymbol{x}}_{i,t+1}$ It is described as

$$J = \left\{j : j \; is \; a \; member \; of \; ellipse \; centered \; at \; \boldsymbol{x}_{i,t}^{k}\right\} \tag{6.30}$$

$$\hat{\boldsymbol{x}}_{i,t} = \underset{j \in J}{argmin} \, L\left(\hat{\boldsymbol{x}}_{i,t+1} - \boldsymbol{x}_{i,t+1}^{k}\right) \tag{6.31}$$

$$\hat{\boldsymbol{x}}_{i,t+1} = f\left(\hat{\boldsymbol{x}}_{i,t}, u_t\right) \tag{6.32}$$

First a solution candidate is selected and an elliptical region which is centered at that solution candidate is drawn, this is expressed in Equation (6.30). Then the best solution in this elliptical region is selected, see Equation (6.31) and projected forward in time using the process model, see Equation (6.32). Thus $\hat{\boldsymbol{x}}_{i,t+1}$ is the predicted mean for $\boldsymbol{x}_{i,t+1}^{k}$whose ancestor is $\boldsymbol{x}_{i,t}^{k}$. The pseudo code for novel solution based on differential evolution is given in Table 6.7.

6.3 Conclusion and Discussion

In this dissertation, the ***G**lobal **U**rban **L**ocalization* problem is investigated thoroughly. Despite the fact that the GPS technology is the prevalent means of

providing position data, it has some shortcomings. That is, the GPS data may be distorted due to multi-path effect and thus the accuracy of the position data depends on the number of observed satellites, so the nearby objects, like building may affect the position data adversely. Furthermore, it may not provide position data under certain conditions such as traveling inside tunnels, underwater applications or robots operating in mining halls. Since the GPS technology may not provide accurate position data always, an alternative approach must be sought.

In this chapter, a map matching approach is proposed but the unique side of the study is that it uses Internet web sites (i.e. Google Earth) to download the satellite images about the local region. The same web site, Google Earth, can localize a person who uses a mobile phone in a circular region. Hence, the information can be put together to localize a mobile platform anywhere on Earth. Although the localization methods which are valid for global localization can be used for the novel problem defined in this study, note that ***G****lobal* ***U****rban* ***L****ocalization* problem is different from *global localization*. This point is elaborated in the introduction part of this chapter.

In order to solve this problem, several methods including Viterbi Algorithm, Particle Filter, Gaussian Sum Filter, Path Registration and Genetic Algorithm are evaluated. Particle Filter has been investigated within the context of position tracking and global localization problems but here it is evaluated for a novel problem, namely ***G****lobal* ***U****rban* ***L****ocalization* problem. Path matching is evaluated as an alternative with minor modification to the observation model. Hardly any papers are found which implements Viterbi algorithm for mobile robot localization, so it argued that this paper is one of pioneers that implements Viterbi algorithm for localization and the Viterbi algorithm is modified for efficiency exploiting the physics of the problem at hand. Gaussian Sum Filter which is direct extension of Kalman Filter is used and Gaussian Sum Filter has not been used for global localization and this research presents

the first results. Lastly, a novel differential evolution based approach which in effect imitates restricted mating, is proposed.

The experimental results imply that Particle Filter, Gaussian Sum Filter and Differential Evolution are the most successful ones. However, the success of Viterbi Algorithm which is evaluated for localization first in this work is comparable to those of others. Path-matching success may vary depending on the shape of the road network being considered. Lastly, it is observed that the past trajectory of the mobile robot along with the road network has big impact on the results.

As for the future works; it has been observed that 3D model of the environment and/or 3D scan of the environment by the robot may result in better results, but the attractive feature of (2D) satellite image is that they are freely available on the Internet. Finally, it is believed that virtual GPS may find wide application in car navigation in urban areas.

CHAPTER 7

SIMULATIONS AND EXPERIMENTAL RESULTS

7.1 Introduction

In this chapter, the experimental studies are presented to verify the proposed solution to the *Global Urban Localization* (GUL) problem. In chapter five, a classical solution and in chapter six several novel solutions are presented for GUL problem. These are:

- Monte Carlo Localization
- Viterbi Algorithm
- Markov Localization
- Gaussian Sum Filter
- A novel solution based on differential evolution

Few experimental results are presented to demonstrate the performance of these methods, in chapter five and six and the thorough investigation of these methods through simulations are postponed to chapter seven (Simulations and Experimental Results Chapter).

Here these methods are investigated thoroughly and detailed examination of results is made. Simply three tests are conducted in well developed urban environments. These are:

- Birlik district test
- Yıldız district test
- Dikmen district test

The satellite images of these regions are downloaded from Google Earth software and converted into maps by using the methods discussed in chapter

four, namely the hybrid method composed of FCM and ANFIS. Note that this hybrid method is developed in chapter four of this study.

The experimental results are presented in a systematic way. First the results are simulations are printed in discrete simulation steps, because it is almost impossible to print all simulation steps on one image. The color codes of the simulations are given in figure caption. The color code and color order is the same for all three tests. The path computed with GPS data is also printed to check the simulation steps visually. The numerical check is made by using entropy and KL-divergence metrics. Entropy metric is suitable for checking statistical characteristics of localization whereas KL-Divergence metric checks the characteristics and distribution of particle with respect to a model/GPS data. The laser range finder polar plots are printed on the satellite image to show how compatible they are. The results are demonstrated in a systematic way by using: Entropy metric, KL-Divergence metric, one-sigma confidence interval and comparison of simulation results with ground truth data. Besides these numerical metrics, pictorial descriptions are used to display and compare these results as snapshot.

The rest of the chapter is organized as follows: First test sites are described, then the novel approach based Monte Carlo Localization is investigated. The last part of the chapter is devoted to cross examination of several methods proposed in this study.

7.2 Test Sites

Three field tests were conducted in

- Dikmen district
- Birlik district
- Yıldız district

The satellite images of these local regions are given in Figure (7.1-7.3). In those Figures, the ground truth robot path obtained from the GPS coordinates are plotted in blue for Dikmen, in red for Birlik and in green for Yıldız district.

Figure 7.1 Dikmen district and the path of mobile robot in blue line

Figure 7.2 Birlik district and the path of mobile robot in red line

Figure 7.3 Yıldız district and the path of mobile robot in green line

As it is apparent from the figure, these sites are well developed urban settings. Therefore, there are enough landmarks i.e. buildings, and the amount of vegetation is limited. Indeed, experimental studies have shown that vegetation is not a favorable landmark, especially single tree and bushes are not displayed in satellite images truly because the top and side view of these elements are not consistent. Besides, these elements scatter the laser beam and affect the method proposed here adversely.

These experimental sites have different characteristics;

- Dikmen test site has long corridors and a single curved path patch, the number of turns is the least among the test sites considered here. It looks like a distorted rectangle
- Birlik test site has a complex path at the start and the number of turns is the highest and to the end of test drive the path is almost straight, the final part has again many turns.

- Yıldız test has a polygon path with straight corridors. Note that, Yıldız test site and Dikmen test site have got similar geometric properties.

The approximate path length changes between 1.7 km and 2.2 km. These sites are selected because their nature represents a true urban scenario. These experiments were conducted late at night when the sensor is not disturbed by nearby moving objects. Therefore it can be claimed that the environment is quasi-static. The utility vehicle is a Fiat Doblo whose roof is far above the height of the any passenger cars, so car and other objects which are not modeled in satellite image truly are eliminated because their position can not be estimated. At the time of experiments, cars and other objects are in park positions. However there are times, heavy duty vehicles, trucks and bus are encountered. These vehicles certainly disturbed the characteristics of measurements. However the simulations have shown that their effects on the results are insignificant.

Although the experimental sites are old part of the city, where urbanization has been completed there are land pieces where the satellite image models assumes unoccupied but the reality turns out that there is a building there. At few instances there are constructions sites around, which cannot be guessed or modeled with satellite images. All these errors are corrected manually according to our observations.

During the experiments, the GPS coordinates are received from an electronic device and the laser range finder measurements are recorded at constant time intervals (5 second), the utility vehicle had been driven by a human operator who follows a certain path at constant speed (1^{st} gear at speed 10-12 km/h). This driving condition justifies the use of kinematic model developed in previous chapter. However the method which is suggested for the solution of global localization of a mobile robot/platform requires that the odometer and heading parameters are measured. These data are computed manually, using GPS data. Note that, in literature there are a number of studies which suggest

the use of odometer, compass and gyro together. The odometer data gives the translation and heading information, but odometer measurements are incremental measurements (it measures the internal parameters). Therefore, the localization method which uses the odometer alone will certainly diverge. In order to overcome this difficulty, researchers fuse data coming from different resources. Most studies uses Kalman Filter for sensor fusion and odometer, compass and gyroscopes are common elements in such a sensor fusion. The odometer measures the relative translation and heading information, gyroscope measures the relative heading, its output is unjammable and insensitive external disturbance, however gyroscopes suffers from bias drift and their measurements becomes it becomes unreliable in long run due to heating. Hence compass which is an absolute measurement device which senses the heading is used together with gyroscope and odometer to reduce the heading error.

In this study, it is assumed that the mobile platform is equipped with a virtual odometer and virtual compass whose parameters are computed by using the GPS data. The computation of translation and absolute heading can be justified easily with examples in literature.

Note that the resolution of LMS-291, laser range finder is 10 mm, typical systematic error is 35 mm and standard deviation is 10 mm. The accuracy of the laser range finder is much less than the resolution of the satellite image used in this dissertation which is approximately one pixel equals to two meters. Therefore the error in readings is negligible when compared to the virtual laser range finder reading generated by using the satellite images. Indeed, several authors (Wulf 2004, Sun 2002, Triebel 2006) used laser range finder to generate 3D map of the outdoor and indoor environments successfully. In (Ye 2002), response of SICK laser range finder was investigated under different conditions; target surface properties, incidence angle. The followings statistical data was computed under different conditions. These data is summarized here

for the sake of completeness and to justify that noise in sensor reading is negligible when compared to the resolution of the maps used in this study.

Thirteen target tests were done. The target were placed at 192 mm and moved to 3692 mm in 500 mm steps. The following statistical properties were computed:

Table 7.1 Standard deviation of laser scanner (Ye 2002)

Shiny Color	Cardboard with gold-colored	$\sigma = 4.0\ mm$
	Cardboard with Silver-colored	$\sigma = 3.5\ mm$
	Polished aluminum	$\sigma = 2.1\ mm$
Matted Colors	Red	$\sigma = 4.3\ mm$
	Purple	$\sigma = 4.0\ mm$
	Yellow	$\sigma = 3.9\ mm$
	Blue	$\sigma = 3.8\ mm$
	Green	$\sigma = 3.4\ mm$
Gray Levels	RGB=255	$\sigma = 3.0\ mm$
	RGB=191	$\sigma = 4.7\ mm$
	RGB=127	$\sigma = 5.5\ mm$
	RGB=63	$\sigma = 6.5\ mm$
	RGB=0	$\sigma = 5.5\ mm$

In order to measure the effect of incidence angle thirteen different angles are set: 0, 10, 20, 30, 40, 50, 60 degrees and 10,000 samples were collected. It was observed that the incidence angle between -20 and +20 degrees has insignificant effect on the readings. At 30 degree a large deviation is observed and between 30 and 60 degrees the effect of incidence angle on readings is not significant. Sixteen millimeter difference between the largest mean distance measured and the smallest mean measured at incidence angle of plus/minus 60

and plus/minus 10 degrees was computed. It is claimed that incidence angle has a larger impact on the mean of measured distance than target gray level (Ye 2002). Note that the information about laser range finder extracted from (Ye 2002) suggests that laser range finders are accurate enough for the objectives of this dissertation. Especially when the resolution of map is taken into account, the sensor readings statistical characteristics is negligible.

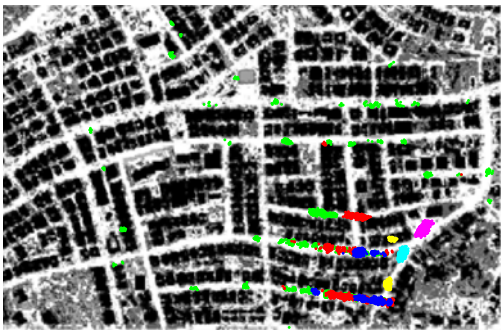
Next the experimental verification of Monte Carlo Localization based solution will be elaborated.

7.3 Experimental Verification of Solution for GUL Problem Based on Monte Carlo Localization

The Monte Carlo Localization technique is proposed as an integral part of the primary solution to GUL problem defined in chapter five of this study. A

Figure 7.4 GPS coordinates of selected simulation steps in Dikmen test, are printed on satellite images. Note that, the simulation step starts from 1 whereas GPS coordinates labels starts from 11.

Figure 7.5 Odometer and compass data are plotted on map, Dikmen Test



Simulation step: 5,10,15,20,25,30

Simulation step: 40,50,60,70,80,90

Simulation step:100, 110, 120, 130,

140, 150, 163

Figure 7.6 Simulation results for Dikmen test *Color code: green, red, blue, yellow, magenta, and cayenne.* Each color is assigned to a simulation step, in order.

limited experimental verification is presented in chapter five. Here this method is examined with intensive experiments. These experiments are conducted in:

- Dikmen district
- Birlik district
- Yıldız district

The nature of these sites are discussed in preceding section, here only the results and experimental simulations are discussed.

7.3.1 Dikmen District Test Results

As the vehicle navigates around, the laser scanner measures the relative distance to nearby landmarks and the virtual odometer records the relative displacement of the vehicle. The process noise is characterized by a Gaussian distribution whose standard deviation is 4 pixels and the standard deviation of measurement noise is 28 pixels. The selected GPS coordinates at different locations are plotted in Figure 7.4. These GPS coordinates are used to compute the output of virtual odometer which is published in Figure 7.5. Note that in this study it is assumed that the vehicle heading angle is measured by a compass. Figure 7.6 displays the simulation results at successive position. The complete GPS data is plotted in Figure 7.7.

Once the simulation is started, the particles concentrates on parallel roads and the Monte Carlo Localization method certainly tracks multiple hypothesis on those parallel horizontal roads. Even this single experiment verifies that tracking methods which holds a unimodal belief in position would certainly fail i.e. Extended Kalman filter. Note that most of the urban sites have symmetrical locations and roads are almost mutually perpendicular to each other. This is true for Dikmen district. One of the peculiar aspects of Dikmen test is that: the road that is followed by the vehicle is simple and almost represents a rectangular road patch. Indeed one can obtain multiple paths by scaling the ground truth data and fit it into the satellite images. Hence the mobile robot can only detect its location on the curved patch or when it turns a corner. The

uncertainty in robot belief in its position drops dramatically when it turns the first corner.

As the vehicle navigates along a road, its belief in position gets worse, because it can only detect the road and almost nothing else. On a road there are buildings on both sides and gaps between buildings, this has the same effect of an indoor corridor, whereas government buildings and road crosses are ideal locations to position the mobile robot because they have distinct signature which reduces the uncertainty in position. These observations may be verified by examining Figure 7.6. The MCL technique used with map is able to localize the vehicle at 20^{th} simulation. The rest of simulation is reduced to a position tracking. In Figure 7.8, several laser scanner data is plotted on the satellite images to compare them with local settings. It is observed that some of these laser scanner outputs are distorted because of vegetation. The entropy metric and KL-Divergence metric plotted in Figure (7.9, 7.10), respectively supports the simulation results.

Figure 7.7 GPS coordinates of Dikmen path

Figure 7.8 Satellite Image augmented with laser range finder polar scan

The entropy metric starts from approximately 3 and it drops to almost 0.5 steadily until 20th simulation step. Then it oscillates around 0.5 until 60th simulation step. Note that between 20th and 60th simulation step the robot is moving along the curved path and after 60th simulation step and then the vehicle enters the upper horizontal road. This road is almost straight and the corridor dilemma is faced with again. There are no distinct landmarks around to reduce the uncertainty in position of the robot so the particle distribution gets coarse. Note that, the robot cannot be localized along this road and particles are spread on the road is stretched along the path, so entropy metrics starts increasing until 136th simulation step where it turns another corner. This maneuver drops the entropy again, dramatically. It is interesting to note that once the vehicle enters the last vertical road, entropy starts increasing again due to corridor dilemma.

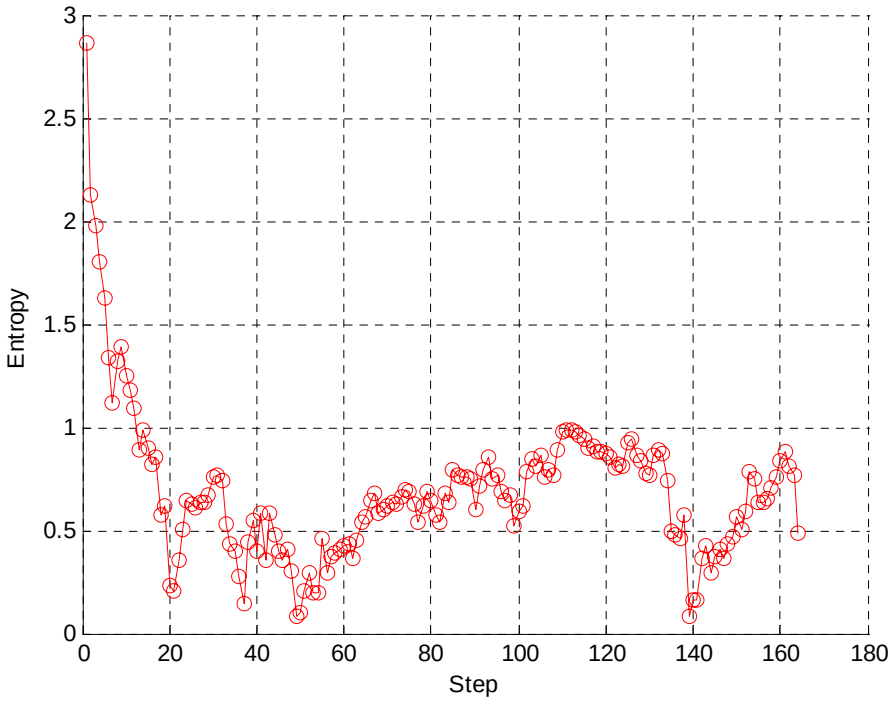


Figure 7.9 Entropy metric for Dikmen test

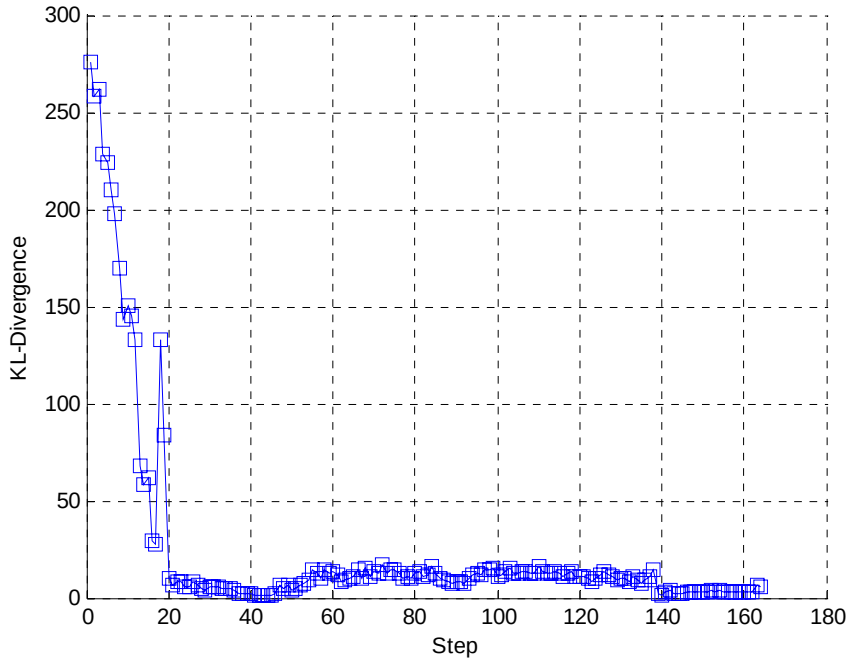
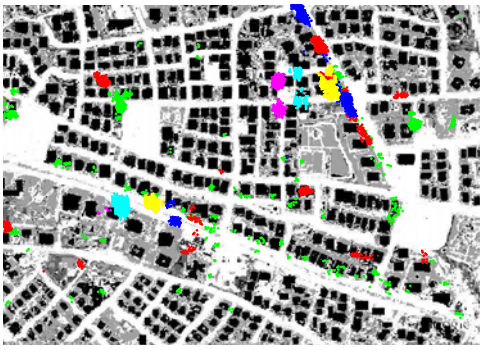


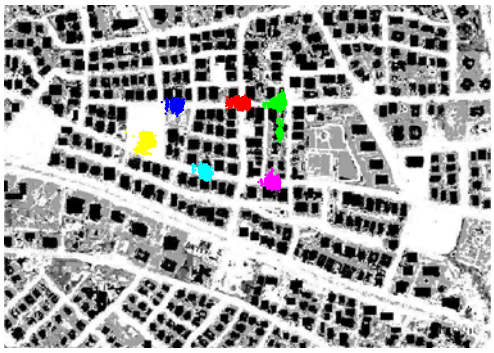
Figure 7.10 KL-Divergence metric for Dikmen test

Figure 7.11 GPS coordinates of selected simulation steps of Yıldız test, are printed on satellite images. Note that, the simulation step starts from 1 whereas GPS coordinates labels starts from 3.

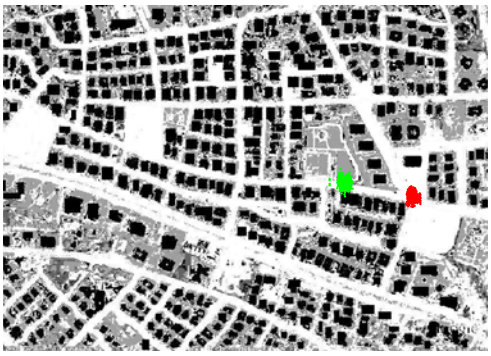
Figure 7.12 Odometer, compass data plotted on map, Yıldız Test



Simulation step: 5,10,15,20,25,30



Simulation step: 35,40,50,60,70,80



Simulation step: 90,100

Figure 7.13 Simulation results for Yıldız test *Color code: green, red, blue, yellow, magenta, and cayenne.* Each color is assigned to simulation step, in order.

7.3.2 Yıldız District Test Results

A second experiment is conducted in Yıldız district. In this simulation the standard deviation of process and measurement noise are 2 pixels and 20 pixels respectively. The selected GPS coordinates of the path and the vehicle path computed by the virtual odometer are plotted in Figure 7.11 and 7.12, respectively. Figure 7.12, shows the map created from satellite images. This map is used with Monte Carlo Localization method to localize the vehicle. Note that the incline road at the start of navigation has a distinct signature if the road network is considered. This helps reducing the vehicle uncertainty in great deal and as the vehicle turns the first corner, the uncertainty in vehicle position drops dramatically. After it turns the second corner, the vehicle location is determined with full confidence. Then the simulation reduces to position tracking as it is the case in previous test. Particle Filter results are published in Figure 7.13 by using a color wheel.

If the general outlook of Yıldız district is analyzed, it is observed that: The start position has a distinct signature, an inclined road and large open space. These signatures are very helpful at the start of simulation. The corners again play a critical role in increasing the robot belief in its position. The entropy decreases to 0.5 steadily until 20^{th} simulation step and then oscillates between 0.5 and 1. Note that between 40^{th} and 60^{th} simulation step which corresponds to the horizontal road patch after the vehicle turns the third corner, there is an increase in entropy. After it travels across the open space and enters the last corridor the entropy oscillates around one. The examination of KL-divergence metrics has shown that the vehicle cannot recover the true position along the road after 60^{th} simulation step because it does not turn any corners thereafter.

The sample laser scanner data is plotted in Figure 7.14 where the simulation step is printed next to the each separate window. The complete GPS data is printed in Figure 7.15. The entropy and KL-divergence metrics are plotted in Figure 7.16, 7.17, respectively.

Figure 7.14 Satellite Image augmented with laser range finder polar scan

Figure 7.15 GPS coordinates of Yıldız path

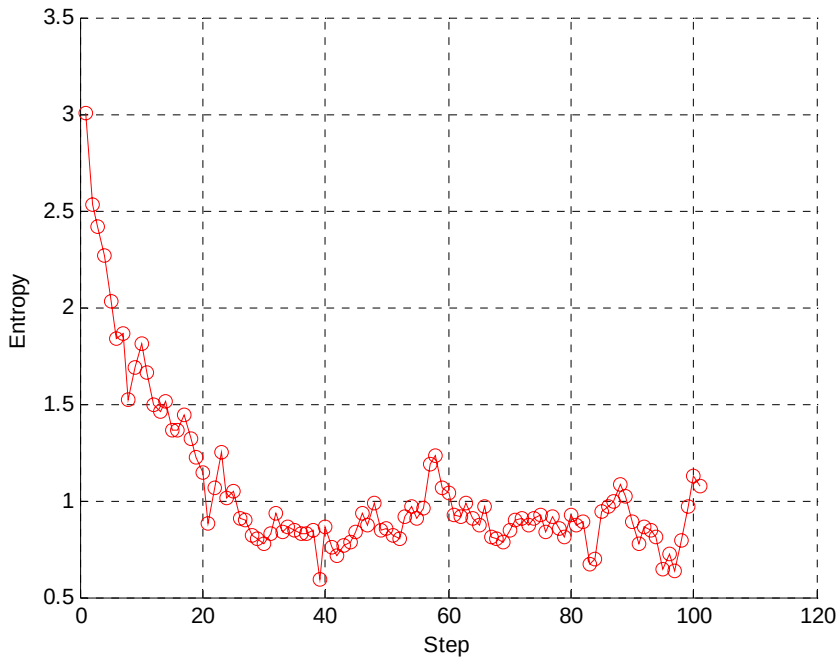


Figure 7.16 Entropy metric for Yıldız path

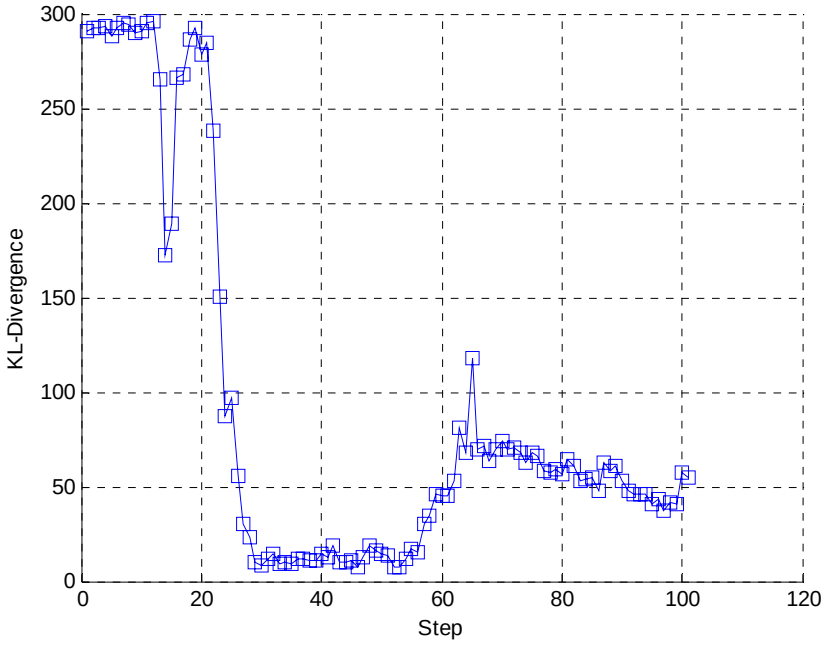


Figure 7.17 KL-Divergence metric for Yıldız path

Figure 7.18 GPS coordinates of selected simulation steps of Birlik test, are printed on satellite images. Note that, the simulation step starts from 1 whereas GPS coordinates labels starts from 11.

Figure 7.19 GPS coordinates of Birlik path

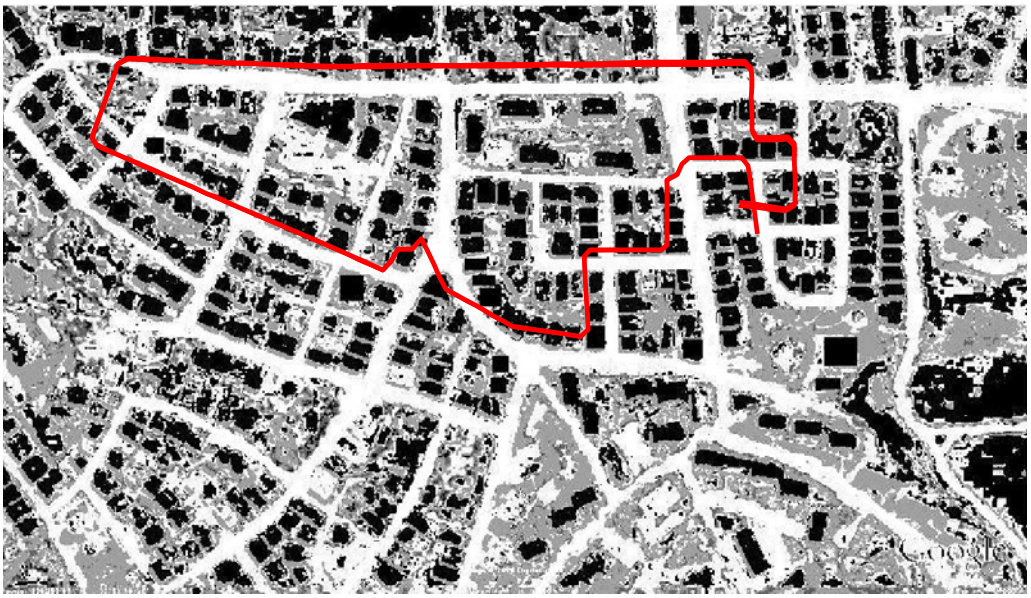
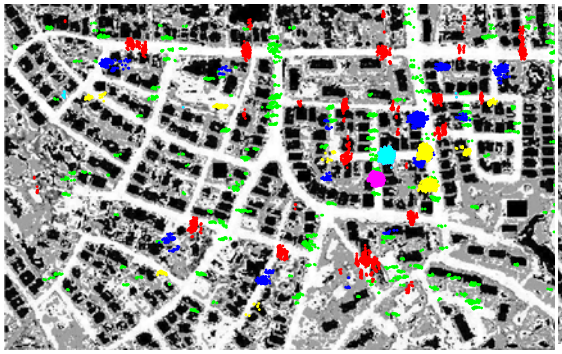
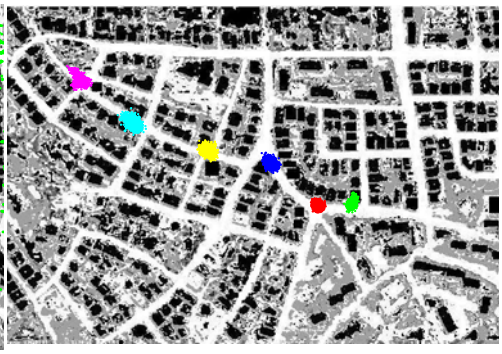


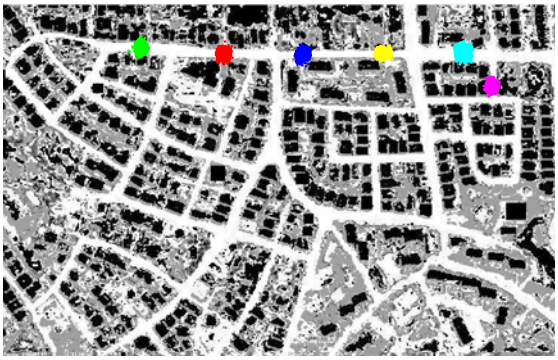
Figure 7.20 Odometer, compass data are plotted on map, Birlik Test



Simulation step: 5,10,15,20,25,30



Simulation step: 35,40,50,60,70,80



Simulation step: 90,100,110,120,130,140

Simulation step: 150,158

Figure 7.21 Simulation results for Birlik test, *Color code: green, red, blue, yellow, magenta, cayenne.* Each color is assigned to simulation step, in order.

Figure 7.22 Satellite Image augmented with laser range finder polar scan

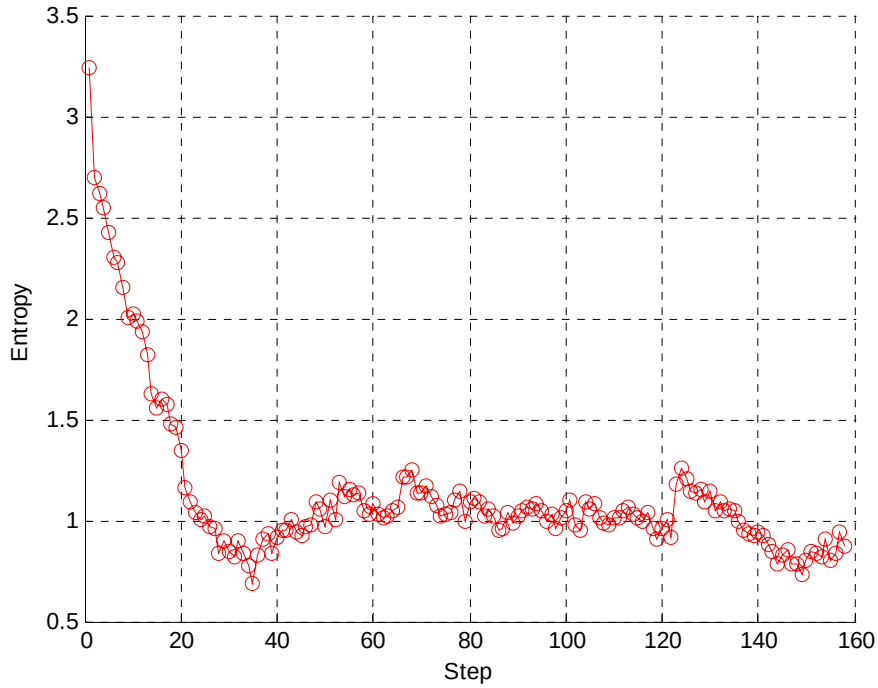


Figure 7.23 Entropy metric for Birlik test

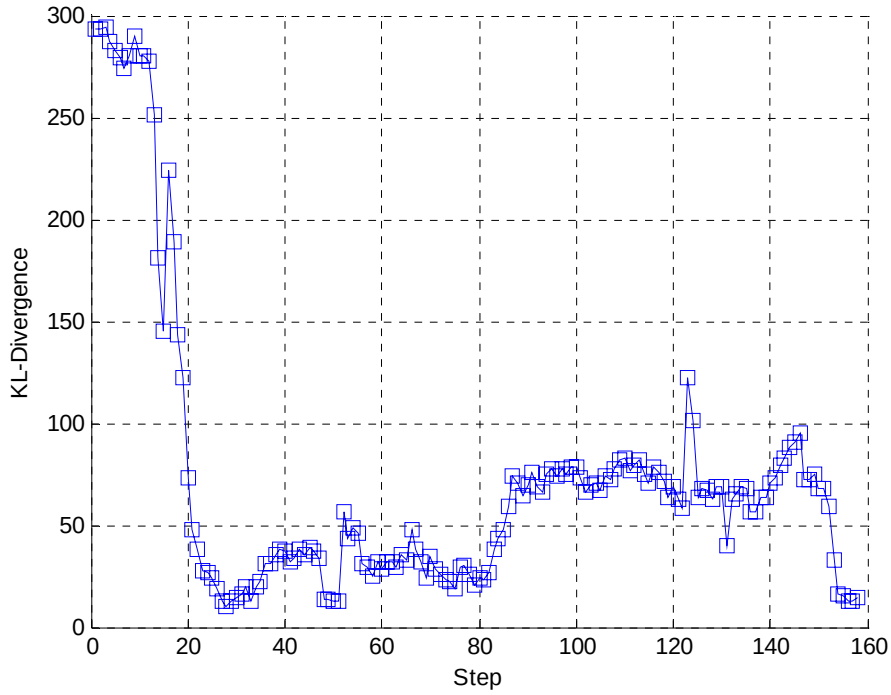


Figure 7.24 KL-Divergence metric for Birlik test

7.3.3 Birlik District Test Results

The selected GPS coordinates are plotted in Figure 7.18 and complete set GPS coordinates are printed in Figure 7.19. The output of virtual odometer is plotted in Figure 7.20. Figure 7.21 shows the simulation results with color wheel. The process and measurement noise are assumed to be Gaussian and the standard deviation of these Gaussians are 2 pixels and 20 pixels. As it is apparent on the figure, the robot is able to localize itself after several corners, because corner and road cross are ideal places with distinct signatures.

The selected laser scanner output is printed in Figure 7.22. This figure also supports the idea that road crosses have ideal signatures for localization and roads are nothing but have the effect of corridor with gaps between buildings are door along the corridor. Hence it may be reasoned that there is no way out for a mobile robot navigating along a straight road to localize itself. It certainly needs some discreet locations to localize itself. This will be discussed in conclusion and discussion part of this chapter with examples. The entropy and KL-Divergence metric are plotted in Figure 7.23-24.

7.4 Comparison of Proposed Method with Alternative Techniques

In the following script, Monte Carlo Localization is compared with alternative solution methods. These are:

- Monte Carlo Localization
- Viterbi Algorithm
- Markov Localization
- Gaussian Sum Filter
- A novel solution based on differential evolution

7.4.1 Yıldız District Test Results

In order to verify the methods evaluated for GUL problem, several experiments are conducted: one of them is presented here. This experiment was conducted

in Yıldız district in Ankara city. Approximately 1.6 km path has been traversed and polar scan of the environment was obtained by a laser range finder while the approximate displacement of the mobile platform has been recorded by a virtual odometer. Thus, these two data sets are fed into the afore-mentioned algorithms. The results are published in the form mean and standard deviation of the position and compared with ground truth position obtained by GPS. All methods start with random grid points. The ground truth path is printed on Figure 7.15. Note that the y axis of the Cartesian coordinates of the image is facing downwards and the position data is printed in pixels.

The simulation parameters are as follows: In particle filter, the process and measurement noise are assumed to be Gaussians whose standard deviations are 2 pixels and 20 pixels respectively and the number of particles used in this simulation is 2000. In Viterbi algorithm, the number of grid points is set to 1000 and diagonal elements of covariance matrix of process model is 8 whereas the off-diagonal elements are set to zero. Using this process noise covariance matrix the constrained Viterbi algorithm is realized as the next states are restricted by motion model. In Gaussian sum filter, the number of Gaussians is set to 1000 and the initial covariance of the EKF is set to square root of the distance between mean of two Gaussians. The diagonal elements of process model covariance matrix are set to 16 pixels and the diagonal elements of the measurement model covariance matrix are set to 900 pixels. In differential evolution, the diagonal elements of motion model covariance matrix are set to one. In path matching the standard deviation of translation in x-direction and y-direction is set to one pixel and the standard deviation of noise in heading is assumed to be 5 degrees. In this simulation uncertainty in heading is taken into account because the road network extracted from the satellite images are thinner than the original road network and the particles must align themselves with the road as they evolve.

The results are illustrated in Figures 7.25-7.29. The Particle Filter converges to the true pose of the mobile robot almost after 10^{th} simulation step and then it

tracks the position of the mobile robot very accurately. The one-sigma confidence interval for the computed position is very tight which implies that the particles distribution representing pose of robots is very compact. Similarly, Gaussian Sum Filter converges at the 10^{th} simulation step but the one-sigma confidence interval for the early phase of the simulation is less deterministic than the Particle filter, the confidence of Gaussian Sum Filter performance in position tracking is also slightly less than particle filter. However note that Particle filter and Gaussian sum filter converge quicker than the other methods as can be verified by the steep decrease in the standard deviation of x and y positions. After the transient regime, the belief about the position of the robot becomes more robust than other methods. Path matching performs the worst and its convergence is slow because it does not exploit the laser scanner data. Therefore several likely paths i.e. almost parallel roads, appear as candidates for the mobile robot position. It converges after the mobile robot turns several

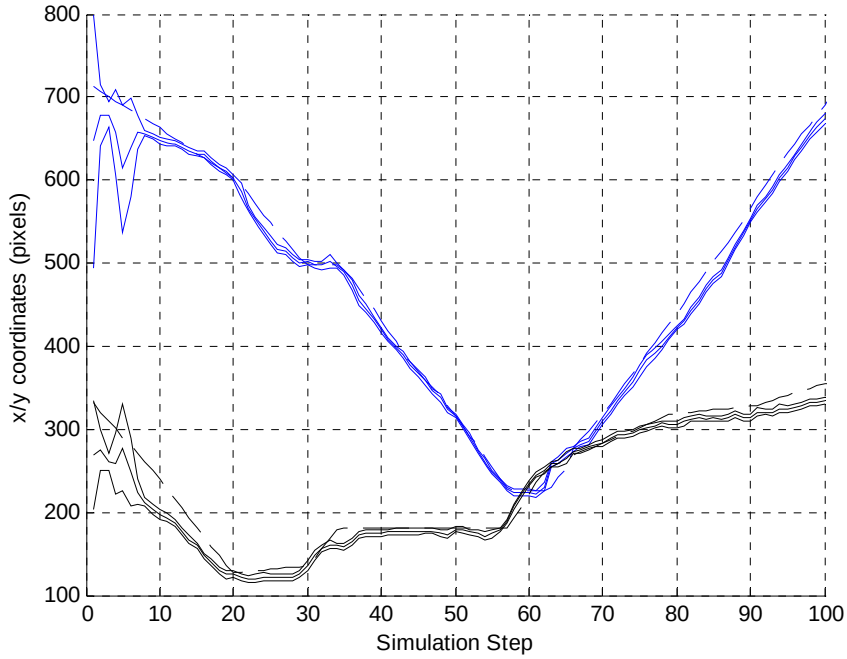


Figure 7.25 Particle Filter Results, Yıldız Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

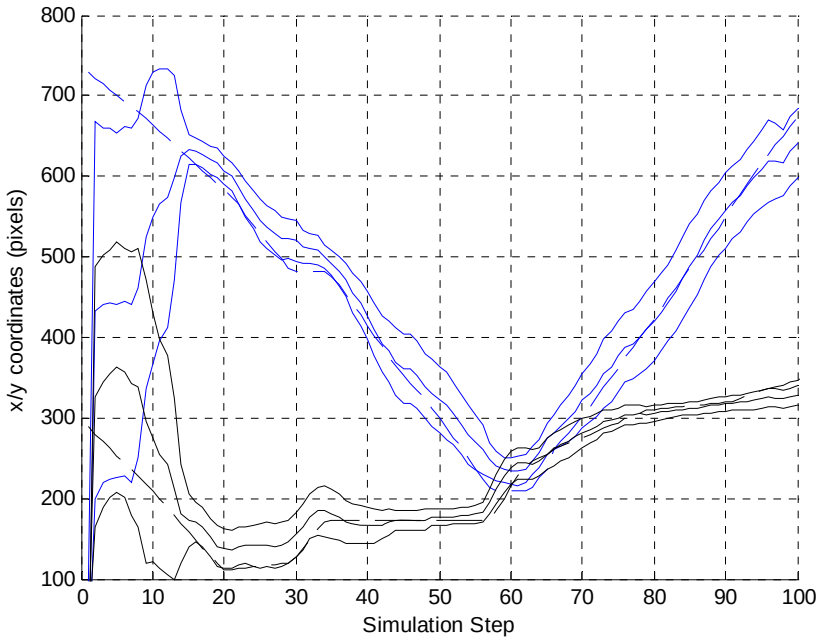


Figure 7.26 Viterbi Algorithm Results, Yıldız Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

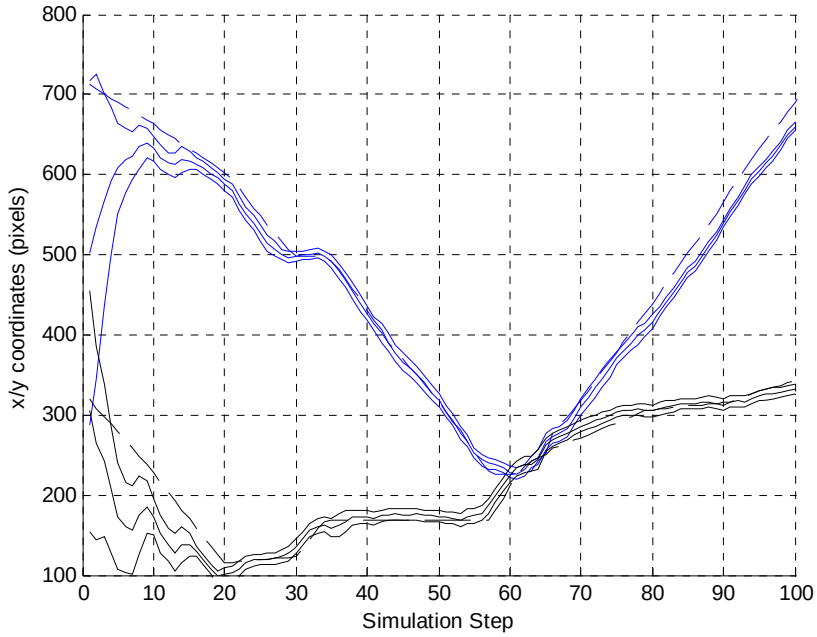


Figure 7.27 Gaussian Sum Filter Results, Yıldız Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

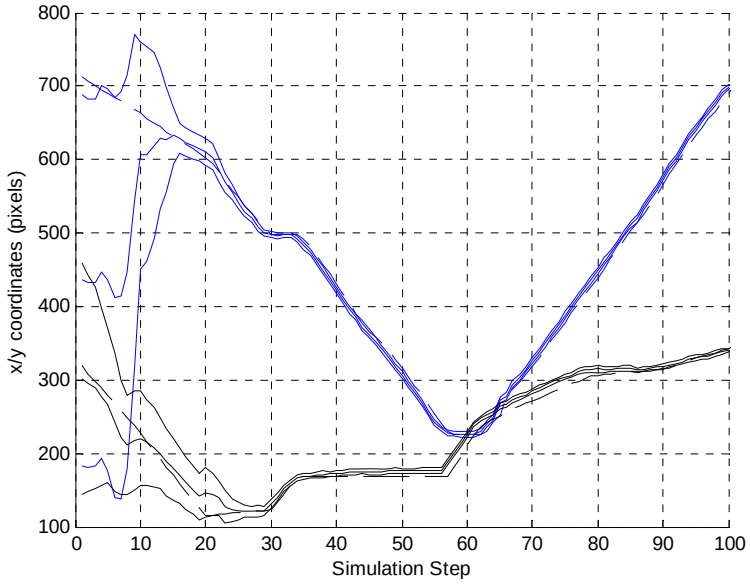


Figure 7.28 Differential Evolution Results, Yıldız Test; blue line represents the x coordinate and black line represents the y coordinates of robot

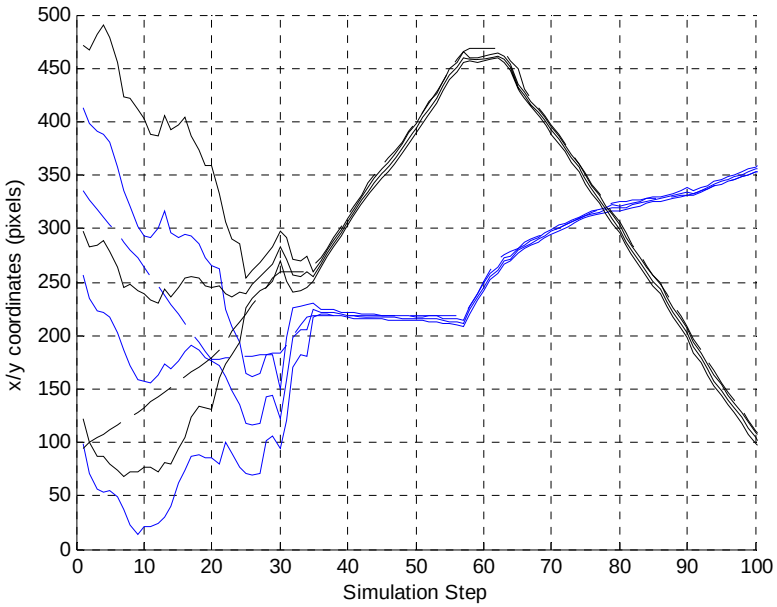


Figure 7.29 Path Matching Results, Yıldız Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose. Note that the scale and size of the image of road networks is different from those of other satellite images

corners. Its convergence is delayed until the 30^{th} simulation step. Although the convergence rate of the Viterbi algorithm is comparable to Particle Filter and Gaussian Sum Filter, its accuracy is less than those of the both methods.

Because Viterbi algorithm run on a fixed grid and size of grid is made comparatively coarse to reduce the computational cost. Lastly, all simulations, including convergence rates and position data, indicate that the shape of the path has significant impact of the results.

7.4.2 Birlik District Test Results

Birlik district is illustrated in Figure 7.2. The approximate path length is 2 km. The apparent feature Birlik test drive is that it has many turns.

The simulation parameters are as follows: In particle filter, the process and measurement noise are assumed to be Gaussians whose standard deviations are 2 pixels and 20 pixels respectively and the number of particles used in this simulation is 2000. In Viterbi algorithm, the number of grid points is set to 1000 and diagonal elements of covariance matrix of process model is 6 pixels whereas the off-diagonal elements are set to zero. Using this process noise covariance matrix the constrained Viterbi algorithm is realized as the next states are restricted by motion model. In Gaussian sum filter, the number of Gaussians is set to 2000 and the minimum distance between Gaussian mean is set to 10 pixels. The initial covariance of the EKF is set to square root of the distance between mean of two Gaussians. The diagonal elements of process model covariance matrix are set to 4 pixels and the diagonal elements of the measurement model covariance matrix are set to 900 pixels. In differential evolution, the diagonal elements of initial covariance matrix are set to 4 and the covariance matrix of motion model is set one. Note that there are 2000 chromosomes at the start and the number of iteration in one simulation step is set to three. In path matching the standard deviation of translation in x-direction and y-direction is set to one pixel and the standard deviation of noise in heading is assumed to be 5 degrees. The number of particles in xy plane is

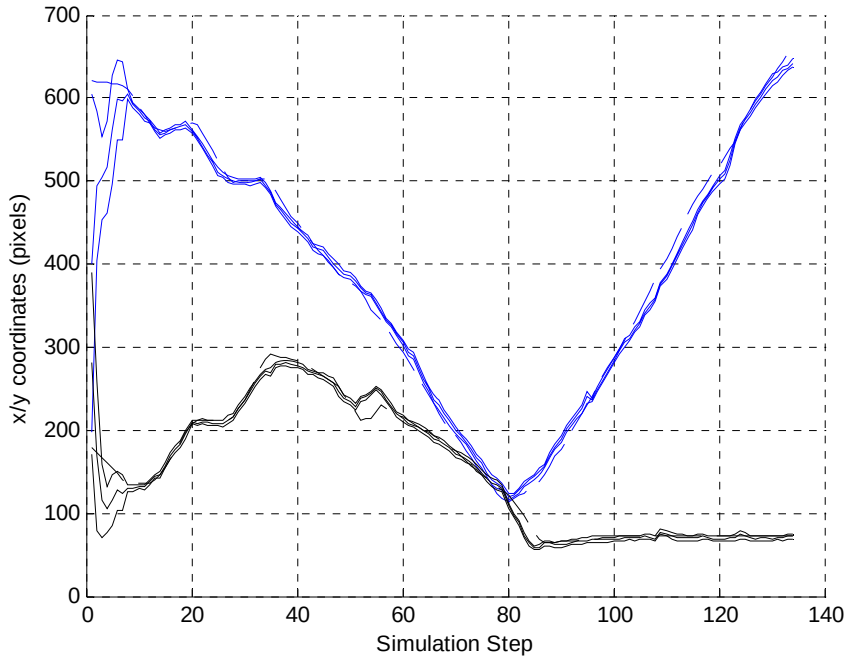


Figure 7.30 Particle Filter Results, Birlik Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

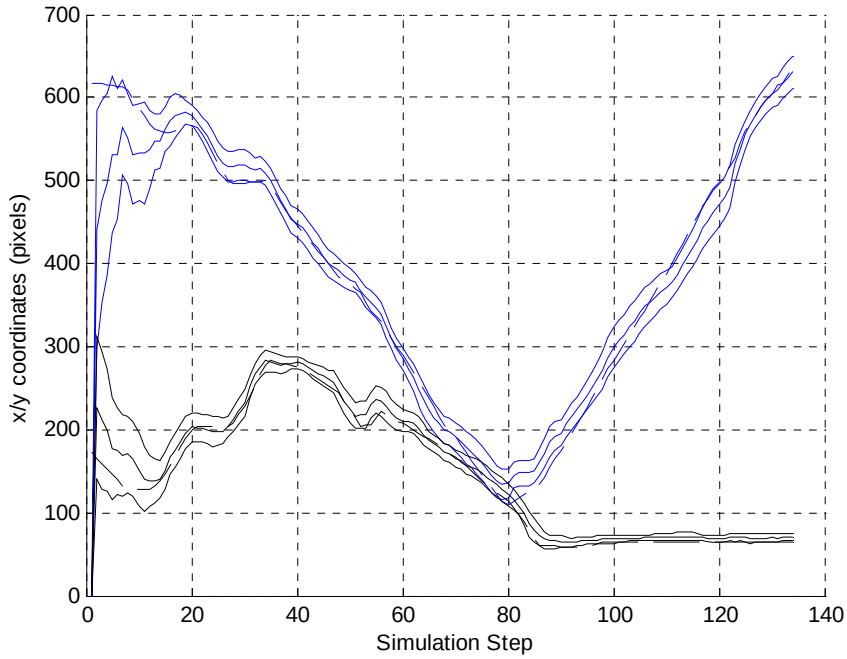


Figure 7.31 Viterbi Algorithm Results, Birlik Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

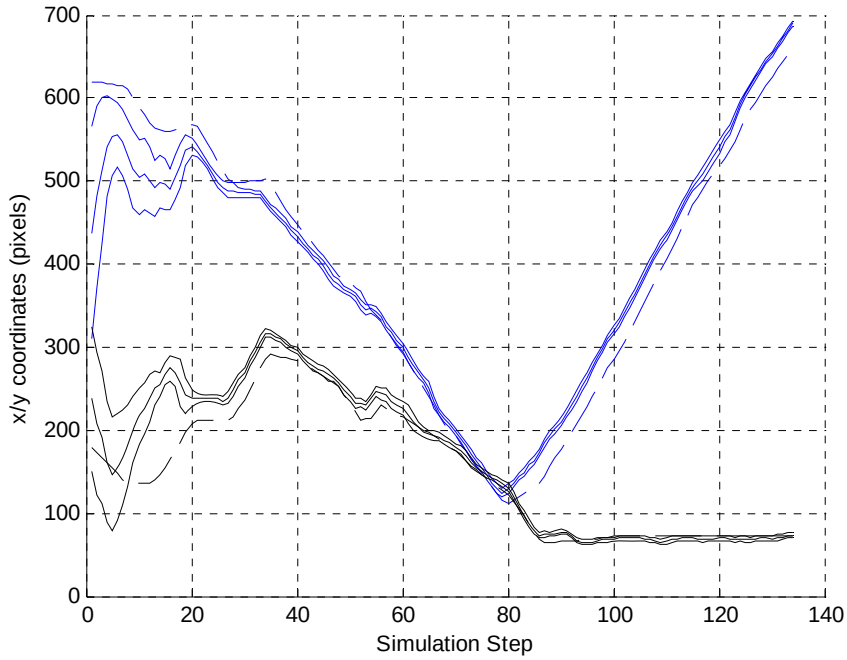


Figure 7.32 Gaussian Sum Filter Results, Birlik Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

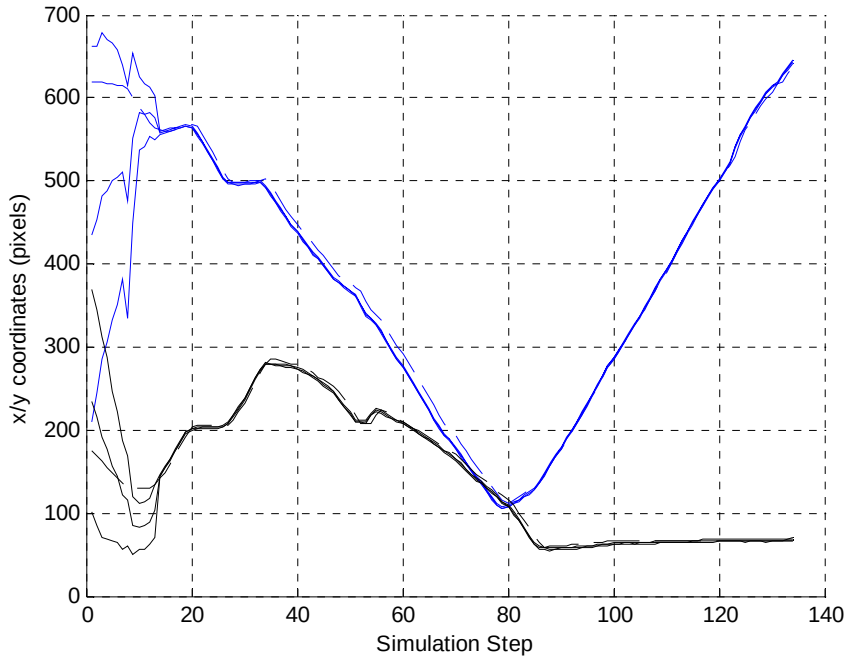


Figure 7.33 Differential Evolution Results, Birlik Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

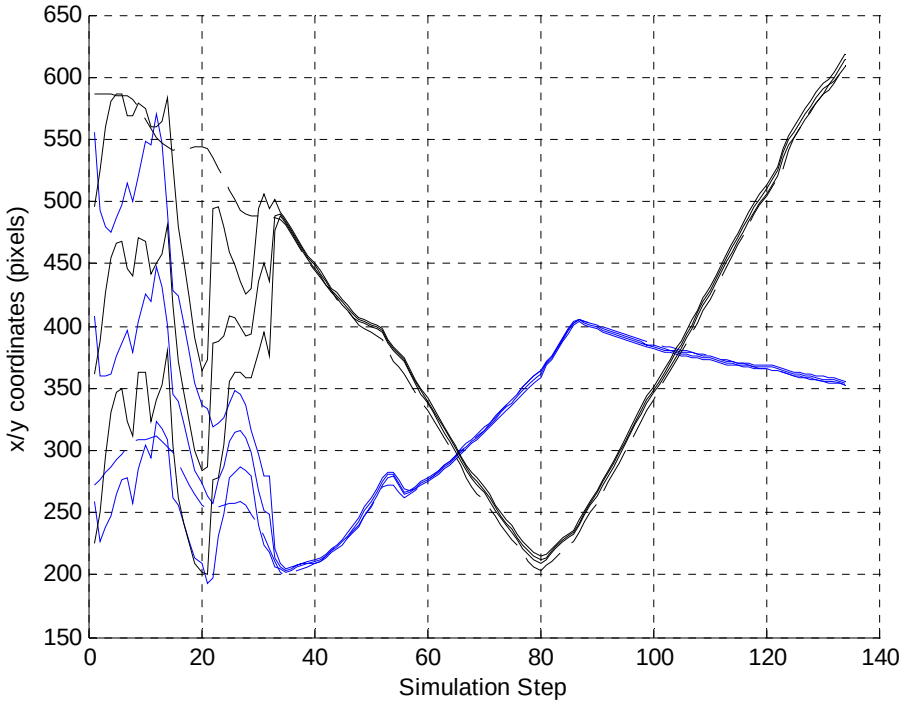


Figure 7.34 Path Matching Results, Birlik Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose. Note that the scale and size of the image of road networks is different from those of other satellite images

set to 500 and for each particle in xy plane there five particles in heading space, so the total number of particles is 2500.

The five algorithms are tested on this path successively and the results are published in Figure 7.30-7.34. Particle filter and Differential Evolution converge faster than the other algorithms and their confidence interval is more compact. Although the transient region of Viterbi algorithm ends at the same simulation step its confidence interval is larger than Particle Filter and Differential evolution. In

In this test drive Gaussian Filter lags the true solution until it finally closes the gap after turning many corners. This also suggests that corners have big impact

on the results. As it is the case, path matching has large deviation from the true solution at the very beginning of the simulation because its observation model is much simpler than other methods

7.4.3 Dikmen District Test Results

The last test drive is made in Dikmen district Note that Dikmen test drive has long corridors and the number of corners is less than Birlik district. Dikmen test drive is very similar to Yıldız test drive in shape of path.

The simulation parameters are as follows: In particle filter, the process and measurement noise are assumed to be Gaussians whose standard deviations are 2 pixels and 20 pixels respectively and the number of particles used in this simulation is 2000. In Viterbi algorithm, the number of grid points is set to 2000 and diagonal elements of covariance matrix of process model is 4 pixels whereas the off-diagonal elements are set to zero. Using this process noise covariance matrix the constrained Viterbi algorithm is realized as the next states are restricted by motion model. In Gaussian sum filter, the number of Gaussians is set to 1000 and the initial covariance of the EKF is set to square root of the distance between mean of two Gaussians which is five pixels. The diagonal elements of process model covariance matrix are set to 36 pixels and the diagonal elements of the measurement model covariance matrix are set to 900 pixels. In differential evolution, the diagonal elements of motion model covariance matrix are set to one and the number of chromosomes encoding robot pose is set to 1000. In path matching the standard deviation of translation in x-direction and y-direction is set to one pixel and the standard deviation of noise in heading is assumed to be 5 degrees.

The ground truth data of Dikmen test drive is plotted in Figure 7.1, and the results of several techniques are plotted in Figure 7.35-7.39. In this test differential evolution is the most successful one and Particle Filter is in second

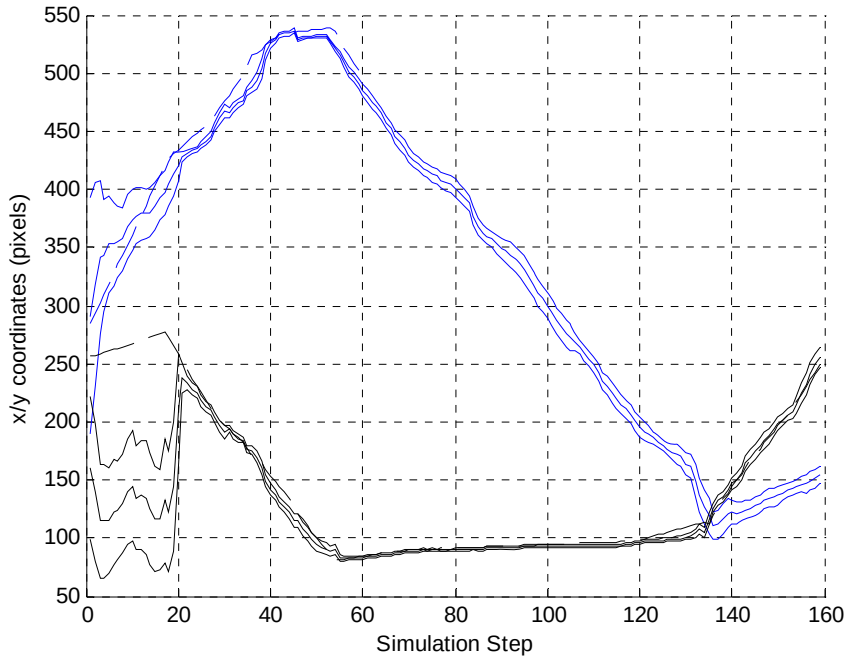


Figure 7.35 Particle Filter Results, Dikmen Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

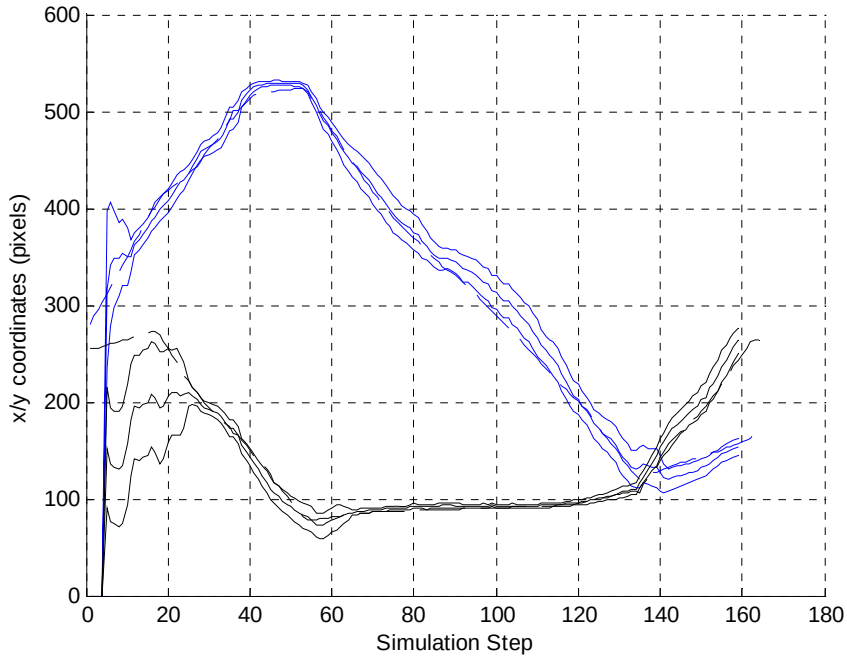


Figure 7.36 Viterbi Algorithm Results, Dikmen Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose

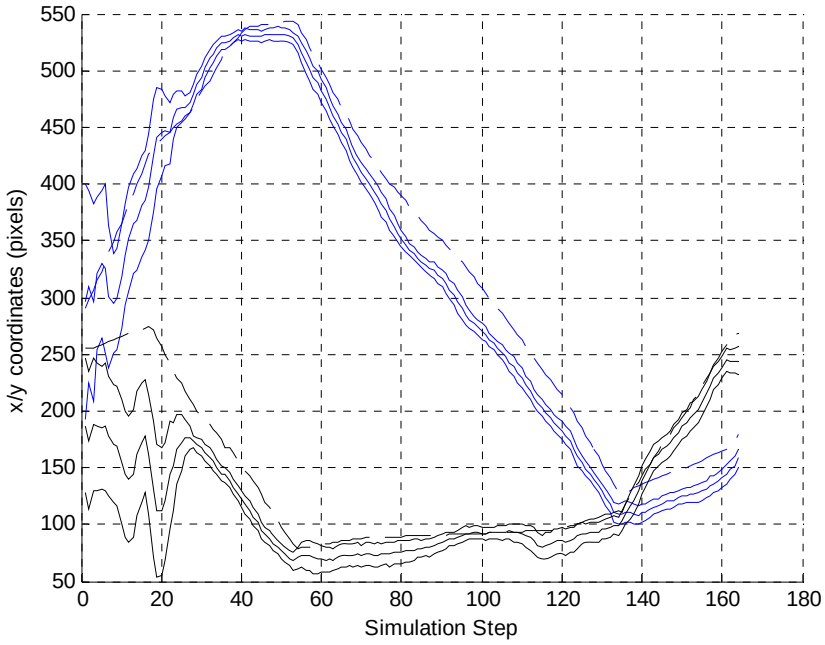


Figure 7.37 Gaussian Sum Filter Results, Dikmen Test; blue line represents the x coordinate and black line represents the y coordinates of robot

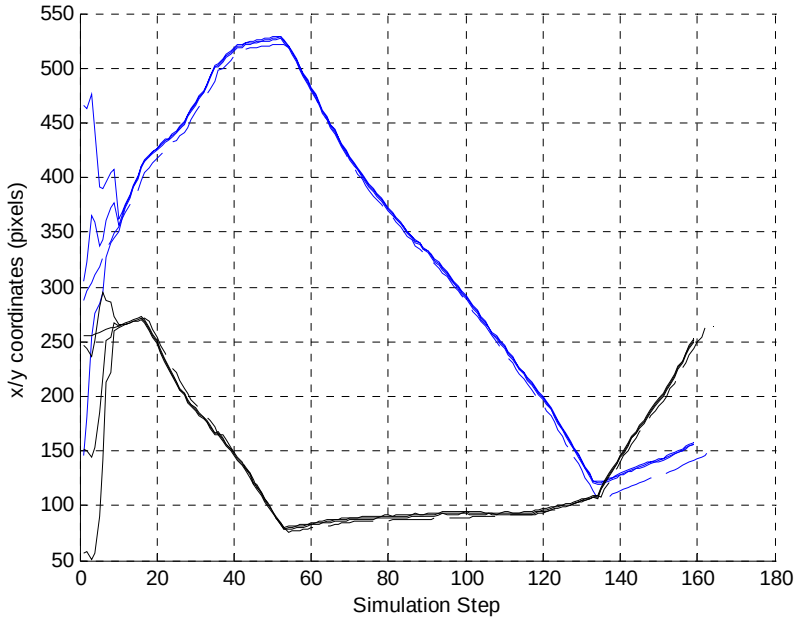


Figure 7.38 Differential Evolution Results, Dikmen Test; blue line represents the x coordinate and black line represents the y coordinates of robot

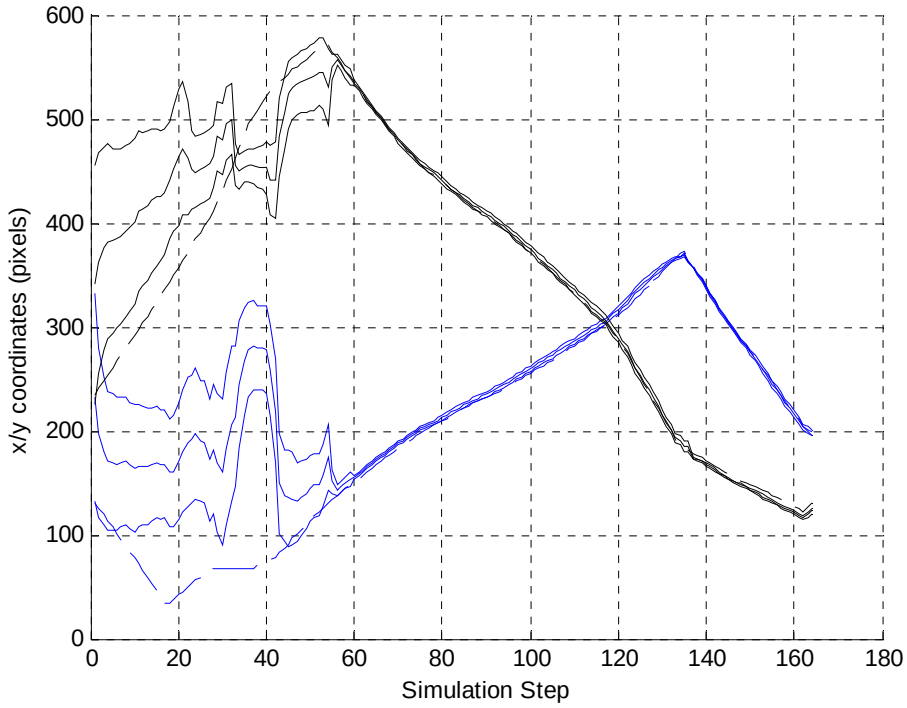


Figure 7.39 Path Matching Results, Dikmen Test; blue line represents the x coordinate and black line represents the y coordinates of robot pose. Note that the scale and size of the image of road networks is different from those of other satellite images

place. Gaussian Sum Filter converges to the true solution at a later time and it lags the true solution until many corners are turned. Path matching is accurate once it converges but it requires that many corners are turned before it converges to the true solution. The Viterbi algorithm suffers from coarse grids and its accuracy is less and confidence interval is less accurate. Note that all algorithms are very sensitive corners. Actually, the corridor dilemma has shown its face in every simulation that is made. Matching past trajectory to the road network has big impact on all simulation results. This will discussed in detailed later

7.5 Conclusions

In this chapter, the following algorithms are tested:

- Monte Carlo Localization
- Viterbi Algorithm
- Markov Localization
- Gaussian Sum Filter
- A novel solution based on differential evolution

Three test sites are selected:

- Dikmen district
- Birlik district
- Yildiz district

All algorithms converge to the true solution but it is observed that past trajectory and the particular shape of the road network has big impact on the results. A robot navigating along a corridor cannot detect its location precisely but the road junctions and government buildings which have different characteristic from the rest of objects/landmarks in the map have reduced the uncertainty in pose belief dramatically.

It has been observed that vegetation poses big problem in sensing and it must be solved at the sensory level. Some objects i.e. construction sites cannot be modeled truly by satellite images since their time period extends the capture frequency of the satellite. Although the experiments were conducted late at night, dynamic object could not be eliminated totally and moving cars were detected, they distorted the sensor readings.

The performance of different filters for three different districts; Dikmen, Yıldız and Birlik will be compared in terms of three parameters, the simulation time step to converge to the true position and the error in localization after global localization is achieved, see Table 7.2.. Indeed once the filter converges

Table 7.2 Performance characteristics of filters

Filter Name	Test	T	μ_x	μ_y	σ_x	σ_y
Particle Filter	Yıldız	10	10.013	9.843	4.107	3.973
	Birlik	9	9.460	4.544	3.752	3.4674
	Dikmen	20	8.704	3.828	6.897	3.598
Viterbi Algorithm	Yıldız	18	17.927	10.348	33.734	16.784
	Birlik	18	12.120	6.057	18.101	9.786
	Dikmen	29	8.685	5.002	13.009	6.513
Gaussian Sum Filter	Yıldız	10	13.548	10.873	8.275	8.646
	Birlik	84	30.196	3.711	4.927	3.291
	Dikmen	24	24.847	13.626	7.581	9.416
Differential evolution	Yıldız	20	7.792	7.794	4.926	5.268
	Birlik	16	6.975	3.907	1.262	1.177
	Dikmen	11	13.067	3.164	3.415	0.779
Path matching	Yıldız	30	4.069	6.760	4.229	4.736
	Birlik	26	4.867	10.786	3.389	4.544
	Dikmen	56	3.675	2.512	2.482	3.244

T : Simulation step at which global localization is achieved

μ_x, μ_y : Mean of tracking error after global localization

σ_x, σ_y : Standard deviation of tracking error after global localization

to the true position it also achieves global localization. First the performance filters in Yıldız test: A close examination of Table shows that Particle filter and Gaussian sum filter outperforms the other filter in terms of convergence and the error after global localization is achieved. Next the filter performance in Birlik test drive. For this test drive particle filer and differential evolutions are ahead of other filter. Surprisingly Gaussian sum filter performance is the worst in this scenario. This is due the fact that Gaussian sum filter approximates the belief in robot pose in terms of Gaussians. Since the computational power of the computer is limited, the number of Gaussians are tried to be kept minimum. Note that the laser scanner records for Birlik tests drive are less successful due to number of reasons; the vegetation in the startup environment and misreading due to a hill and vibration in vehicle. This makes Birlik simulation more challenging because the initial phase of simulation is of the most importance

for global localization. Lastly, Dikmen test drive shows that that Particle filter and differential evolution are the front runner of the proposed techniques. Note that path matching is the least successful one in Dikmen test drive. This may explained considering the structure of environment and the particular path traversed by the robot. Dikmen test drive has the least number of turns. Path matching depends on the most likelihood projection of past trajectory onto the road network, so the global localization is delayed.

CHAPTER 8

CONCLUSIONS AND DISCUSSION

8.1 Introduction

In this dissertation, the global localization of an outdoor mobile robot is studied. The Global Urban Localization, (GUL) problem is described for the first time in this study and solved by means several techniques.

Global localization assumes that a mobile robot is placed in a known environment and the robot has to find its location from scratch. As it was discussed in previous chapter, the Global Urban Localization problem is different from global localization and this point will be elaborated further in this chapter. Note that global urban localization is defined within the context of global localization which is limited to the known local environment. The ultimate objective of GUL problem is to localize a mobile robot anywhere urban on Earth.

"How can a robot be localized on planet Earth?" is the question that is answered in this dissertation. The main restriction implied is that the robot is in an urban setting, and the satellite images of this setting are available. Mobile phone applications such as Google Earth can localize a person on Earth with in a region of uncertainty, the radius of which is generally around 1km. This is information is acquired by the cellular phones without using GPS. In this study, this information is exploited to make a rough guess for initial location of a mobile robot[1]. Then by using this initial guess, the satellite images of the local setting are downloaded from the internet and thereafter global localization

[1] An alternative to the use of a cellular phone for determining the search region is the one obtained through the IP of a present network connection. In such a case it is assumed that a wireless network connection is available for the robot. However, it is to be noted that use of the cellular phone infrastructure will significantly help reducing the initial search area.

techniques are used to make better estimation of the robot pose. Thus, the mobile robot which is localized in local environment is said to be localized globally on planet Earth.

It could be reasoned that the global localization techniques are valid to solve GUL problem. Because once the local region is identified by a mobile phone and satellite images of the local setting are downloaded, the rest of the problem is reduced to localization on an a priori map.

Right at this stage, how the satellite images are processed so as to create maps suitable for robot navigation should be tackled. A number image processing methods are evaluated for this purpose; besides a novel hybrid method is developed.

Now that the objective of the study is summarized and the key points are stressed, i.e. the initial guess, conversion of satellite images to maps and the suitable localization techniques. Next, a road map of this study which forms the backbone of the novel solution proposed here will be described.

8.2 Road Map of Dissertation

First, the satellite images are converted into maps which are suitable for mobile robot navigation. There are number tools which may serve for this purpose. In this study, neural networks are employed for this purpose. These are:

- Feed-Forward Artificial Neural Networks
- Probabilistic Neural Network
- Fuzzy-C Means

Besides these conventional tools, a hybrid approach composed of Fuzzy C-Means and ANFIS is proposed to convert satellite images into maps. The performances of these tools are compared in chapter four.

Once these maps are obtained, the localization problem is considered as a particular case of probabilistic state estimation problem i.e. tracking, is investigated in chapter three.

The localization problem on a given prior map i.e. maps obtained from satellite images are studied in chapter five with Monte Carlo Localization technique. The next chapter evaluates alternative methods. Some of them are either developed in this study or evaluated here for the first here. These are:

- Gaussian Sum Filter
- Novel localization based on differential evolution
- Constrained Viterbi algorithm
- Path matching

Chapter seven verifies the proposed methods with extensive tests.

8.3 Discussion of Test Runs

Three test drives are made in different regions of Ankara, Turkey. These are:

- Yildiz test drive
- Birlik test drive
- Dikmen test drive

The districts are selected among the well developed regions of the city so as to justify the urban environments assumptions. The path of the robot is chosen as a closed route. It has been observed that Monte Carlo Localization is a reasonable choice for the solution of GUL problem. However, the results may be immediately improved by 3D scanning in great deal. All these tests drives are simulated successfully by the proposed method and the alternative solutions proposed in this dissertation. After the test drives are completed, the following results are reached:

- A robot navigating along road is faced with corridor dilemma. That is to say; a robot navigating along road cannot observe many landmarks to detect its location on the road. Because the roads side is occupied with buildings placed at regular distance.
- Road junctions are distinct places where a robot may find landmarks.
- There are buildings with different characters i.e. government buildings, schools etc. These building also serve as a landmark
- It has been observed that when the robot turns a corner the uncertainty in its belief decreases dramatically
- This suggest that the shape of the robot past trajectory has a big impact on localization.
- Sometimes 2D satellite images are not capable of modeling some events i.e. construction sites whose period extends beyond capture frequency of satellites.
- Single tree and vegetation poses big problem for sensing. This must be solved either at the sensory level or an intelligent approach must be developed to discard these sensor information.
- Although the test sites are selected carefully, it is almost impossible to justify planar world assumption in outdoor environments. Therefore, the terrain must also be modeled.
- In this study, the sensor is place on the roof of Fiat Doblo which is higher than any passenger car. The test drives are made late at night but even at those times, there are dynamic objects, and few driving cars are detected. They distorted the sensor information. The distorted sensor information is discarded.
- Dynamic objects must be discarded from the sensor readings effectively.

8.4 Conclusions

The GUL problem is defined in this dissertation and solved for the first time. Extensive experimental studies have shown that the solution of GUL problem

is possible. It is believed that the solution of GUL problem may serve as an alternative to GPS technology. Nowadays, GPS is the prevalent means of outdoor localization. However, it has some shortcomings i.e. multi path effect, occlusion of observable satellites, and inaccurate position data. In fact the use GPS technology is under control and it is known at certain times, the availability of GPS technology is open to question.

The solution of GUL does not require a central unit or advanced technological infrastructure which is developed for this purpose only. GUL uses the available mobile phone infrastructure. Therefore, its usage does not require any hardware investment. This is a great advantage.

As of the autonomous mobile robot navigation, GUL offers more flexibility to the mobile robot, because a mobile robot using GUL does not depend on a central i.e. GPS and it is capabilities are self-contained.

It is believed that GUL will find many practical applications in car navigation, military missile navigation, operation of mobile robots in other planets for the first time or in operation of underwater mobile robots.

8.5 Future Works

In this study, 2D satellite images are employed to describe a scene in which a mobile robot operates. The most striking advantage of such an information source is that it is free of charge and there is no limitation of the source. However, experimental tests have shown that under certain conditions, the 2D satellite images are not capable of describing some of the elements in the scene. For instance satellite images cannot describe a tree truly or a bush. Because of their nature, these elements tend to scatter the laser beam and this cannot be predicted at all. The 2D satellite images assume a planar horizontal environment. However even in urban environment, a mobile robot may encounter up and downs. Hence it is clear that the use of 2D satellite images are limited and more detailed description of the scene is required. Actually

Google Earth models the terrain in 3D as well. This data may also be downloaded from internet and incorporated into the current version of GUL problem.

A complex scene analysis is desirable that takes into account the geometry of the objects in the scene and their relative position. This will help improving the quality of maps. Because it is believed that segmentation based on color content is limited in nature.

It is believed that the 2D satellite images may be converted into abstract 3D scene by using advanced techniques i.e. camera model and an object library of some common shapes for instance prismatic rectangle, hemisphere, tetrahedron and other primitive geometries.

Additionally 3D scanning of the environment will help improving the results in great deal. This will solve most of the problems at sensory level.

Although all experiments are conducted late at night, dynamic objects i.e. truck, bus etc. could not be totally avoided, and hence new techniques must be introduced so as to eliminate these dynamic objects from the sensor reading

Lastly, as it was discussed in experimental results chapter, the shape of the path has a big impact on the results therefore this may be a parallel research direction to the solution method proposed and evaluated for GUL problem[2].

[2] Indeed in chapter seven this alternative is taken into account and path matching is considered as a solution to GUL problem

REFERENCES

1. Abdallah, F., A. Gning, and P. Bonnifait. 2007. Adapting particle filter on interval data for dynamic state estimation acoustics, speech and signal processing. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07) 2: 1153-1156.

2. Allesandri, A., G. Bartoloni, P. Pavanati, E. Punta and A .Vinci. 1997. An application of the extended Kalman filter for integrated navigation in mobile robotics. Proceedings of the American control conference, Albuquerque, New Mexico: 527-531.

3. Amato, S.L., and N.M. J. Fellers. 2000. Localization based on visibility sectors using range sensors. Robotics and Automation Proceedings. ICRA'00. I 4: 3505-3511.

4. Anousaki, G.C., and K.J. Kyriakopoulos. 1999. Simultaneous localization and map building for mobile robot navigation. IEEE Robotics and Automation Magazine 6(3):42-53.

5. Arulampalam, M.S., S. Maskell, N. Gordon, and T. Clapp. 2002. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing 50(2):174 – 188.

6. Ashokaraj, I.A.R., P.M.G. Silson, A. Tsourdos, and B.A. White. 2003. Implementation of an adaptive EKF to multiple low cost navigation sensors in wheeled mobile robots. Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV' 02), Singapore: 608-612.

7. Bailey, T., and H. Durrant-Whyte. 2006. Simultaneous Localization and Mapping (SLAM): Part I the essential algorithms. Robotics and Automation Magazine 13: 99–110.

8. Bailey, T., and H. Durrant-Whyte. 2006. Simultaneous Localization and Mapping (SLAM): Part II the state of the art. Robotics & Automation Magazine, 13(3): 108 – 117.

9. Barnes, I. 2003. Aerial remote sensing techniques used in the managements of archeological monuments on the British army's Salisbury plain training area, Wiltshire, UK. Archeological Prospection 10:83-90.

10. Barshan, B. and H. F. Durrant-Whyte. 1993. An inertial navigation system for a mobile robot. Proceedings of the IEEE/RSJ International Conference On Intelligent Robots And Systems, Yokohama, Japan 23-26: 2243-2248.

11. Basnayake, C., O. Mezentsev, G. Lachapelle, and M.E. Cannon. 2005. An HSGPS, inertial and map-matching integrated portable vehicular navigation system for uninterrupted real-time vehicular navigation. International Journal of Vehicle Information and Communication Systems 1(1/2): 131-151.

12. Bonnifait, P., and G. Garcia. 1999. 6 DOF dynamic localization of an outdoor mobile robot. Control Engineering Practice 7:383-390.

13. Booker, G. 2007.Correlation of millimeter wave radar images with aerial photographs for autonomous navigation UAV. 2nd Int. Conference on Sensing Technology, Palmerston North, New Zealand: 529-533.

14. Borenstein, J., H. R. Everett, L. Feng, and D. Wehe. 1997. Mobile robot positioning: sensors and techniques. Journal of Robotic Systems 14(4): 231–249.

15. Burgard, W., D. Fox, D. Hennig, and T. Schmidt. 1996. Estimating the absolute position of a mobile robot using position probability grids. In Proceedings of the Thirteenth National Conference on Artificial Intelligence.

16. Burgard, W., D. Fox, D. Hennig, and T. Schmidt. 1996. Position tracking with position probability grids. Proceedings of the First Euromicro Workshop on Advanced Mobile Robot: 2-9.

17. Burgard, W., et al. 1999. Experiences with an interactive museum tour-guide robot. Artificial Intelligence 114: 3-55.

18. Christophe A., N. de Freitasi, A. Doucet, and M.J. Jordan. 2003. An introduction MCMC for machine learning. Machine Learning 50: 5-43.

19. Csorba, M., J.K. Uhlmann, and H. Durrant-Whyte. 1997. A suboptimal algorithm for automatic map building. Proceedings of the American Control Conference, Albuquerque, New Mexico: 537-541.

20. Czogala E., and J. Leski. 2000. *Fuzzy and neuro-fuzzy intelligent systems (studies in fuzziness and soft computing)*. Heidelberg, Newyork: Physica-Verlag.

21. Dissanayake, G., P. Newman, S. Clark, H. Durrant-Whyte and M. Csorba. 2001. A solution to simultaneous localization map building (SLAM) problem. IEEE Transactions on Robotics and Automation. 17 (3): 229-241.

22. Dissanayake, G., S.B. Williams, H. Durrant-Whyte and T. Bailey. 2002. Map management for efficient simultaneous localization an mapping (SLAM). Autonomous Robots 12: 267-286.

23. Dogruer, C.U., B. Koku and M. Dolen. 2007. A novel soft computing algorithm to segment satellite images for mobile robot localization and navigation. IEEE / RSJ International Conference on Intelligent Robots and Systems, San Diego, California: 2077-2082.

24. Dogruer, C.U., B. Koku and M. Dolen. 2008. Global urban localization of outdoor mobile robots using satellite images. IEEE / RSJ International Conference on Intelligent Robots and Systems, Nice, France: 3927-3932.

25. Dogruer, C.U., B. Koku and M. Dolen. 2008. Global urban localization of an outdoor mobile robot with genetic algorithms. Springer Tracts in Advanced Robotics, European Robotics Symposium 2008, Springer Berlin / Heidelberg 44:103-112.

26. Doucet, A., N. Freitas, and N. Gordon. 2001. *Sequential Monte Carlo methods in practice*. Springer.

27. Dulimarta, H.S., and A.K. Jain. 1997. Mobile robot localization in indoor environment. Pattern Recognition 30(1): 99-111.

28. Eliazar, A. and R. Parr. 2003. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. Proceedings of 18th International Joint Conference on Artificial Intelligence.

29. Eliazar, A. and R. Parr. 2006. Hierarchical linear/constant time SLAM using particle filters for dense maps. Advances in Neural Information Processing Systems.

30. Eliazar, A., and R. Parr. 2004. DP-SLAM 2.0. Proceedings - IEEE International Conference on Robotics and Automation: 1314-1320.

31. Ernest, P., R. Mazl, and L. Preucil. 2004. Train locator using inertial sensor and odometer. IEEE Intelligent Vehicle Symposium: 860-865.

32. Estrada, C., J. Neira, and J. D. Tardos. 2005. Hierarchical SLAM: Real-time accurate mapping of large environments. IEEE Transactions on Robotics 21 (4): 588-596.

33. Forney, G.D., and J.R. 1973. The Viterbi algorithm. Proceedings of the IEEE 61(3) 268-278.

34. Fox. D. 1998. Markov localization: a probabilistic framework for mobile robot localization and navigation. Ph.D. dissertation, Institute of Computer Science III, University of Bonn, Germany.

35. Fox, D. 2003. Adapting the sampling size in particle filters through KLD-sampling. The International Journal of Robotics Research, 22(12):985-100.

36. Gayan, J., and C.J. Watts. 2005. The use of remote sensing for estimating ET of irrigated wheat and cotton in northwest Mexico. Irrigation and Drainage Systems 19:301-320.

37. Giulio, R., A. Vargas, K. Nagatani, and K. Yoshida. 2007. Adaptive Kalman filtering for GPS-based mobile robot localization. Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics, Rome, Italy: 1-6.

38. Gordon, N., D.J. Salmond, and A.F.M. Smith. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proceedings-F 140(2): 107-113.

39. Guivant, J., and E. Nebot. 2001. Optimization of simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation. 17 (3): 242-257.

40. Guivant, J., and R. Katz. 2007. Global urban localization based on road maps. IEEE / RSJ International Conference on Intelligent Robots and Systems, San Diego, California: 1079-1084.

41. Guivant, J., E. Nebot, and S. Baiker. 2000. Localization and mapping using laser range sensors in outdoor applications. Journal of robotics system. 17 (10): 565-583.

42. Guivant, J., E. Nebot, J. Nieto, and F. Masson. 2004. Navigation and mapping in large unstructured environments. The International Journal of Robotics Research 23: 449-472.

43. Guivant, J.E., R.F. Masson, and M.N. Eduardo. 2002. Simultaneous localization and map building using natural features and absolute information. Robotics and Autonomous Systems 40: 79–90.

44. Hakyoung, C., L. Ojeda and J. Borenstein. 2001. Accurate mobile robot dead reckoning with a precision-calibrated fiber-optic gyroscope. IEEE Transactions on Robotics and Automation 17 (1): 80-84.

45. Harrower, M., J. McCorriston, and E.A. Oches. 2002. Mapping the roots of agriculture in southern Arabia: the application of satellite

remote sensing, global positioning system and geographic information system technologies. Archeological Prospection 9:35-42.

46. Haug, A.J. 2005. A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes. Mitre Organization [cited 09.08.2008]. Available from http://www.mitre.org/work/tech_papers/tech_papers_05/05_0211/index.html

47. Hertz, J., A. Krogh, and R. G. Palmer. 1991. *Introduction to the theory of neural computation*. Westview Press.

48. Howard, A., and L. Kitchen. 1999. Navigation using natural landmarks", Robotics and Autonomous Systems 26: 99-115.

49. Iocchi, L., and D. Nardi. 2002. Hough localization for mobile robots in polygonal environments. Robotic and Autonomous Systems 40: 43-58.

50. Ippoliti, G., L. Jetto and S. Longhi. 2005. Localization of mobile robots: development and comparative evaluation of algorithms based on odometric and inertial sensors. Journal of Robotic Systems. 22 (12): 725-735.

51. Isaev, A.S. et al. 2002. Using remote sensing to assess Russian forest fire carbon emissions. Climatic Change 55:235-249.

52. Julier, S.J., J.K. Uhlmann, and H.F. Durrant-Whyte, A new approach for filtering nonlinear systems. Proceedings of the American Control Conference, 1995, 3 :1628-1632 vol.3

53. Kotecha, J.H., and P.M. Djuric. 2003 Gaussian sum particle filtering. IEEE Transactions on Signal Processing 51(10): 2602-2612.

54. Jetto, L., S. Longhi, and D. Vitali. 1999. Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter. Control Engineering Practice. 7: 763-771.

55. Jetto, L., S. Longhi, and G. Venturini. 1999. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. IEEE Transactions on Robotics and Automation 15 (2): 219-229.

56. Khan, N.I. and A. Islam. 2003. Quantification of erosion patterns in the Brahmaputra-Jamuna river using geographical information system and remote sensing technology. Hydrological Processes 17:959-966.

57. Kim, S., and J.H. Kim. 2001. Adaptive fuzzy-network-based c-measure map-matching algorithm for car navigation system. IEEE Transactions on Industrial Electronics 48(2):432-441.

58. Kim, S.H., C.W. Roh, S.C. Kang and M.Y. Park. 2007. Outdoor navigation of a mobile robot using differential GPS and curb detection. IEEE International Conference on Robotics and Automation, Roma, Italy: 3414-3419.

59. Knoll, A, M. Mueller, and A. Wolfram. 1997. A rule based fuzzy approach to the classification of man made objects in satellite image data. Proceeding of the 4th Fuzzy-Neuro-Systems, Soest, Germany: 247 – 254.

60. Kosecka, J., F. Li, and X. Yang. 2005. Global localization and relative positioning based on scale-invariant key points. Robotics and Autonomous Systems 52: 27-38.

61. Kwok, C., D. Fox, and M. Meila. 2004. Real-time particle filters. Proceedings of the IEEE 92(3): 469 – 484.

62. Lahrech, A., C. Boucher, and C. Noyer. 2004. Fusion of GPS and odometer measurements for map based vehicle navigation. IEEE International Conference on Industrial Technology: 944-948.

63. Lee, D., W. Chung, and M. Kim. 2002. Probabilistic localization of the service robot by map matching. Proceedings of International Conference on Control, Automation and Systems (ICCAS'2002) Muju, Korea: 1667-1627.

64. Leonard, J.J. and H. Durrant-Whyte. 1991. Mobile robot localization by tracking geometric beacons. IEEE transactions on robotics and automation. 17 (3): 376-382.

65. Lin, C., and R.L.Tummala. 1997. Mobile robot navigation using artificial landmarks. Journal of Robotics Systems 14(2): 93-106.

66. Lin, C.T., Y.C. Lee, and H.C. Pu. 2000. Satellite sensor image classification using cascaded architecture of neural fuzzy network. IEEE Transactions on Geosciences and Remote Sensing 38(2):1033-1043.

67. Liu, J. S. 1999. Markov Chain Monte Carlo and related topics" Proceedings of the IX General Assembly: 451-454.

68. Liu, J.S., and R. Chen. 1998. Sequential Monte Carlo Methods for dynamic systems. Journal of the American Statistical Association 93: 1032-1044.

69. Lu, F., and E. Milios. 1997. Globally consistent range scan alignment for environment mapping. Autonomous Robots 4: 333-349.

70. Madhavan R. and H.F. Durrant-Whyte. 2005. 2D map-building and localization in outdoor environments. Journal of Robotics Systems 22 (1): 45-63.

71. Madhavan R., H.F. Durrant-Whyte. 2004. Natural landmark-based autonomous vehicle navigation. Robotics and Autonomous Systems 46: 7-95.

72. Margrit, B., and L. Gurvits. 1997. Mobile robot localization using landmarks. IEEE Transactions on Robotics and Automation 13(2): 251-263.

73. Martinelli, A., V. Nguyen, N. Tomatis, and R. Siegwart. 2007. A relative approach to SLAM based on shift and rotation invariants. Robotics and autonomous systems 55: 50-61.

74. Masson, F., J. Guivant, and E. Nebot. 2003. Robust navigation and mapping architecture for large environments. Journal of Robotic System 20 (10): 621-634.

75. Meng, Q, and R. Bischoff. 2003. I-based pose determination and errors measurement for a mobile robot with two steerable drive wheels", Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing Changsha, China: 605-610.

76. Montemerlo, M., S. Thrun, D. Koller, and B. Wegbreit. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. Eighteenth National Conference on Artificial Intelligence: 593-598.

77. Montemerlo, M., S. Thrun, D. Koller, and B. Wegbreit. 2007. FastSLAM 2.0. Springer Berlin / Heidelberg. Springer Tracts in Advanced Robotics.

78. Moreno, L., S. Garrido, and M.L. Munoz. 2006. Evolutionary filter for robust mobile robot global localization. Robotics and Autonomous Systems 54(7): 590-600.

79. Moreno, L., S. Garrido, and D. Blanco. 2007. Mobile robot global localization using an evolutionary MAP filter", Journal of Global Optimization 37(3): 381-403.

80. Neira, J., and J.D. Tardos. 2001. Data association in stochastic mapping using the joint compatibility test. IEEE Transactions on Robotics and Automation 17 (6): 890-897.

81. Nguyen, V., A. Harati, A. Martinelli and R. Siegwart. 2006. Orthogonal SLAM: A step toward lightweight indoor autonomous navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems Bejing: 5007-5012.

82. Nieto, J., J. Guivant, and E. Nebot. 2006. DenseSLAM: simultaneous localization and dense mapping. The International Journal of Robotics Research 25 (8): 711-744.

83. Ochieng, W.Y., and K Sauer. 2002. Urban road transport navigation: performance of the global positioning system after selective availability. Transportation Research Part C 10:171–187.

84. Ojeda, L.., and J. Borenstein. 2000. Experimental results with the KVH C-100 fluxgate compass in mobile robot. Proceedings of the IASTED International Conference Robotics and Applications, Honolulu, Hawaii, USA.

85. Oussalah, M., H. Maaref., and C. Barret. 1997. Positioning of a mobile robot with landmark based method. IEEE/RSJ International Conference on Intelligent Robots and Systems 2: 865-872.

86. Park, P.K., J.A. Elrod, and D.R. Kester. 1991. Applications of satellite remote sensing to marine pollution studies. Chemistry and Ecology 5:57-73.

87. Patel, N.R., B. Bhattacharjee, A.J. Mohammed, B. Tanupriya, and S.K. Saha. 2006. Remote sensing of regional yield assessment of wheat in Haryana, India. International Journal of Remote Sensing 19:4071-4090.

88. Pietroniro, A., and T.D. Prowse. 2002. Applications of remote sensing in hydrology. Hydrological Processes 16:1537-1541.

89. Pradalier, C., J. Hermosillo, C. Koike, C. Braillon, P. Bessiere, and C. Laugier. 2004. The CyCab: A car-like robot navigating autonomously and safely among pedestrians. Robotics and Automation 2: 1945-1950.

90. Qi, S. and H. Jian-Da. 2008. An adaptive UKF algorithm for the state and parameter estimation of a mobile robot. Acta Automatica Sinica 34 (1): 72-79.

91. Randall, S., S. Matthew, and P. Cheeseman. 1990. Estimating uncertain relationships in robotics. Autonomous Robot Vehicles**.** New York, NY, USA . Springer-Verlag New York, Inc. 167 – 193.

92. Ratner, D., and P. McKerrow. 2003. Navigating an outdoor robot along continuous landmarks with ultrasonic sensing. Robotics and Autonomous Systems 45: 73–82.

93. Richards, J. A., and X. Jia. 2006. *Remote sensing digital image analysis*. 4th edition, Springer-Verlag Berlin Heidelberg.

94. Sarup, J., M. Muthukumaran, N. Mathur , and V. Peshwa. 2006. Study on tectonics in relation to the seismic activity of Dalvat Area, Nasaik District, Maharashtra, India using remote sensing and GIS Techniques," International Journal of Remote Sensing 27:2371-2387.

95. Shin, D., and K. Lee. 2005. Use of remote sensing and geographical information systems to estimate green space surface temperature change as a result of urban expansion. Landscape Ecology Engineering 1: 169-176.

96. Spagnol, C., et al. 2005. Model based GPS/INS integration for high accuracy land vehicle applications. IEEE/ASME International Conference Advanced Intelligent Mechatronics Proceedings: 952 – 956.

97. Stramondo, S., C. Bignami, M. Chini, N. Pierdiccai and A. Tertulliani. 2006. Satellite radar and optical remote sensing for earthquake damage detection: results from different case studies. International Journal of Remote Sensing 20:4433-4447.

98. Suksakulchai, S., et al. 2000. Mobile robot localization using an electronic compass for corridor environment. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Nashville, Tennesse.

99. Sun Y, J.K Paik, A. Koschan, and M.A. Abidi. 2002. 3D reconstruction of indoor and outdoor scenes using a mobile range scanner. Proceedings International Conference Pattern Recognition, Canada, Quebec, Canada 3: 653-656.

100. Thrun, S., D. Fox, W. Burgard, and F. Dellaert. 2001. Robust Monte Carlo localization for mobile robots. Artificial Intelligence 128:99–141.

101. Thrun, S., Y. Liu, Z. Ghahramani, A.Y. Ng, and H. Durrant-Whyte. 2004. Simultaneous mapping and localization with sparse extended filters. The International Journal of Robotics Research, 23 (7-8): 693-716.

102. Triebel R, P. Pfaff, and W. Burgard Multi-level surface maps for outdoor terrain mapping and loop closing. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on Beijing 2006: 2276-2282.

103. Tsai, C.C., H.H. Lin and S.W. Lai. 2005. Multisensor 3D posture determination of a mobile robot using inertial and ultrasonic sensors. Journal of Intelligent and Robotic Systems 42: 317–335.

104. Tsai,C. C. 1998. A localization system of a mobile robot by fusing dead reckoning an ultrasonic measurement. IEEE Transactions on Instrumentation and Measurement. 47 (5): 1339-1404.

105. Ushimi, N., M. Yamamoto, and A. Mohri. 2003. Two wheels caster type odometer for omni-directional vehicles. Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan September 14-19 2003: 497-502.

106. Von der Hardt, H.J., D. Wolf and R. Husson. 1996. The dead reckoning localization system of the wheeled mobile robot ROMANE. Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensory Fusion And Integration For Intelligent Systems: 603-610.

107. Wang, Z., S. Huang, and G. Dissanayeke. 2007. D-SLAM: A decoupled solution to simultaneous localization and mapping. The International Journal of Robotics Research. 26 (2): 187-204.

108. Weber, J., L. Franken, K.W. Jörg, and E. Puttkamer. 2002. Reference scan matching for global self-localization. Robotics and Autonomous Systems 40: 99–110.

109. Weng, Q. and S. Yang. 2006. Urban air pollution patterns, land use, and thermal landscape: an examination of linkage using GIS," Environmental Monitoring and Assessment 117:463-489.

110. Wilkinson, G. G. 2005. Results and implications of a study of fifteen years of satellite image classification experiments. IEEE Transactions on Geosciences and Remote Sensing 43(3): 433-440.

111. Wulf O., K.O Arras, H.I Christensen, and B. Wagner. 2D mapping of cluttered indoor environments by means of 3D perception. Proceedings of IEEE International Conference on Robotic sand Automation, New Orleans, LA 2004: 4204-4209.

112. Wulf O., C. Brenneke, and B. Wagner. Colored 2D maps for robot navigation with 3D sensor data. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, 3(28): 2991 – 2996.

113. Yen, J. and R. Langari.1999. *Fuzzy logic : Intelligence, control and information,* Upper Saddle River, New Jersey : Prentice Hall, 1999, ch. 13-16.

114. Zalama, E., et al. 2002. Concurrent mapping and localization for mobile robots with segmented local maps. Proceedings of the IEEE/RSJ International Conference on Intelligence Robotics and Systems EPFL, Lausanne, Switzerland: 546-551

APPENDIX A

MONTE CARLO LOCALIZATION CODE: FILE EXPLANATIONS

MAIN FILE

_Trajectory : *Reads the odometer data*

_CreateMap : *Reads the satellite image*

_DataCompRead : *Reads the stored virtual laser scans*

_DataComp : *Generates the virtual laser scans*

_Localization : *Starts Localization*

_LOCALIZATION

Scales the odometer data considering the sacel of the map

Reads real polar measurements made by the mobile robot

Start a while loop in which the navigation of mobile robot simulated

MR_Xavier_ParticleFilter *: Starts the Monte Carlo Localization*

_PARTICLEFILTER

Sets the probabilistic motion models

_ParticleDistribution : *Computes the weight of particles*
: *P[z|x]*

_ProbabilisticMotion : *Do resampling*
: *Predict the next state*
: *P[x(t+1)|x(t),u(t+1)]*

_PARTICLEDISTRIBUTION

_Dervish_EXE : *Filters the raw laser scan*

_MapConsistentDistribution : *Adjust the weights of particles*
: *considering the map*

_StoredRange : *Selects the nearest grid point*
: *and read polar scan stored for*
: *that grid*

_Particle_Weight : *Computes the weight of particle*
: *P[z|x]*

_RANGEFINDER

Computes the polar scan for a particular state

_TRAJECTORY

Stores the odometer data

APPENDIX B

GAUSSIAN SUM FILTER CODE: FILE EXPLANATIONS

MAIN FILE

Reads real measurements

Reads odometer data

_Constrained_Population : *Spread Gaussian PDFs*

Adjust their meand and covariance matrix

Starts a FOR loop to simulate mobile robot

_MultipleEKF : *Runs each EKF independenly*

GSF_ActiveGF : *Selects the Gaussians with large weights*

_MULTIPLEEKF

Starts a loop in which eac EKF runs seperately

_EKF : *Runs EKF algorithm*
: *Predict the states*
: *Update states*
: *Computes raw weights*

_EKF

Run EKF algorithm

_StoredRange : *loads the stored laser scan data for*
: *nearest grid point*

_ExperimentalData : *loads the laser scan record*

APPENDIX C

PATH MATCHING CODE: FILE EXPLANATIONS

MAIN FILE

_loadPATH : *Load road network*

_loadODOMETERDATA : *Load odometer data*

_movelocalizeROBOT : *Start simulation*

_MOVELOCALIZEROBOT

_movePARTICLE : *Start path matching*

_MOVEPARTICLE

_formMOVETEMPLATE : *Constructs the path from*
: *odometer*

_matchMOVETEMPLATE : *Computes the weight of each*
: *particle P[z|x]*
: *Counts the number of hits*
: *between road network and past*
: *path*

_mapconsistentDISTRIBUTION : *Adjust the weight of particles*
: *considering the road network*

_posteriorPDF : *Do resampling*
: *Predict the next set of*
: *particles using P[x(t+1)|x(t),u(t+1)]*

APPENDIX D

DIFFERENTIAL EVOLUTION CODE: FILE EXPLANATIONS

MAIN FILE

Read polar scans

Read satellite image

_Trajectory : *Read odometer data*

_DataCompRead : *Read stored virtual laser scan*

Start a FOR loop to simualte mobile robot navigation

_EKF

_Global_Differential_Evolution : *Starts localization with Differential*
: *Evolution*

_PredictedMotionObservation : *Predicts the state for ith parameter*
: *vector using P[x(t+1)|x(t),u(t+1)]*

_StochasticMove : *Simulate P[x(t+1)|x(t),u(t+1)]*

_EliminateMember : *Eliminate parameter vector which are*
: *inconsistent with the map*

_GLOBAL_DIFFERENTIAL_EVOLUTION

_Select_Parameter_Vectors : *Select a parameter randomly*

_Dynamic_SubPopulation : *Select parameters which are*
: *members of the local ellipse*
: *centered at parameter vector*
: *being considered*

_Local_Differential_Evolution : *Apply differential evolution to*
: *the members of local ellipse*

_LOCAL_DIFFERENTIAL_EVOLUTION

_ExperimentalData : *Selects the real measurement*
: *for nth simulation step*

_StoredRange : *Stores the virtual scan for the*
: *ith parameter vector*
: *Note that virtual scans are*
: *stored before the simulation is*
: *started*

_DE_Determine_Best : *Determines the best parameter*
: *vector which is a member of*
: *local ellipse*

_Performance : *Computes the fitness of*
: *parameter vectors in terms* `of`
: *cost function*

APPENDIX E

CONSTRAINED VITERBI CODE: FILE EXPLANATIONS

MAIN FILE

Read satellite image

_CreateGrid : *Create fix grid points i.e. state space*

_Trajectory : *Load odometer data*

_DataCompRead : *Reads virtual polar scans atatched to grid points*

Read laser scanner measurements

_Initialize : *Initialize Viterbi algorithm*

Start FOR loop to simulate robot navigation

_Recursion : *Starts Viterbi recursion*

_Termination : *Terminate Viterbi and selects the best state at end time*

_Backtracking : *Trace backward starting from best node to starting*
: node. The optimal links through state space i.e optimal
: path

_RECURSION

_ExperimentalData : *Load experimental measurements for the nth*
: *simulation step*

Start FOR loop to move among states

_DynamicDomainXY : *Select the most probable states by considering motion*
: *model i.e : P[x(t+1)|x(t),u(t+1)]*

_CostXY : *Computes the cost move from state x to state y*

_BestLinkXY : *Select the optimal link to reach to y state*

_ActiveNodes : *Select a set of active nodes whose cost is optimal*

CIRRICULUM VITAE

PERSONAL INFORMATION:

Name	:	Can Ulaş Doğruer
Place of Birth	:	Ankara
Adress	:	Hacettepe University, Beytepe Campus Mechanical Engineering Department Ankara / Turkey
Telephone Number	:	090 312 297 62 08 090 533 259 85 42
e-mail	:	cdogruer@hacettepe.edu.tr cdogruer@gmail.com
Nationality	:	Turkish

EDUCATION:

2003 Sept.- PhD Programme at METU CGPA: 3.40/4.0	Mech. Eng. Dept.	Ankara
1996-1999 MSc Programme at METU CGPA: 3.56/4.0	Mech. Eng. Dept.	Ankara
1991-1996 BSc Programme at METU Honor Student CGPA: 3.30/4.0	Mech. Eng. Dept.	Ankara
1984-1991 High School Degree	Ankara Atatürk Anadolu Lisesi	Ankara

WORK EXPERIENCE:

2004 Jan.- Research Assistant	Hacettepe University	Ankara

2001 Sept.-2003 Dec. Production Engineer	Aselsan	Ankara
2001 March-2001 Sept. Production Engineer	Toyota	Adapazarı
1996 Aug.-1999 Aug. Research Assistant	METU	Ankara

PUBLICATIONS

Maliha R., C.U. Dogruer, and H.N. Ozguven. 2004. Nonlinear dynamic modeling of gear - shaft - disk - bearing systems using finite elements and describing functions. ASME Journal of Mechanical Design 126(3): 528-534.

Dogruer C.U., B. Koku, and M. Dolen. 2008. Global urban localization of outdoor mobile robots using satellite images. IEEE / RSJ International Conference on Intelligent Robots and Systems, Nice, France: 3927-3932.

Dogruer, C.U., B. Koku, and M. Dolen. 2007. A novel soft-computing technique to segment satellite images for mobile robot localization and navigation. IEEE / RSJ International Conference on Intelligent Robots and Systems, San Diego, California, USA: 2077-2082.

Kilic E., M. Dolen, B. Koku, and C.U. Dogruer. 2007. Novel Position Estimators Schemes for Timing Belt Drives. WESIC 2007 Warsaw, Poland.

Dogruer, C.U., E. Kilic, B. Koku, and Dolen M. 2007. Nonlinear position estimators based on artificial neural networks for low cost manufacturing systems. WESIC 2007, Warsaw, Poland.

Dogruer, C.U., and H.N. Ozguven. 2004. Dynamic analysis of spur gears on elastic bearings and shafts: effect of transverse shaft and bearing flexibility interacting with backlash. The Eleventh International Conference on Machine Design and Production, Antalya, Turkey.

Maliha, R., C.U. Dogruer, and H.N. Ozguven. 2003. Nonlinear dynamic modeling of gear-shaft-disk-bearing systems using finite elements and describing functions. Proceedings Of The 9th International Power Transmission And Gearing Conference, Chicago, Illinois 2003 - Later Published in a Journal In Revised Form.

Dogruer, C.U., B. Koku, and m. Dolen. 2008. Global urban localization of an outdoor mobile robot with genetic algorithms. Springer Tracts in Advanced Robotics, European Robotics Symposium 44: 103-112.

Submitted

Dogruer, C.U., B. Koku., and M. Dolen. 2008. Mapping and localization using satellite images", submitted to Robotica in December 2008.

Dogruer, C.U., B. Koku., and M. Dolen. 2009. Novel solutions for global urban localization. submitted to Robotics an Autonomous Systems in January 2009.

PROFESSSIONAL AFFILIATIONS:

Member of Union of Turkish Engineers and Architects, Chamber of Mechanical Engineers

COMPUTER SKILLS:

PRO/ENGINEER	Professional
MATLAB	Professional

LANGUAGES:

English	Fluent	
	KPDS	:90
	TOEFL	:253

INTERESTS AND HOBBIES:

Fitness, Traveling, History and Ancient Architecture

REFERENCES:

Assist. Prof. Dr. Buğra Koku	Mechanical Engineering Department
	Middle East Technical University
	+90 312 210 52 51
Assist. Prof. Dr. Melik Dölen	Mechanical Engineering Department
	Middle East Technical University
	+90 312 210 52 72