Occupancy Grid Mapping for Mobile Robot Using Scale Invariant Feature Transform

A Dissertation

Submitted in partial fulfillment of the

requirements for the award of the degree of

MASTER OF TECHNOLOGY

by

UTPAL KANT

DEPARTMENT OF INSTRUMENTATION AND CONTROL ENGINEERING

Dr. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR (PUNJAB), INDIA

June, 2011

ABSTRACT

In order to achieve full autonomy for a mobile robot, it is essential to sense the environment accurately. Sensor plays a fundamental role in the autonomy process of mobile robot. Various types of sensor such as sonar sensor, infrared sensor, laser sensor, vision sensor etc. are used for environment mapping. Each of the above mentioned sensors has its own drawbacks. The sonar range finders for instance suffer from wide beam cone, specular reflection, crosstalk etc. On the other hand lasers available in market are expensive and output of the sensor depends upon the surface brightness. The stereo vision systems are very sensitive to changes in illumination and the algorithms used for vision are computationally expensive. Hence it is clear that it is difficult for one sensor alone to give satisfactory information for mapping of mobile robot's environment. Researchers used many ways to improve sensory information such as by using sensor fusion techniques etc. The proposed works is focused upon improvement of vision accessibility in mobile robot. The first section deals with the selection of appropriate sensor for particular environment. The second section deals with optimization of threshold ratio used in Scale Invariant Feature Transform (SIFT). The occupancy grid mapping of environment is computed by using SIFT feature modelling and fusion of grid (using recursive Bays rule) with prior information is given in the last section of the dissertation.

ORGANIZATION OF DISSERTATION

The dissertation has been divided in to five chapters followed by a list of References. Short descriptions of various chapters have been given in the following paragraphs:

Chapter 1 This chapter gives the basic introduction about various sensors used for navigation in the mobile robot along with drawbacks of these sensors for which modelling is required. Some detail of literature survey is described. The objective and contribution is also presented. At the end of chapter application of sensor fusion in various fields is explained.

Chapter 2 This chapter deals with detail study of methods which is commonly used for the internal representation as well as sensors which is used for environment mapping. Finally sensor model by Moravec and Elfes and sensor model by Robin R. Murphy are presented.

Chapter 3 This chapter deals with sensor fusion and multisensor integration. Various type of fusion and detail study of various approaches for sensor fusion are presented. Recursive Bayes and map updating by using Bayesian approach have been discussed. Finally the fusion process for sensors is discussed.

Chapter 4 This chapter shows experimental setup and simulation results for distance measurement using stereo vision system, occupancy grid mapping using SIFT feature model and integrating the occupancy grid obtained from SIFT feature model by using Recursive Bays Rule.

Chapter 5 On the bases of the results obtained in the previous chapter’s conclusions have been discussed. Future scope of the project has also been projected at the end of this chapter.

TABLE OF CONTENTS **Page No**

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
BPA	Basic Probability Assignment
DoG	Difference of Gaussian Function
DS	Dempster-Shafer
RANSAC	RANdom SAmple Consensus
SDF	Sensor Data Fusion
SIFT	Scale Invariant Feature Transform
SVS	Stereo Vision System

Chapter 1
Introduction

Autonomy in mobile robots is a relatively new research area in the field of robotics which deals with the design and implementation of algorithms for control, artificial intelligence (AI), vision, mapmaking, sensor fusion, path planning, knowledge representation and reasoning. However, rather than being simply a collection of such algorithms, robotics is also the physical embodiment of the algorithms. One of the main challenges in robotics is that a mobile robot has to achieve autonomous navigation in unknown and uncertain environments, a task that is easy for humans and animals, but more difficult for robots. The reason for this is that it involves practically every aspect of AI in robotics: sensing, acting, planning, architectures, hardware, computational efficiency and problem solving.

Environments, the robot has to deal with such as indoor environments which are highly dynamic and where things are moving continuously. An abstraction of such environment can be stored in a world model which is usually defined as high level internal representations. Sensory information can be either added to a predefined model of the world or used to create it dynamically during the operation.

Sensing is the mechanism in which the mobile robot receives information about the environment. Sensors play a fundamental role in the autonomy process of mobile robots they help the mobile robot to respond to an input quantity by generating a functionally related output. Hence it is necessary to equip any robot with sensors, allowing it to sense the surroundings to detect obstacles; different modalities must be taken into consideration when added in the architecture of the mobile robot.

The fusion process is an important step when dealing with different readings from different sensors and by combining them on the other hand the mobile robot can achieve localization and obstacle avoidance. Since sensors are by nature inaccurate, uncertainty has to be taken into account when the mobile robot measures its environment. The result of the fusing process from different sensors can be used to construct a local as well as global robot’s world environment. The robot can also adapt by itself to unexpected situations in the environment.

Even though there have been advances in sensor data fusion paradigm there is still a necessity to investigate new approaches which can for instance improve aspects of the sensor fusion paradigm. This dissertation describes a new and more accurate approach to sensor data modelling and sensor data fusion to scene environment. The occupancy grid mapping is being generated by using stereo vision system. Scale Invariant Feature Transform (SIFT), the feature based correspondence algorithm is being used to find correspondence between features from left and right image of the stereo vision system for extracting the range information of the objects present in the scene. The approach consists of an enhanced sensor data fusion application for occupancy grid mapping based on experiments applied to sensor readings such as vision sensor.

1.1 Background and Motivation

To operate a mobile robot in highly unorganized environment, sensors play vital role to convert physical quantity into an electronic signal. The sensors which is used in mobile robotics are sonar, infrared, laser, vision etc. All of the sensors have their own advantages and disadvantages. Sonar sensor is most widely used sensor due to low cost, simple in operation and does not depend on brightness of the surface but it suffers from some serious problem as specular reflection and wide beam. Laser range finders are expensive and having problem of transparency in some cases. Vision and infrared depend upon illumination and brightness of the surface.

My motivation is to find a solution which will make a system more allegiant that can operate in both organized as well as unorganized environment and reform the cost of the system. In my work, the combinations of vision sensor and priory to the environments have been used. Vision sensor is although expensive but at the same time it is most powerful sensor if its full potential is utilized.

1.2 Literature Review

The literature related to occupancy grid modelling can be divided in three categories.

a) The first category is related to map building based grid representation. There are many works in the literature dealing with occupancy grid. In 1987, Elfes anticipate the occupancy grid method that is better formalized in his PhD thesis [3]. His work is implemented in two robots, Neptune and Terregator and is the part of the more complete system that integrates navigation and mapping based on sonar. This system is named Dolphin.

Moravec anticipate a system, in which the occupancy grid map is based on sonar data and stereo data [2].

Flynn has presented a method in which two different sensors (Sonar and Infrared) were combined to reduce uncertainty and to produce a refined map of the robots work space.

Konolige in 1997 has given a method to treat the problems relative to the sonar in a more efficient way. Example of problems is the specular reflection and redundancy of readings [10].

Thuran has introduced a forward sensor model to deal with the problem of dependence among cells presented by the standard algorithm of Elefs [4].

In 2006, Kong has implemented a localization system based on EKF (Extended Kalman Filter) in which features inside the environment, as corners and flat surfaces , are detected.

b) The second category belongs to sensor fusion of multiple sensor information by updating the map. Early work in map updating was starting in 1985 by Moravec and Elfes, where the fused operation performed on the occupied and empty areas of the sensor model are simply done by using probabilistic addition formula [1].

In 1987, Elfes and Mathies proposed the use of Bayesian estimation scheme to integrate and update the existing map with successive sensor readings.

Luo and Kay, in 1988, gathered and presented the issues concerning the effective integration of multiple sensors into the operation of intelligent system. In a latter publication, Luo and Kay, in 1989, presented a more complete survey of the variety of the approaches of multisensor integration and fusion [47].

In 1990, Luo and Kay, comes with a tutorial on multisensor integration, where multisensor integration is discussed in terms of basic integration functions, such as sensor model, sensor registration, world model etc [11].

Moshe Kam, in 1997, makes a review of sensor fusion techniques in robot navigation. These integration techniques falls into two categories: low level fusion which is used for direct integration of sensory data, resulting in parameter and state estimates, high level fusion is used for indirect integration of sensory data in hierarchical architectures [15].

Artificial Neural Network (ANNs) can be applied as a sensor fusion technique in mobile robots. ANNs learn complex relationship among the individual sensor data values. Application of ANNs in sensor fusion can be found in the literature. For instance, in 1993, Tharun utilizes a mobile robot called “COLUMBUS”, in which a neural network is trained. The inputs of the neural network are the sensor measurements and the outputs are the occupancy values of the cells. The mapping between the sensor measurements and the occupancy of the cells of the local robot’s environment can be learned.

Krose in his paper in 1996 proposes a neural network method to learn the probabilistic sonar sensor model. The conversion of the sensor data remains adaptive to change in either the sensor or its environment.

Moshe in 1997 presents a hierarchical neural network for a mobile robot control. The network receives input from the sensors and transmits on/off commands to the motors [15].

c) The third category deals with computer vision. The research in the field of computer vision for mobile robots can broadly be classified in two lines: vision-based navigation for indoor robots and vision- based navigation for outdoor robots [48]. Over the years, stereo vision has proved to be an economical sensor for obtaining three-dimensional range information which can be used to map building. Early approaches for indoor environments attempted to identify edges in images and infer the geometry. Some work has been done in gathering this information for building the robot’s map. [49] For instance uses the canny edge detector to extract edges (features) from the scene; the features are matched from the pair of images near to the horizon line to obtain range data. The result of this process is a set edge points on the horizon line where depth is known, and the range data uncertainty model is used to create an occupancy grid.

The work done by Rencken in [50] consists of localization of features by the mobile robot in the environment. Such features were used by the mobile robot to localize itself in such environment. Meanwhile, the robot was seeking for new features which could be used for the same purpose, i.e. localization of the robot in the environment.

Some research was focused on using stereo vision with other sensors for navigation. More precisely, [51] has used stereo vision together with odometry to instantiate a symbolic world model. The uncertainty in sensor data is represented by a multivariate normal distribution and the uncertainty is reduced by the use of Extended Kalman filter.

Research applications using vision can be found in the year 1997. [52] For instance, uses a trinocular stereo vision to produce high accurate depth images, which are mapped on an occupancy grid framework. In this approach the combination of potential field together with shortest path method is carried out, yielding good results.

The development of images matching by using a set of local interest points can be traced back to the work of Moravec [53] on stereo matching using corner detector. The Moravec detector was improved by Harris and Stephens [54]. The Harris corner detector is very sensitive to changes in images scale, so it does not provide a good basis for matching images of different sizes. David Lowe [55] extended the local feature approach to achieve scale invariance and came up with a new method for image feature generation called the Scale Invariance Feature Transform *(SIFT)*. The work also describes a new local descriptor that provides more distinctive features which are also invariant to image translation, scaling, rotation and partially to illumination changes and affine or 3D projection. This method was developed originally for object recognition in cluttered real-world scenes. [58] Uses the *SIFT* – features proposed in [55] to build up a 3D map of the robot. Later on, Lowe in [57] provides a more in-depth development and analysis of [55] and at the same time it presents improvements in stability and feature invariance.

Efficient code to implement SIFT is available as part of the open source VLFeat library [61]. A.Vedaldi in [62] provides a more efficient implementation of the SIFT detector and descriptor.

1.3 Objective and Contribution

The most important issue in the application of sensor fusion is the accurate conversion of sensor measurements in a spatial representation to which actual fusion method will be applied. The main objective of this thesis can broadly be classified in three steps.

a) Finding a proper combination of sensors.

b) Finding cost effective vision algorithm.

c) Finding strategies to accurately represent sensor information into an internal model.

d) Finding sensor fusion strategies for combining sensor information.

e) To represent the environment of the mobile robot in a global map by using fusion of vision sensor information. This work constitutes the results showing improved occupancy grid mapping by using stereo vision system.

The main contributions of the work done in this dissertation can be outlined in the following:

Reduction in sensor uncertainty is achieved by interpreting and combining sensor reading. The uncertainty in the incoming sonar data is interpreted using probabilistic sensor model (Moravec and Elfes model). Bayesian theory is used to integrate and update the sensor data reading over the occupancy grid. The data from stereo vision system is taken of the view in front of a mobile robot. Finally vision occupancy grid is being updated by priory information for mapping the environment of mobile robot where fusion process is carried out. This process shows the enhancement in occupancy grid mapping by stereo vision system.

1.4 Application of Sensor Fusion

Redundant and complementary sensor data can be fused and integrated using multisensor fusion techniques to enhance system capability and reliability. In recent years, benefits of multisensor fusion have motivated research in a variety of application areas as follows.

A. Robotics

Robots with multisensor integration and fusion capabilities enhance their flexibility and productivity in industrial applications such as material handling, part fabrication, inspection, and assembly [11 and 12]. Recent advances in robotics include multirobot Cooperative system, dexterous hands, under actuated and nonholonomic systems, interaction between the robot and the environment, teleportation, visual serving, etc. [13]. Mobile robots present one of the most important application areas for multisensor fusion and integration [14 and 15]. When operating in uncertain or unknown dynamic environments, integrating and fusing data from multiple sensors enable mobile robots to achieve quick perception for navigation and obstacle avoidance. Perception, position location, obstacle avoidance, vehicle control, path planning, and learning are necessary functions for an autonomous mobile robot. Luo and Kay [11] reviewed some of multisensor-based mobile robots including Hilare, Crowley's mobile robot, ground surveillance robot, Stanford mobile robot, CMU's autonomous land vehicles, and the DARPA autonomous land vehicle.

The body of the Honda humanoid robot is equipped with an inclination sensor that consists of three accelerometers and three angular rate sensors. Each foot and wrist is equipped with a six-axis force sensor and the robot head contains four video cameras [16]. Alfredo C. Plascencia [9] presented sensor fusion model for autonomous mobile robot navigation. Multi sensor fusion and integration of vision, tactile, thermal, range, laser radar, and forward looking infrared sensors plays a very important role for robotic systems.

B. Military Applications

Military applications of multisensor integration and fusion are in the area of intelligence analysis, situation assessment, force command and control, avionics, and electronic warfare. Radar, optical, and sonar sensors with various filtering techniques have been employed for tracking targets such as missiles, aircrafts, and submarines. Hall and Llinas [17] pointed out some defence-related applications such as ocean Surveillance, air-to-air and surface-to-air defence, battlefield intelligence, surveillance, target acquisition, and strategic warning and defence. Filippidis and Martin [18] used fusion of imagery from a multispectral camera and an infrared sensor to reduce false-alarm rate and improve the surface land-mine detection. Carson et al. [19] proposed fusion algorithms to fuse radar data and identification friend or foe (IFF) data. The overall system tracking and target identification can be improved significantly by fusing different types of sensors. S. Singh and D. Smith [20] studied the Approaches to Multi sensor Data Fusion in Target Tracking [20].

C. Remote Sensing

Remote Sensing Applications of remote sensing include monitoring climate, environment, water sources, soil and agriculture as well as discovering natural sources and fighting the import of illegal drugs [21]. Fusing or integrating the data from massive multispectral sensors and active radar sensors is necessary for extracting useful information from satellite or airborne imagery. Fuzzy logic and neural network based multisensor fusion techniques have been used for classification of remote sensed imagery. Solaiman [22] proposed a thematic class membership level between the data level and the decision level. Inspired by expert reasoning approach, the proposed fuzzy classifier is based on the multisensor data and contextual information associated with membership values. The class membership values can be updated by using the membership values assigned to the multisensor data and contextual information until predefined decision conditions are satisfied. The proposed Scheme was successfully applied to land cover classification using ERS-1/JERS-1 SAR Composites. Chiuderi [23] used a neural network approach for data fusion of land cover classification of remote sensed images on an agricultural area. By using supervised and unsupervised neural network, the optical-infrared data and microwave data were fused for land cover classification. Dempster–Shafer evidence theory was applied by Le Hegarat–Mascle to unsupervised classification in multisource remote sensing. Using different combinations of sensors or wavelengths, the proposed method can effectively identify the land cover types. Multisensor fusion of remote sensed data was also used for monitoring land environment [24], optical depth from satellite data [25], and algae blooms in the Baltic Sea [26]. Solaiman et al. proposed an information fusion method for multispectral image classification post processing. Fusion of the thematic map and the edge map provided a series of closed contours corresponding to individual fields and containing a unique class.

D. Equipment Monitoring and Diagnostics

Condition-based monitoring of complex equipment such as automated manufacturing systems, turbo machinery, and drive trains can improve safety and reliability as well as reduce the repair/maintenance costs. For example, monitoring of tool condition plays an important role for manufacturing systems to ensure quality and efficient production. Researchers have applied multisensor fusion techniques via an artificial neural network to fuse measurement data, such as force signal, acoustic emission, accelerometer data and power signal to predict tool wear [27–31]. Collected data from multiple sensors and machine parameters can be used to train the multi-layer neural network to identify the tool wear. Experimental results indicate that neural-network-based schemes can successfully fuse multisensor data for the complicated manufacturing system and improve the accuracy of identification of tool wear conditions. A fusion process for condition-based equipment maintenance using JDL model was investigated in [32].

E. Biomedical Applications

Multisensor fusion has been applied to critical care monitoring [27] and medical images. In a special topic section on biomedical data fusion, Hernandez et al. [33] used multisensor fusion techniques to enhance automatic cardiac rhythm monitoring by integrating electrocardiogram (ECG) and hemodynamic signals. Redundant and complementary information from the fusion process can improve the performance and robustness for the detection of cardiac events including the ventricular activity and the atria activity. Case-based data fusion methods were proposed by Azuaje et al. [34] to improve clinical decision support. Three different data fusion models were established for case-based decision support and reasoning. Evaluated results indicate that the proposed method can improve the fusion significantly at the retrieval level for heart disease risk assessment. Medical image fusion is one of the most important biomedical application areas for multisensor fusion. Solaiman et al. [35] studied the problem of detecting the oesophagus inner wall from ultrasound medical images. Fuzzy logic based fusion methods were used for feature extraction from the images. The proposed schemes were implemented on real medical images and the results show good quality detection.

F. Transportation Systems

Transportation systems, such as automatic train control systems, intelligent vehicle and highway systems, GSP-based vehicle navigation, and aircraft landing tracking systems, utilize multisensor fusion techniques to increase reliability, safety, and efficiency. Mirabadi and Schmid [36] discussed sensor fusion for train speed and position measurement using different combination of global positioning by satellite (GPS), inertia navigation systems (INS), tachometers, Doppler radar, etc. A Kalman filter based sensor architecture was proposed in [37] for fault detection and isolation in multisensor train navigation systems. Kobayashi et al. [38] investigated the problem of improving accurate positioning of vehicles by fusing measurement data from differential GPS, wheel speedometer, and optical fibre rate gyro via Kalman filtering. Robust vision sensing techniques for a multisensor transportation system were proposed by Smith et al. [39] to increase safety in a variety of traffic situations. Applications for vehicle tracking and pedestrian tracking were used to demonstrate the effectiveness of the proposed schemes. Korona and Kokar [40] used an extended Kalman filter and learning algorithm to integrate passive sensor data from a laser range finder (LRF) and an infrared camera (FLIR) for racking a landing aircraft.

G. Other Applications

Other application areas of multisensor fusion and integration include space, agricultural mechanization, drug interdiction, etc. The most important multisensor-based applications in space are the increasing use of autonomous systems for repair and maintenance of satellites, assembly of space structures and object sensing. Sato et al. [41] studied an automatic harvester that can operate in the rice field without human operator. The harvester was equipped with contact, revolution, level sensors and gyroscopes as well as actuators and on-board computer. Chong and Liggins [42] proposed a distributed architecture to fuse data such as radar, infrared, and database from different law enforcement agencies for drug detection, tracking, and interception.

1.5 Conclusions

This chapter gives the basic introduction of autonomy in mobile robot and environment mapping. Different types of sensors that are used in environment mapping and problems generated from different sensors and solutions of that problem are shown in brief. Previous work done related to occupancy grid mapping as well as sensor fusion are also presented. Objective and contribution of the work are also shown. Finally the applications of sensor fusion in robotics as well as in other field are presented.

Chapter 2

Occupancy Grid Mapping

2.1 Introduction

Occupancy grid mapping is an approach to map the environment of the mobile robot, representing areas as occupied, empty and unknown. Occupied area is that area where objects may be present, empty area is that area where there is no object present and unknown area is the area that is not covered by the sensor. In the map, occupied area is represented as hill, empty area is represented as valley and unknown area is represented as plane. In this approach whole environment of the robot is represented by a matrix in which each element represents a place in the environment that can be empty or occupied or can be unknown area. This representation is called as occupancy grid. Occupancy grid is the easiest and more accurate method for representing the environment of a robot. Other types of internal representation that are used in the mapping of robots environment are also presented. Internal representation is helpful to classify which area of environment is occupied and which area of environment is empty.

In mapping of robot’s environment, sensors play fundamental role. Different types of active and passive sensors are used for the purpose of mapping of robot’s environment. In this chapter different type of sensors are explained.

At the end of chapter two type of probability sensor (sonar) model given by different researcher are shown. One is given by Moravec and Elfes and another one is given by RR Murphy. This type of mapping is useful in estimating the confidence in form of probability to recognize which area is occupied and which area is empty.

2.2 Internal Representation

An important requirement for an efficient mobile robotic system is a correct spatial description of its underlying space that can be constructed from its own sensorial data. This description, a trustable map, makes possible a coherent interaction of the robot with its environment, so that it can perform its tasks efficiently and also can deal with unexpected situations like dynamical obstacles appropriately. The process of construction of this representation is generally named as mapping and the result is a map of environment. Two main approaches are used to represent an environment in a map: the topological and the metric.

2.2.1 Metric

In the metric approach, the geometry of the environment is defined in a more detailed way, presenting, accurately, the position of the objects inside of the environment, for instance walls, chairs and desks. In this approach elements are typically represented as geometric features or as evenly spaced grids. Now a days grid based Internal representations are commonly used.

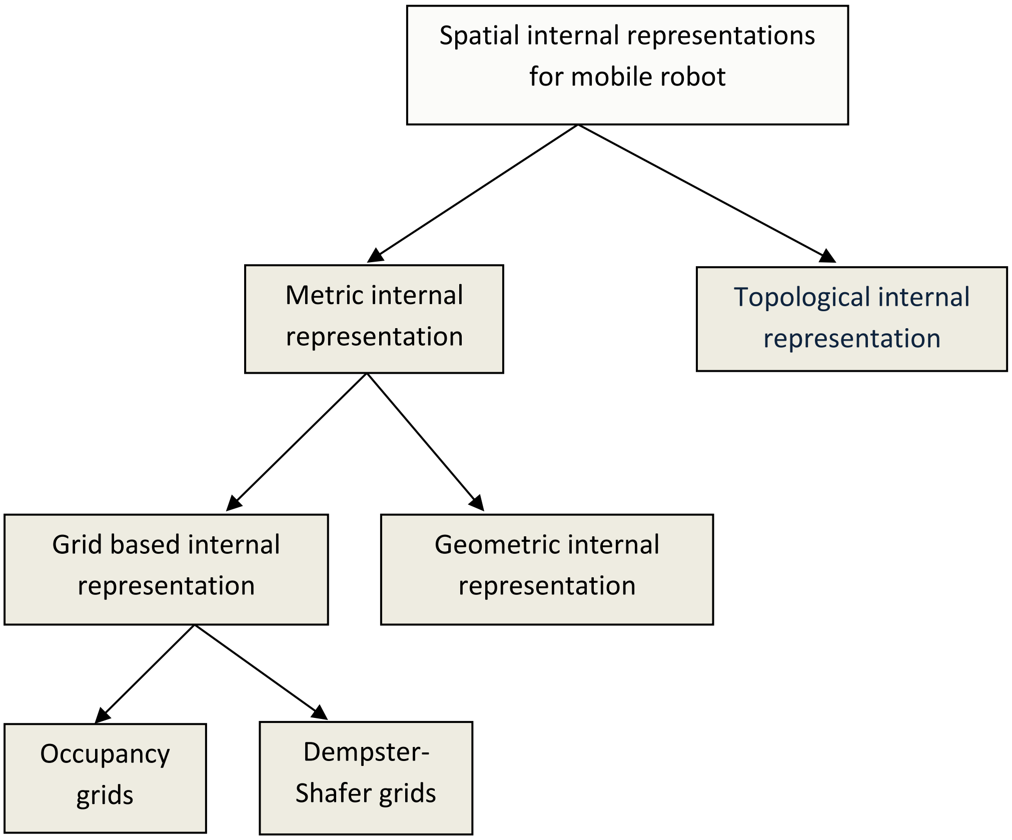

Spatial internal representations for mobile robot
Metric internal representation
Topological internal representation
Grid based internal representation
Geometric internal representation
Occupancy grids
Dempster-Shafer grids

Figure 2.1: Taxonomy of the two fundamental paradigm of internal representation

2.2.1.1 Geometric Representations

Geometric internal representations are mainly constructed by geometric discrete primitives such as: lines, polygons, points, polynomial functions, and so forth. A geometrical model needs only a few parameters to represent large region of space, i.e. numerical variables specify the position of these objects in a global coordinate system. This feature makes geometric maps highly memory efficient. Geometric maps in mobile robotics are composed of the union of simple geometric primitives: for instant, points, polygons and circles in 2D, or cubes and ellipsoids in 3D [9]. The representation of geometric map is shown below.

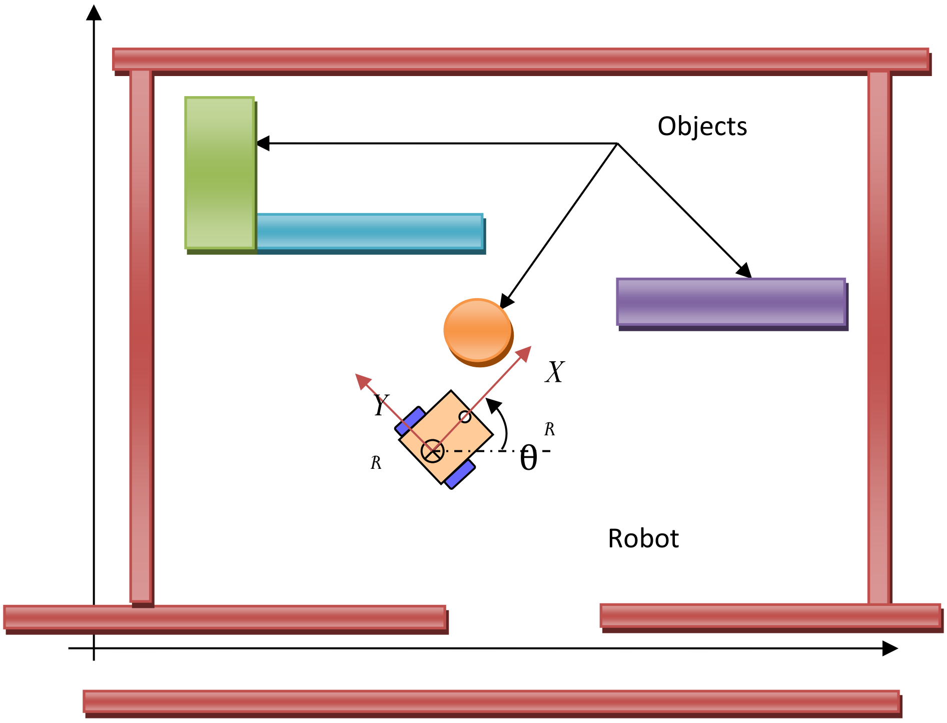


Figure 2.2 Geometric map

2.2.1.2 Grid Based Internal Representation

Two most widely used methods of evenly spaced grids are: occupancy grids and Dempster-Shafer evidential grids.

(A) Occupancy Grids

An occupancy grid is a grid-based approach, and it was proposed by Moravec-Elfes (1985) and Alberto Elfes (1987).

"The occupancy grid is a multidimensional random field that maintains stochastic estimates of the occupancy state of the cell in the spatial lattice". The Occupancy grid representation employs a multidimensional (typically 2D or 3D) tessellation of space into cells, where each cell stores a probabilistic estimate of its state [3 and 45]. Formally, an occupancy field *O*(x) is a discrete-state stochastic process defined over a set of continuous spatial coordinates $X= (x_1, x_2, \ldots\ldots x_n)$, while the occupancy grid is a lattice process, defined over a discrete spatial lattice. The state variable *s(C)* associated with a cell *C* of the occupancy grid is defined as a discrete random variable with two state, occupied and empty, denoted *OCC* and *EMP*. Consequently, the occupancy grid corresponds to discrete-state binary random field. Since the cell state are exclusive and exhaustive, *P[s(C) =OCC] + P[s(C) =EMP] = 1*.

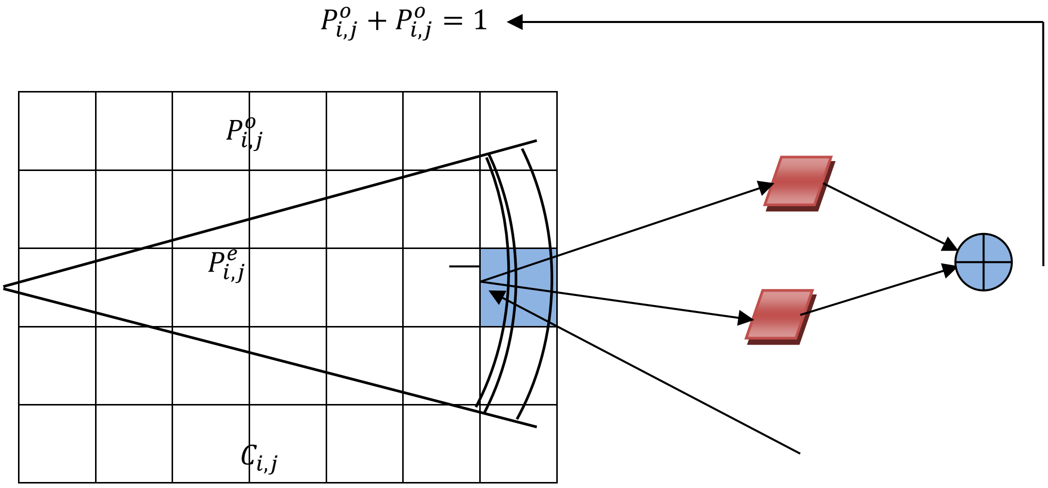


Figure 2.3 An Occupancy Grid is placed over a sensor model.

(B) Dempster-Shafer Grid

An alternative theory of evidence is Dempster-Shafer theory which produces results similar to Bayesian probabilities. It is a much newer theory, originating in the work of A.P. Dempster, a mathematician at Harvard, during 1960's with extensions by Glen Shafer in 1987. Whereas Bayes' rule relies on evidence being represented by probability functions, Dempster-Shafer theory represents evidence as a possibilistic belief function. Possibilistic means that the function represents partial evidence. For example, a reading may provide direct evidence for an event H, but, due to occlusions, not be perceived the entire object. Therefore, there is a possibility that the evidence could be higher than was reported. The possibilistic belief functions, also called Shafer belief functions, are combined used Dempster's rule of combination. The rule of combination is very different from Bayes' rule, although they provide similar results. Unlike Bayes' rule, Dempster's rule has a term which indicates when multiple observations disagree. This conflict metric can, be used by the robot to detect when its occupancy grid may be subject to errors [5].

2.2.2 Topological Representations

Topological representation of large scale spaces seems to be more close to human representation of the space. For instant, when providing directions to someone like: "Go down to the street, turn to left" and so forth. Topological approaches represent the robot's environment by a graph $G_p = (V, E)$, where V represents node in such graph, E represents arcs of edges that connect the nodes if there exists a path between them. In the former definition of topological internal representation, the nodes correspond to places or landmarks. A landmark is a perceptually distinctive feature of interest on an object of interest. Landmarks are needed for robot navigation: they tell the robot when a segment, arc or edge has ended and another will begin. Particular landmark are doorways and gateways. For example, an intersection of two hallways is a gateway: the robot can choose to go straight or turn to the left or to the right. Landmarks can be artificial or natural. Artificial landmarks are extra features added to an object to increase the recognition of the landmark. Natural landmarks are features that already exist on an object for recognition which were not designed for the perceptual activity.

2.3 Robotic Sensors

Sensors in the natural world include those which equip us with our five senses-sight, hearing, smell, taste, and touch. These convert the diverse inputs to electrochemical signals that can be used to inform or control the living organism.Sensors are used in mobile robot to provide useful information about the environment as well as the state of vehicle. A mobile robot uses a wide verity of sensors for varying purposes in order to achieve autonomy. The aim of this section is to provide an overview of the most common sensors used in mobile robots as well as their characteristics.

2.3.1 Robotic Sensor Classification

Systems of sensors are constructed for a variety of applications, including surveillance, imaging, mapping, and target tracking. In some cases, the sensor provides their own source of illumination and is referred to as active sensors. Passive sensors, on the other hand do not provide illumination and depend on variations of natural conditions for detection.

- Active sensors: This excitation signal is modified by the sensor to produce an output signal in the form of a beam of energy or a field. They are restricted to frequencies that can be generated and radiated fairly easily and therefore exclude part of the far infrared (IR) band (above 3×10^{12} Hz), part of the ultraviolet (UV) band, and the gamma ray region. However, inroads are being made into these regions with the development of terahertz sources and detectors based on artificial photonic crystal for examples: Laser range, Radar, Sonar or ultrasonic and infrared (IR) etc.
- Passive sensors: Passive sensors directly generate an electrical signal in response to a stimulus. That is, the input stimulus is converted by the sensor into output energy without the need for an additional source of power to illuminate the environment. The salient characteristics of most passive sensor are that they do not emit radiation (or an excitation signal) as part of the measurement process but rely on a locally generated or natural source of radiation (light from the sun), an available energy field (gravity) or even a chemical gradient. Passive sensor can exploit electromagnetic (EM) radiation of any frequency in which some natural phenomenon radiates and for which a detection mechanism exists. This can extend from extra-low frequency (ELF) (below 3×10^{3} Hz) up to gamma rays (above 3×10^{19} Hz). For examples: Inertial, Compass and Vision etc.

Both classifications contain internal and external sensors.

- Internal sensors. This type of sensors measures the internal state of the robot.
- External sensors. These sensors are used to determine the state of the mobile robot's environment.

Figure 2.4 shows the block diagram of different sensors.

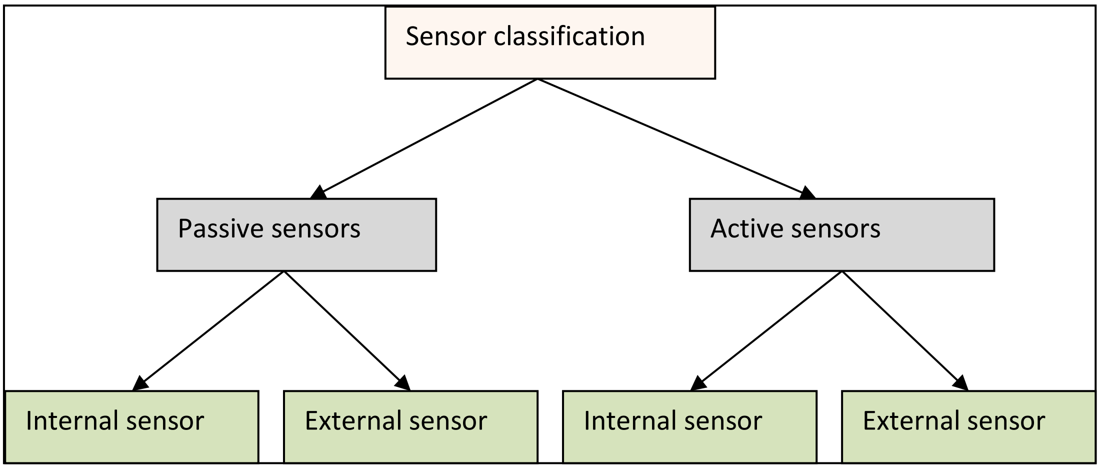


Figure 2.4 Sensor classifications

2.3.2 Active Sensors

2.3.2.1 Laser Range Finder

Laser stands for Light Amplification by Simulated Emission of Radiation. Some of the applications that are involved in the laser are: vehicle guidance, collision control and estimation of distance to objects. The distances which have been obtained by the laser can be used for obstacle detection and terrain mapping in 2D as well as 3D. A common method of laser range-finder is to obtain the distances based upon the time of flight principle, which consists of laser pulse in a narrow beam towards the objects and measuring the time taken by the pulse to be reflected off the target and returned to the sender. Some of the advantages of the laser are: it provides fast high- resolution readings over a long measurement range; it is able to produce 2D and 3D contours of the surrounding terrain; the insensitivity to illumination makes the laser eliminate the majority of noise and false readings associated with other active sensors. Some of the disadvantages or limitations of the laser range finder are: lasers are the most expansive sensor system devices available on the market; lasers can be transparent to some materials such a glass [9].

Figure 2.5 SICK Laser Sensors.

2.3.2.2 Radar

Radar stands for radio detecting and ranging. It is very much like sonar in its principle, i.e. it works under the principle of time of flight. Radar systems emit a microwave or millimeter-wave signal that is reflected off the objects. The time of flight can be used to compute distances to the objects. The frequencies of operation of the Radars are between 3 to 300 GHz and it is effective over both short and long distances making it suitable for outdoor applications. Radar provides better accuracy, resolution and noise performance than ultrasonic sensors. Radars can also provide information about structure because they can penetrate the surface layer of the objects. Radar is also used in tasks like robot mapping and navigation. Some of the drawbacks of a radar are spatial resolution and high cost [6].

2.3.2.3 Sonar or Ultrasonic

Sonar refers to any system for using sound to measure range. Sonar stands for Sound Navigation and Ranging. Sonars for different applications operate at different frequencies; for example, sonar for underwater vehicles would use a frequency appropriate for traveling through water, while a ground vehicle would use a frequency more suited for air. Ground vehicles commonly use sonars with an ultrasonic frequency just at the edge of human hearing. As a result the terms "sonar" and "ultrasonic" are used interchangeably when discussing extracting range from acoustic energy.

Ultrasonic is possibly the most common sensor on commercial robots operating indoors and on research robots. They are active sensors which emit a sound and measure the time it takes for the sound to bounce back. The time of flight (time from emission to bounce back) along with the speed of sound in that environment (remember, even air changes density with altitude) is sufficient to compute the range of the object.

Ultrasonic is common for several reasons. Its evolution paralleled the rise of the Reactive Paradigm: In the mid-1980's, Hans Moravec did impressive robot navigation with a ring of sonars [6]. The ring configuration gave 360^0 coverage as a polar plot. This ring was developed by one of the first mobile robot manufacturers, Denning Robotics, and since then sonar rings are often referred to as "Denning rings," regardless of manufacturer. Besides providing direct range measurements, the transducers were cheap, fast, and had terrific coverage. In the early 1980's, the Polaroid Land Corporation had developed small, inexpensive Sonars for use as camera range finders. A bigger version, the Polaroid Lab Grade ultrasonic transducer, can measure ranges from 1 to 25 feet with inch resolution over a field of view of 30^0. Furthermore, the measurement time was on the order of seconds versus hours for computer vision. Ultrasonic became the sensor of choice for behavior-based robots.

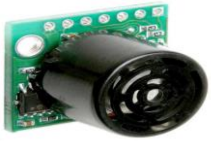


Figure 2.6 Ultrasonic transducer

A robotic sonar transducer is shown in Fig. 2.6 [43]. The transducer is about the size and thickness of a dollar coin, and consists of a thin metallic membrane. A very strong electrical pulse generates a waveform, causing the membrane on the transducer to produce a sound. The sound is audible as a faint clicking noise, like a crab opening and closing its pinchers. Meanwhile a timer is set, and the membrane becomes stationary. The reflected sound, or echo, vibrates the membrane which is amplified and then threshold on return signal strength; if too little sound was received, and then the sensor assumes the sound is noise and so ignores it. If the signal is strong enough to be valid, the timer is tripped, yielding the time of flight.

The measuring the distance between target and the sonar by the following formula, where the r is distance between target and the sonar's position, C is speed of the sound t_0 is the time of flight. For the sonar, 40 KHz sound transmitted at 343.3 m/s. figure 1.4 shows the basic principle of time of flight (TOF). Where, α is the angle between base of the sonar and the targets positions.

$$r = \frac{Ct_0}{2}$$

(2.1)

Where r is the distance between target and the sonar's positions, C is the speed of the sound and the t_0 is the time of flight of the sonar.

The key to how useful the data is requires understanding how the sound wave is generated by the transducer. In reality, the sound beam produces multiple secondary sound waves which interact over different regions of space around the transducer before dissipating.

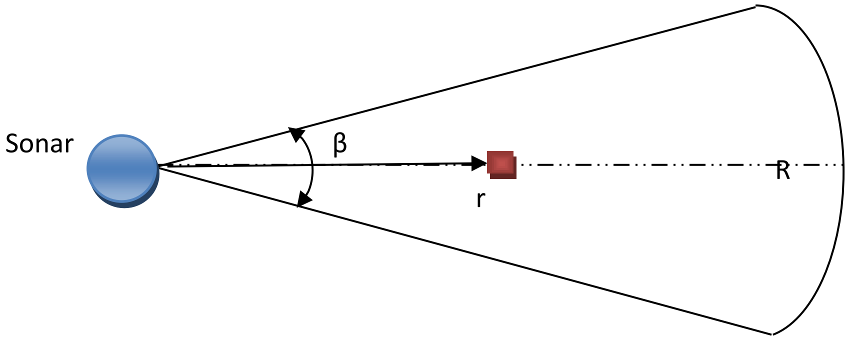


Figure 2.7 Time of flight principle of the Sonar

Secondary sound waves are called side lobes. Most robot systems assume that only sound from the main, or centermost, lobe is responsible for a range measurement. The width of the main lobe is often modeled as being 30^0 wide at about 5 meters away. However, in practice, reactive robots need to respond to obstacles in the 0.3 to 3 meter range. As a result many algorithms only treat the lobe as being between 8^0 and 15^0 wide depending on how reliable the range readings are in a particular environment.

The strength of the main lobe in the environment determines the maximum range that the sonar can extract reliability. In ideal indoor venues, sonar might return ranges of up to 25 feet, while in the outdoors; the same sonar might go no further than 8 feet with any repeatability. So while the upper limit of the range reading depends on the sensor and the environment, the lower limit does not. Ultrasonic transducers have a "dead time" immediately following emission while the membrane vibration decays. The decay time translates into an inability to sense objects within 11 inches because measurements made during this period are unreliable because the membrane may not have stopped ringing.

Regardless of the maximum allowed range return and the width of the lobe, most computer programs divides the area covered by sonar into the three regions shown in Fig. 2.8.

- **Region I** is the region associated with the range reading. It is an arc, because the object that returned the sound could be anywhere in the beam. The arc has a width, because there are some resolution and measurement errors; the width of Region I is the tolerance.

- **Region II** is the area that is empty. If that area was not empty, the range reading would have been shorter

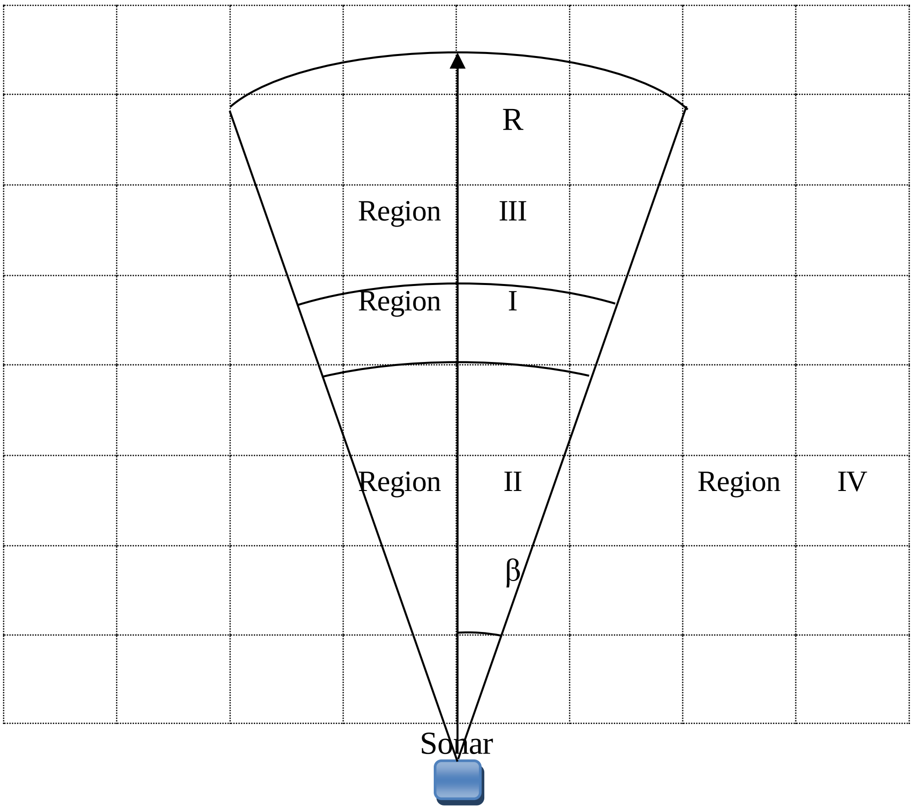


Figure 2.8 The regions of space observed by an ultrasonic sensor

- **Region III** is the area that is theoretically covered by the sonar beam, but is unknown whether it is occupied or empty because it is in the shadow of whatever was in Region I.
- **Region IV** is outside of the beam and not of interest.

Although they are inexpensive, fast, and have a large operating range, ultrasonic sensors have many shortcomings and limitations which a designer should be aware of. Ultrasonic sensors rely on reflection, and so are susceptible to specular reflection. Specular reflection is when the wave form hits a surface at an acute angle and the wave bounces away from the transducer. Ideally all objects would have a flat surface perpendicular to the transducer, but of course, this rarely happens. To make matters worse, the reflected signal may bounce off of a second object, and so on, until by coincidence return some energy back to the transducer. In that case, the time of flight will not correspond to the true relative range.

Even with severely acute angles, the surface is usually rough enough to send some amount of sound energy back. An exception to this is glass, which is very common in hospitals and offices where mail robots operate, but induces serious specular reflection. Fortunately this energy is often sufficiently strong to pass the thresholding in the transducer circuit. However, a new problem, foreshortenings, may occur. Recall that sonar has a 30^0 field of view. This means that sound is being broadcast in a 30° wide cone. If the surface is not perpendicular to the transducer, one side of the cone will reach the object first and return a range first. Most software assumes the reading is along the axis of the sound wave. If it uses the reading (which is really the reading for 15^0), the robot will respond to erroneous data. There is no solution to this problem [6, 9].

(A) Features

- Continuously variable gain for beam control and side lobe suppression
- Object detection includes zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- Serial, 0 to Vcc
- 9600 Baud, 81N
- Analog, (Vcc/512) / inch
- Pulse width, (147uS/inch)
- Learns ring down pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor operates at 42 KHz
- High output square wave sensor drive (double Vcc)

(B) Benefits

- Very low cost sonar ranger
- Reliable and stable range data
- Sensor dead zone virtually gone
- Lowest power ranger
- Quality beam characteristics
- Mounting holes provided on the circuit board

- Very low power ranger, excellent for multiple sensor or battery based systems
- Can be triggered externally or internally
- Sensor reports the range reading directly, frees up user processor
- Fast measurement cycle
- User can choose any of the three sensor outputs

(C) Drawbacks of the Sonar Sensor

(a) Specular Reflection and crosstalk

Specular reflection is not only by itself a significant source of erroneous readings; it can introduce a new type of error in rings of sonars. Consider a ring of multiple sonars. Suppose the sonars fire (emit a sound) at about the same time. Even though they are each covering a different region around the robot, some secularly reflected sound from sonar might wind up getting received by completely different sonar. The receiving sonar is unable to tell the difference between sound generated by itself or by its peers. This source of wrong reading is called cross-talk, because the sound waves are getting crossed. Most robot systems stagger the firing of the sonars in a fixed pattern of four sonars, one from each quadrant of the ring) at a time. This helps some with cross-talk, but is not a complete or reliable solution. If the sonar sound frequency and firing rate can be changed (which is generally not the case), then sophisticated abasing techniques can be applied.

The impact of the problems of specular reflection and cross talk become easier to see with plots of sonar returns overlaid with the perimeter of the area they were taken in; see Figure 2.9. Some walls are invisible, others are too close. Clearly sonar readings have to be taken with a grain of salt.

(b) Foreshortening

The 30^0 cone also creates resolution problems. While sonars often excellent resolution in depth, they can only achieve that at large distances if the object is big enough to send back a significant portion of the sound wave. The further away from the robot, the larger the object has to be. Most chairs and table tops present almost no surface area to the sensor and so the robot will often not perceive their presence and run right into them.

(c) Wide Beam

This refers when the uncertainty in sonar measurement is bigger perpendicular to the main axis of beam then along the axis i.e. the object can be situated any place along the occupied region. If there will be narrow space in front of sonar, it cannot detect that space due to wide beam.

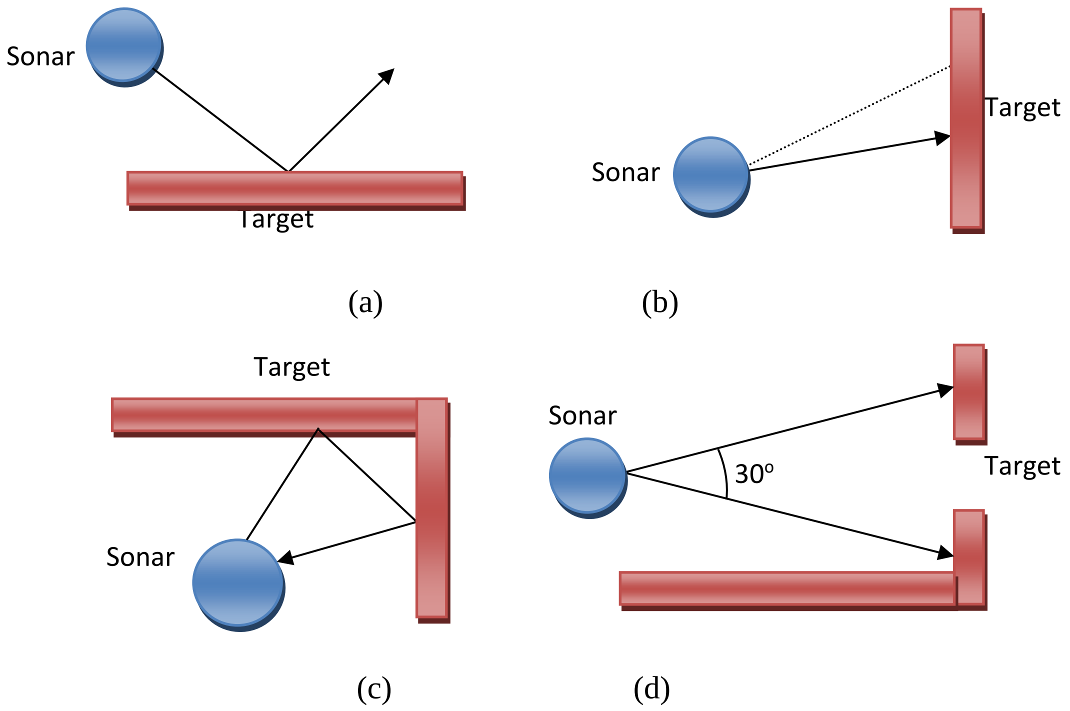


Figure 2.9 (a) Specular Reflection (b) Foreshortening (c) Crosstalk (d) Wide beam

(d) Power loss

In practice, another problem leads to spurious sonar readings: power. The generation of a sound wave requires a significant pulse of energy. If the robot is operating at low power levels, the correct waveform will not be generated and the return signal worthless. This problem is often difficult to debug by looking at the sonar returns, which often suggest specular reflection or crosstalk. One team in a robot competition attached a voltmeter next to the emergency stop button on their robot to ensure that this problem would not go undetected.

One method of eliminating spurious readings, regardless of cause, is to take the average of three readings (current plus the last two) from each sensor. This method is fairly common on purely reactive robots, but other approaches treat the reading as being uncertain and apply formal evidential reasoning techniques to refine the reading. These uncertainty techniques are employed by architectures operating under the Hybrid Paradigm.

2.3.2.4 Infrared (IR)

Infrared sensors are another type of active proximity sensor. They emit near infrared energy and measure whether any significant amount of the IR light is returned. IR sensors have a range of inches to several feet, depending on what frequency of light is used and the sensitivity of the receiver. The simplest IR proximity sensors can be constructed from LEDs, which emit light into the environment and have a range of 3-5 inches. These often fail in practice because the light emitted is often "washed out" by bright ambient lighting or is absorbed by dark materials (i.e., the environment has too much noise).

In more sophisticated IR sensors, different IR bands can be se modulated to change the signal-to-noise ratio. This typically ensures that an object in range doesn't absorb the light, causing the sensor to presence of the object. Nomad robots have an IR sensor option.

2.3.3 Passive Sensors

2.3.3.1 Inertial

The types of sensors which fall under the class of inertial sensors are: accelerometer and gyroscope. These kinds of sensors measure the derivatives of the robot’s position variable. Accelerometers measure the linear acceleration based on the spring mass system. Gyroscopes measure the angular acceleration of the vehicle and they can also provide information about the position. There are two types of gyroscopes; the most familiar one is the mechanical gyroscope, and an alternative approach is the fiber optic gyroscope.

2.3.3.2 Compass

The magnetic compass uses the earth’s magnetic field to orientate itself with respect to it. By doing the former, the orientation of the mobile robot can be found with respect to the earth’s magnetic field. The mechanic magnetic compass is most common and oldest in the market, it uses a magnet to rotate in the horizontal plane and align itself with respect to the earth’s magnetic fields.

2.3.3.3 Vision

The vision system is perhaps the highest potential sensor used in mobile robots. But it is probably also the most difficult sensor to master. A common camera configuration used in mobile robots is stereo-couple or stereo vision system. It consists of a pair of USB CMOS cameras. Cameras mainly operate by receiving the incoming light which enters the camera’s lens and hits the image sensor which is the heart of the digital camera. CCD and CMOS are two images sensors. The CCD (Charge Couple Device) sensor consists of numerous light-sensitive semiconductor elements called photo sensors. These photo sensors can be seen as a tiny rectangular blocks or pixels (an acronym for picture element). When the incoming photons reach the semiconductor materials, electrons are produced. The numbers of electrons are proportional to the light which reaches the light sensitive part of the sensors. These electrons are stored in a capacitor, which is connected to a MOS transistor acting as a light switch. These electrons contain certain voltage, which is called the video signal. The electrons have to be transported via shift registers to a frame grabber, where all the pixels are digitized i.e. the pixels are converted to integer values and arranged in a $M \times N$ matrix and stored in a memory buffer. The CMOS (Complementary Metal Oxide semiconductor) sensors are, in principle, photo sensitive diodes, which are in series with a resistor. The principle of the CMOS camera allows a continuous transformation of the incoming photons into a resulting voltage. The exact physical process is not so relevant because it is not the main topic of this thesis. Once an image is captured either by a CCD or a CMOS camera it has to be processed in order to be useful to the system, in this case, a mobile robot. There exist many digital image processing algorithms in the literature; which algorithm should be used? The answer of this question depends on the task the mobile robot must fulfill. Even though there has been a lot of research in vision algorithms in the last decades, the algorithms that exist now suffer from being computational expensive; which limits the real time applications. They also suffer from generalization; they can work very well in certain conditions but probably they do not work if light conditions are changed. It is believed to be the most powerful sensor if its full potential is utilized.

(A) Stereo Vision System (SVS)

Stereo vision system has to deal with inferring information from two or more images taken from the scene. This information can be the distance to the objects or a 3D structure of the environment. Two problems in computer vision that must be solved by a stereo system's algorithms are correspondence and reconstruction. Figure 2.10 shows a top view of a simple stereo system model, which is composed of two pin hole camera models. This model is simple because the two cameras are pointing in the same direction, making the geometry simple.

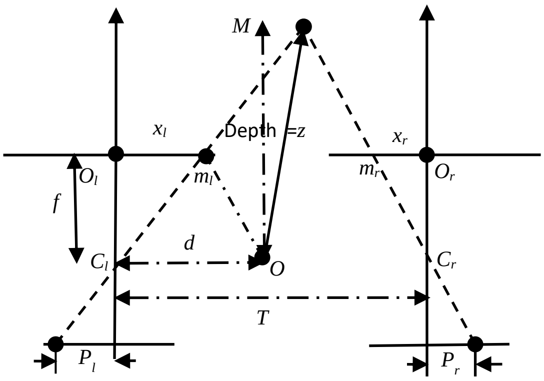


Fig. 2.10 A simple model of a stereo system

The distance z between the point M the baseline T can be computed from the similar triangles (m_l , M, m_r) and (C_l , M, C_r). Equations (2.2) and (2.3) show the procedure.

$$\frac{T+x_l-x_r}{z-f} = \frac{T}{z}$$

(2.2)

$$z = f\frac{T}{d}$$

(2.3)

Where x_l and x_r are the coordinates of m_l and m_r with respect to the principal points O_l and O_r respectively, and d=x_r-x_l is the disparity which measures the difference between the projected points in the two images. This method is usually used to get the distance information OM with equation (2.4).

$$D = \frac{T}{2(P_l+P_r)}\sqrt{(P_l - P_r)^2 + 4f^2} \qquad (2.4)$$

(B) Camera Parameters

The camera parameters have to do with the relation between the camera reference frame and the world reference frame and the relation between pixel coordinates in the image plane and coordinates of the image points in the camera reference frame. The method for estimating the camera parameters is called camera calibration. The cameras were calibrated using the camera calibration tool for MATLAB as suggested by [59].

The extrinsic and intrinsic parameters are the underlying parameters in the camera models. The extrinsic parameters are the ones that uniquely identify a transformation between the camera reference frame and a world reference frame. The extrinsic parameters are:

- A 3D translation vector T, this vector relates the coordinates of the two origins of the two reference frames.
- A 3X3 orthogonal translation matrix$(R^T R = RR^T = I)$. This matrix brings the corresponding axis of both frames onto each other.

The extrinsic parameters can be seen in figure 2.11. The intrinsic parameters are the ones that link the coordinates of a point in an image frame with the corresponding ones in the camera reference frame. Therefore the set of intrinsic parameters is:

- f,focal length,
- (O_x, O_y), the coordinates in pixel of the image center (the principal point),
- (S_x, S_y), the size of the pixel in mm,
- k_g, the geometric distortion introduced by the optics;

The extrinsic parameters can be depicted as in figure 2.11. The equations that make the transformation between camera and image frame coordinates can be seen in equation 2.5.

$$\begin{aligned} x &= (x_m - o_x)S_x \\ y &= (y_m - o_y)S_y \end{aligned} \qquad (2.5)$$

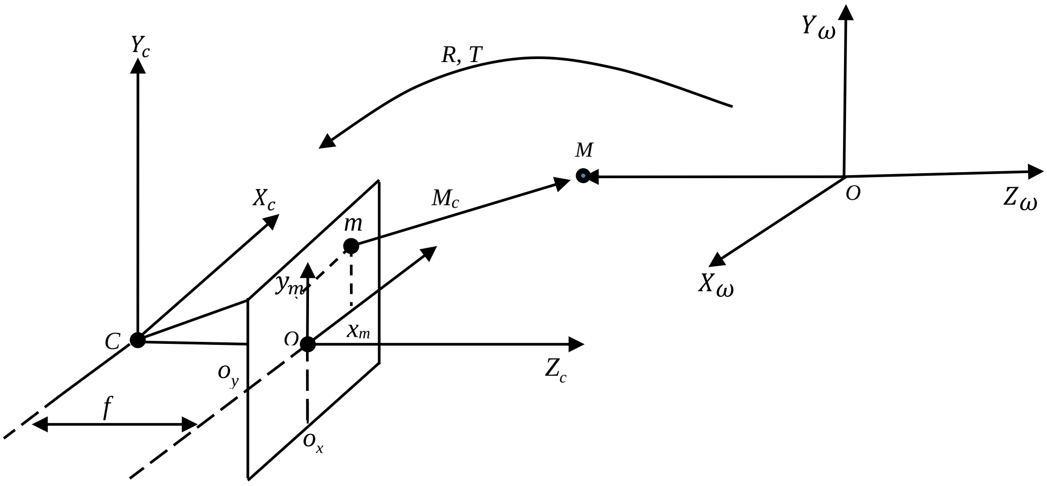


Figure 2.11: Figure showing the extrinsic and intrinsic parameters.

(C) Epipolar Geometry

Epipolar geometry is the geometry of stereo; it is essentially the geometry of the intersection of the image planes with the pencil of planes having the base line as an axis. Figure 2.12 shows the geometric entities involved in epipolar geometry. A pair of cameras and a point *M* in *3D* defines the following. An epipole is the point where the base line intersects the image planes. An epipolar planes is the plane identified by M, C_l, C_r and contains the baseline. An epipolar line is the line formed by the intersection between the epipolar plane and the image plane.

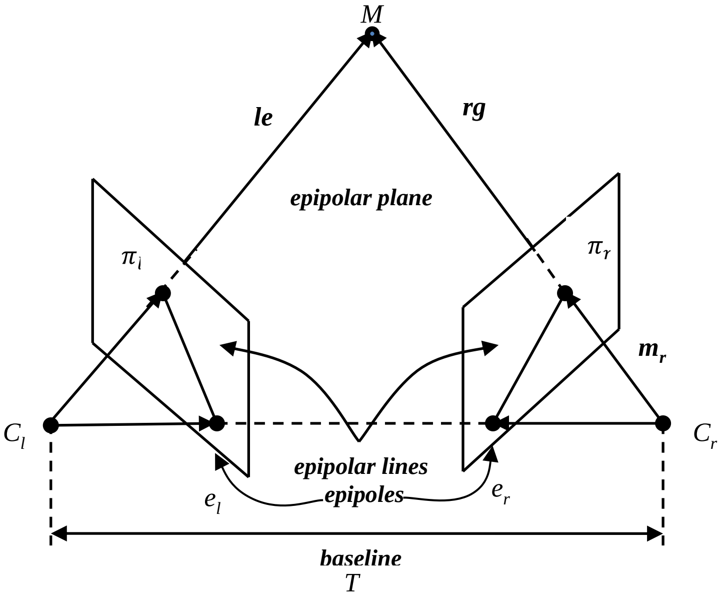


Figure 2.12: Epipolar geometry.

(D) Reconstruction by triangulation

Triangulation is a method of 3D reconstruction and it depends on knowledge of the parameters of the stereo system being available i.e. the intrinsic and extrinsic parameters. Reconstruction is straightforward if the intrinsic and extrinsic parameters are known in advance [60]. Figure 2.13 shows the basic triangulation schematic.

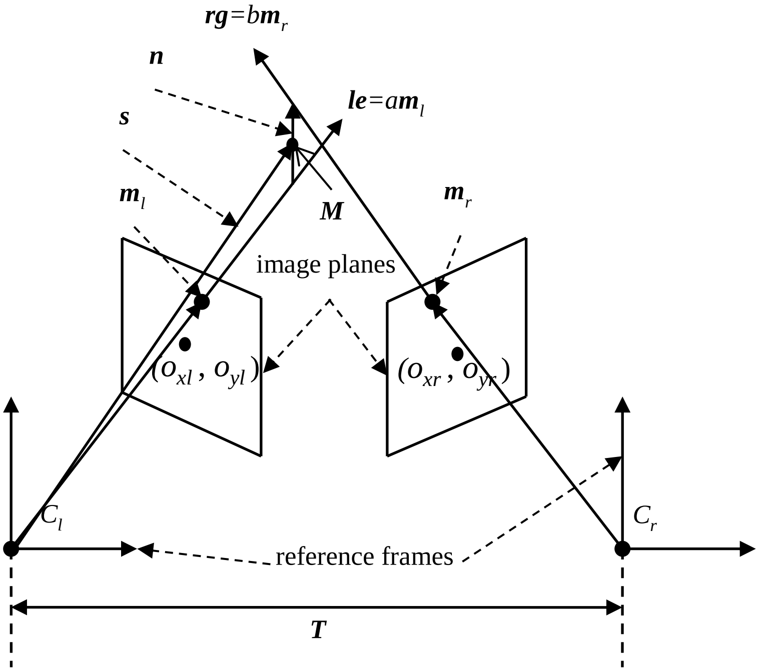


Figure 2.13: Triangulation.

Where the following is defined:

- M is a point in a *3D* space,
- m_l and m_r are two vectors whose coordinates are the projections of the point M onto the image planes.
- $le = am_l$ and $rg = bm_r$ $(a \in R, b \in R)$ are rays which go from the origins C_l and C_r of the right and left reference frames and pass through m_r and m_l respectively.
- rg has the following expression $rg = T + bR^T m_r$when is expressed in the left reference frame.
- n is an orthogonal vector to both le and rg and it is defined as $m_l X R^T m_r$.
- $s_M = am_l + \frac{1}{2}cn$ $(c \in R)$ is a vector which goes from C_l to the middle point of the vector n.

The two rays le and rg do not intersect each other at the location where the *3D* point is located because the parameters and image locations are only known approximately due to errors in the calibration process. The problem of getting the *3D* set of reconstructed points reduces to determining the location of the point *M* which is situated in the middle of the segment parallel to the vector n that joins le and rg. In the other words, the components of the vector $\boldsymbol{s}$ give the location of the point *M* as seen in equation 2.6.

$$s_M = am_l + \frac{1}{2}cn \qquad (2.6)$$

In order to compute the vector, the parameters a and c are needed; these parameters are found by solving the linear system which is derived as follows. The system of vectors shown in figure 2.14 is taken from figure 2.13.

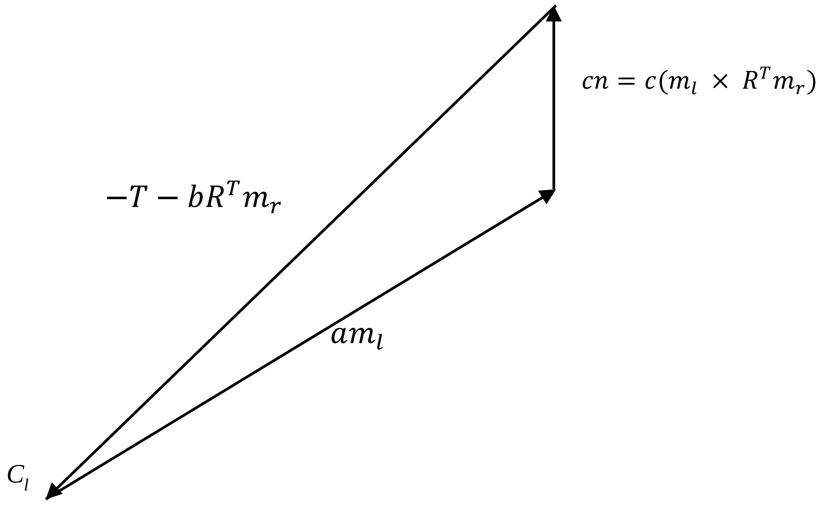


Figure 2.14: System of vectors

Equation 2.7 is derived from figure 2.14.

$$am_l + cn - T - bR^T m_r = 0$$

$$am_l + cn - bR^T m_r = T$$

$$am_l - bR^T m_r + c(m_l \, X \, R^T m_r) = T$$

(2.7)

Equation 2.7 is a linear system of the form $Ax = bb$ which can be arrange as follows.

$$[m_l \; R^T m_r \;\; m_r + c(m_l X \; R^T m_r)] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = T$$

Where

$$A = [m_l\ R^T m_r\ \ m_r + c(m_l X\ R^T m_r)] \tag{2.8}$$

$$x = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{2.9}$$

$$bb = T \tag{2.10}$$

2.4 Sensor Model

All methods of updating uncertainty require a sensor model. Models of sensor uncertainty can be generated in a number of ways. Empirical methods for generating a sensor model focus on testing the sensor and collecting data as to the correctness of the result. The frequency of a correct reading leads to a belief in an observation, the set of beliefs from all possible observations from the model.

Analytical methods generate the sensor model directly from an understanding of the physical properties of the device. Subjective methods rely on a designer's experience, which are often an unconscious expression of empirical testing.

One robotic sensor which has been heavily studied is ultrasonic transducer, or sonar. This chapter will use Polaroid sonar’s as an example; however, the principles of scoring and fusing belief apply to any sensor. Most robotic researchers have converged on a model of sonar uncertainty which looks like Figure 2.15.

2.4.1 Sonar Model by RR Murphy

The basic model of a double sonar beam has a field of view specified by β, the half angle representing the width of the cone, and R, the maximum range it can detect. This field of view can be projected onto a regular grid. The grid will be called an occupancy grid, because each element l (for element) in the grid will be hold a value representing whether the location in space is occupied or empty. As shown in figure 2.15 the field of view can be divided into three regions:

- **Region I:** where the affected elements are probably occupied.
- **Region II:** where the affected elements are probably empty
- **Region III:** where the condition of the affected elements is unknown.

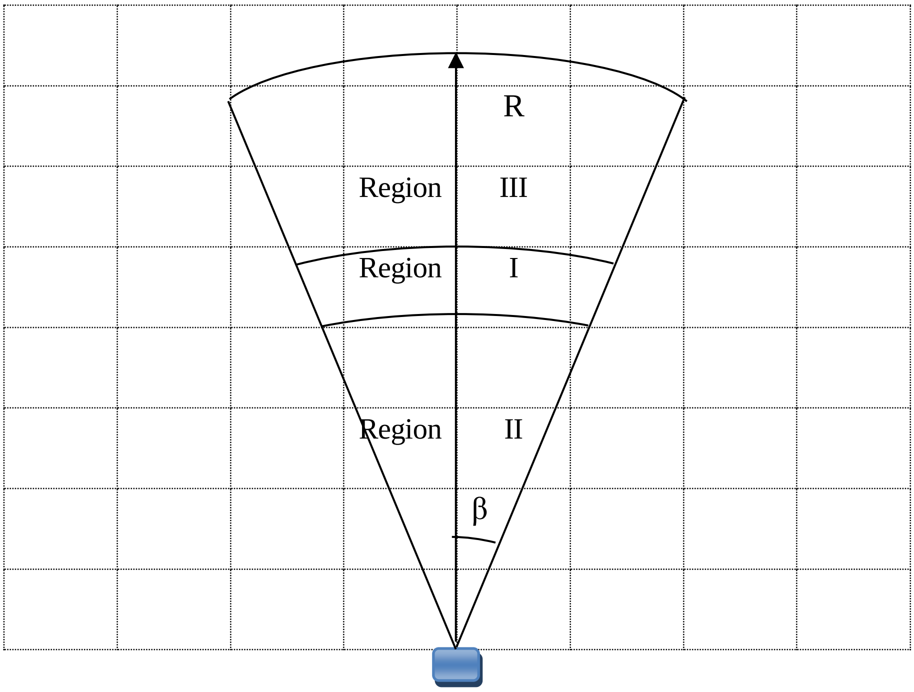


Figure 2.15 Sonar sensor model

Given a range reading, Region II is more likely to be really empty than Region I is to be really occupied. Regardless of empty or occupied, the readings are more likely to be correct along the acoustic axis than towards the edges. Recall that this is in part because an obstacle which was only along one edge would be likely to reflect the beam specularly or generate other range errors.

For every grid element falling into Region I:

$$P(Occupied) = \frac{\left(\frac{R-r}{R}\right)+\left(\frac{\beta-\alpha}{\beta}\right)}{2} \times Max_{occupied} \quad (2.11)$$

$$P\ (Empty) = 1.0 - P(Occupied) \quad (2.12)$$

Where r and α is the distance and angle to the grid element, respectively. The term in Eqn. 2.11 captures the idea that the closer the grid element is to the acoustic axis, the higher the belief. Likewise, the nearer the grid element is to the sonar beam, the higher the belief, the term $\frac{R-r}{R}$. The Max$_{occupied}$ term expresses the assumption that a reading of occupied is never fully believable. A Max$_{occupied}$ = 0.98 means that a grid element can never have a probability of being occupied greater than 0.98. It is important to note that Region I in Fig. 2.15 has a finite thickness. Due to the resolution of the sonar, a range reading of 0.87 feet might actually be between 0.82 and 0.92 meters, or 0.87±0.05 feet. The ±0.05 is often called a tolerance. It has the impact of making Region I wider, thereby covering more grid elements.

Each grid element in Region II should be updated using these equations:

$$P\ (Occupied) = 1.0 - P(Empty) \tag{2.13}$$

$$P(Empty) = \frac{\left(\frac{R-r}{R}\right)+\left(\frac{\beta-\alpha}{\beta}\right)}{2} \tag{2.14}$$

Note that unlike an element in Region I, an element in Region II can probability of being empty of 1.0.

To see how these formulas would be applied, consider the example Fig. 2.16. The sonar has returned a range reading of 3.0 feet with a tolerance of ±0.05 feet. The $\text{Max}_{\text{occupied}}$ value is 0.98. The robot is shown or and all elements are measured relative to it. The element of interest grid*[i][j]* is shown in black, and is at a distance r = 2 feet and an angle of $\alpha = 0^0$ from the robot. The first step is to determine which regions cover the element.

Scale: 1cell = 0.5 feet

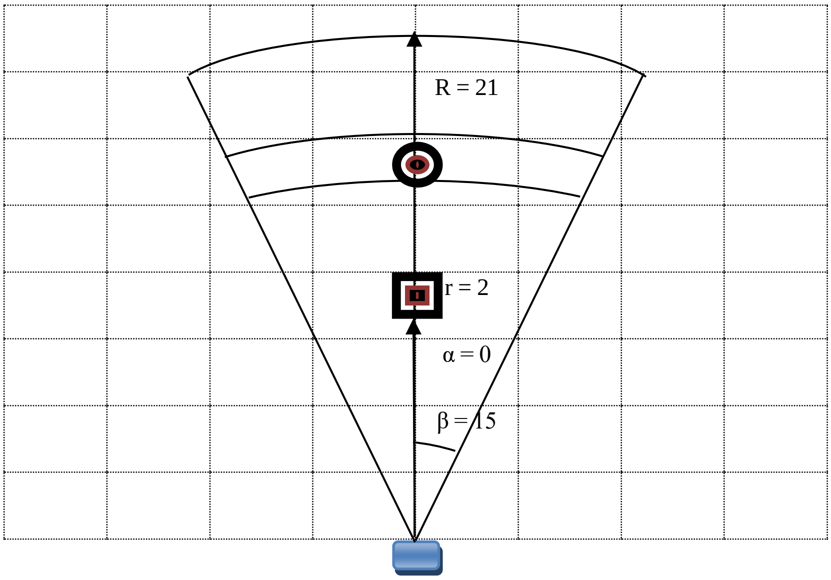


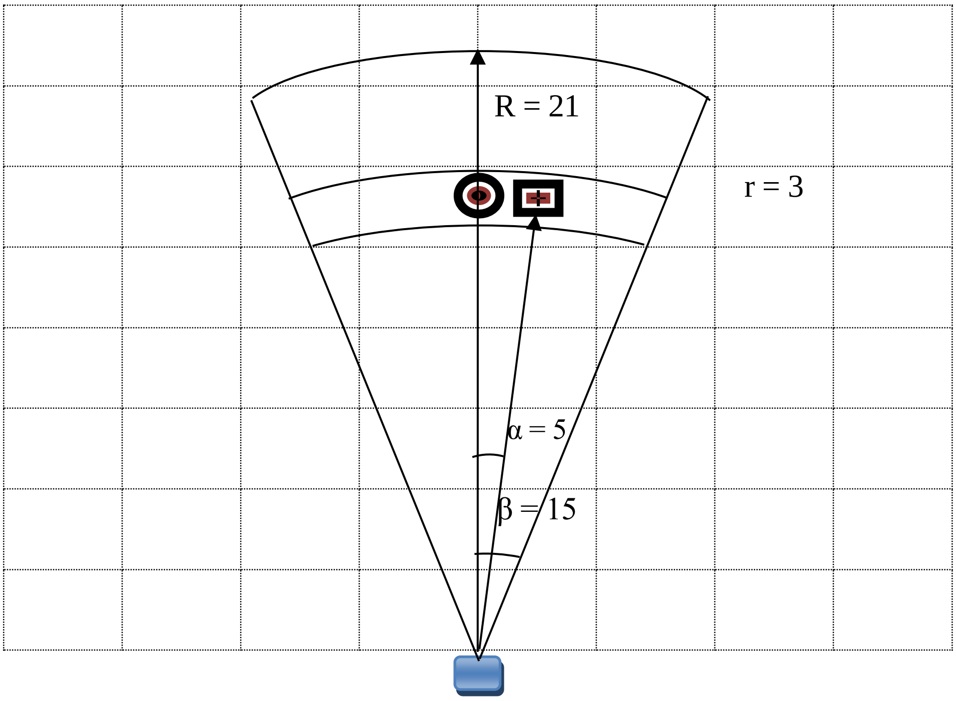
Figure 2.16 Sonar model for calculating the confidence of an element in region II (sonar reading of 3)

Since 2 < (3.0 - 0.05), the element is in Region II. Therefore, the correct formulas to apply are those in Eqn. 2.14.

$$P(Empty) = \frac{\left(\frac{R-r}{R}\right)+\left(\frac{\beta-\alpha}{\beta}\right)}{2}$$

$$P(Empty) = \frac{\left(\frac{21-2}{21}\right)+\left(\frac{15-0}{15}\right)}{2} \quad = 0.95$$

$$P\ (Occupied) = 1.0 - P(Empty) = 1 - 0.95 = 0.05$$



Scale: 1cell = 0.5 feet

Figure 2.17 Sonar model for calculating the confidence of an element in region I (sonar reading of 3)

The example in Fig. 2.17 shows an element in Region I. The probability the element in black is computed the same way, only using the equations for that region is different.

$$P(Occupied) = \frac{\left(\frac{R-r}{R}\right)+\left(\frac{\beta-\alpha}{\beta}\right)}{2} \times Max_{occupied}$$

$$P(Occupied) = \frac{\left(\frac{21-3}{21}\right)+\left(\frac{15-5}{15}\right)}{2} \times 0.98 \ = 0.74$$

$$P\ (Empty) = 1.0 - P(Occupied) \ \ = 1 - 0.74 \ = 0.26$$

2.4.2 Sonar Model by Moravec and Elfes

Elfes and Moravec [1], the researcher models the sonar beam as two probability density function, f_E and f_O. These functions measure the confidence and uncertainty of an empty and occupied region in the cone beam of the sonar respectively. In this model the following is defined:

- r Range measurement.
- $P_{i,j}^{O}$ Probability of a particular cell being occupied.
- $P_{i,j}^{E}$ Probability of a particular cell being empty.

- r_{min}Minimum distance.
- ε Mean sonar deviation error.
- ω Width of the cone.
- S_s sonar sensor
- δ_r Distance between S_s to$C_{i,j}$.
- θ Angle between the main axes of sonar beam to the line$S_sC_{i,j}$.

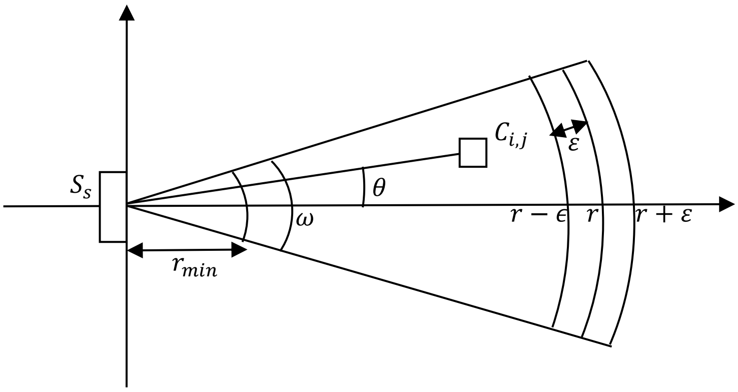


Figure 2.18 Field of view of the sonar sensor

The beam is divided in two regions:

a) The free space area or empty probability region which is the part of the beam between the sensor and the range where the obstacle was detected. This includes cell $C_{i,j}$ inside the sonar beam. Each cell has an empty probability:

$$P_{i,j}^{e} = E_r(\delta_r).E_a(\theta) \quad (2.15)$$

$E_r(\delta_r)$ is the estimation of the free space cell based on the range measurement from the sonar. The closer it is to the sensor the more likely it is to have a high estimation that the cell is empty.

$$E_r(\delta_r) = 1 - \left(\frac{\delta_r - r_{min}}{r - \epsilon - r_{min}}\right)^2 \; for \;\; r_{min} \le \delta_r \le r - \epsilon \quad (2.16)$$

$$E_r(\delta_r) \; = 0 \qquad otherwise$$

$E_a(\theta)$ is the estimation that the cell is free based on the angle of the cone beam.

The closer it is to the main axis and to the sonar the more estimate that it is empty.

$$E_a(\theta) = 1 - (\frac{2\theta}{\omega})^2 \qquad for \; \frac{-\omega}{2} \le \theta \le \frac{\omega}{2} \quad (2.17)$$

$$E_a(\theta) = 0 \qquad otherwise$$

b) The occupied area or probability occupied region. This is the area where the obstacle was detected. In this region the uncertainty to the exact distance to the obstacle (ε) has to be taken into account. The probability of a cell being inside the occupied region is :-

$$P_{i,j}^{o} = O_r(\delta_r).O_a(\theta) \qquad (2.18)$$

$O_r(\delta_r)$ is the estimation is based on the range reading . The closer is obstacle to the sonar the higher is probability that the cell is occupied

$$O_r(\delta_r) = 1 - \left(\frac{\delta_r - r}{\varepsilon}\right)^2 \quad for\ (r-\varepsilon) \le \delta_r \le (r+\varepsilon) \qquad (2.19)$$

$$O_r(\delta_r) = 0 \qquad otherwise$$

$O_a(\theta)$ is the estimation is based on the difference of the angle between the obstacle and the beam axis. The closer the obstacle is to the sonar the higher the probability that the cell is occupied.

$$O_a(\theta) = 1 - (\tfrac{2\theta}{\omega})^2 \qquad for\ \frac{-\omega}{2} \le \theta \le \frac{\omega}{2} \qquad (2.20)$$

$$O_a(\theta) = 0 \qquad otherwise$$

To see how these formulas would be applied, consider the example Fig. 2.19. The sonar has returned a range reading of 4.0 feet with a tolerance of ±0.05 feet. The element of interest grid*[i] [j]* is shown in black, and is at a distance r = 4 feet and an angle of θ = 7^0 from the robot. The first step is to determine which region covers the element. Since $\delta_r = 4$, which is $3.95 \le \delta_r \le 4.05$ both return range as well as distance of the cell of interest. Therefore, the correct formulas to apply are those in Eqn. 2.15

Given r = 4 feet, δ_r = 4 feet, ε = 0.05 feet, r_{min} = 0.5 feet, θ = 7°, ω = 30°.

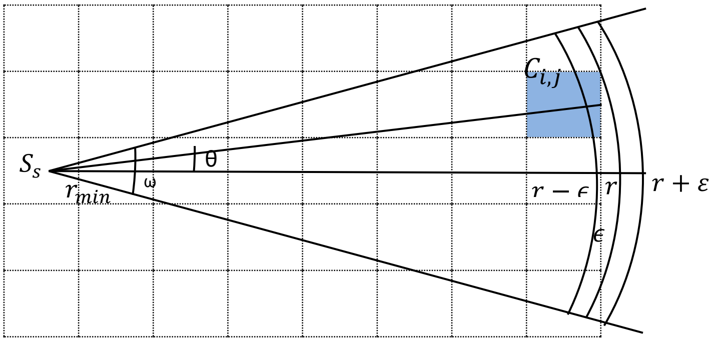


Figure 2.19 Field of view of sonar for calculating confidence of a particular cell in occupied region.

$$P_{i,j}^{o} = O_r(\delta_r).O_a(\theta)$$

Now $O_r(\delta_r) = 1 - \left(\frac{\delta_r - r}{\varepsilon}\right)^2$

$$= 1 - \left(\frac{4-4}{.05}\right)^2 = 1$$

$$O_a(\theta) = 1 - \left(\frac{2\theta}{\omega}\right)^2$$

$$= 1 - \left(\frac{2*7}{30}\right)^2 = 0.78$$

$P^o_{i,j} = 1 * 0.78 = 0.78$

Consider the example Fig. 2.20. The sonar has returned a range reading of 4.0 feet with a tolerance of ±0.05 feet. The element of interest grid*[i][j]* is shown in black, and is at a distance r = 4 feet and an angle of θ = 7^0 from the robot. The first step is to determine which region covers the element. Since $\delta_r = 3$, which lies in the range $0.5 \leq 3 \leq 3.95$. Therefore, the correct formula to apply is those in Eqn. 2.13

Given r = 4 feet, $\delta_r = 3$ feet, ε = 0.05 feet, $r_{min} = 0.5$ feet, θ = 7°, ω = 30°.

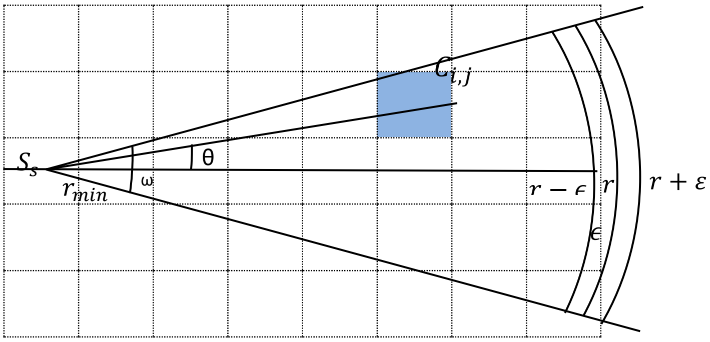


Figure 2.20 Field of view of sonar for calculating confidence of a particular cell in empty region.

$P^e_{i,j} = E_r(\delta_r).E_a(\theta)$

$$E_r(\delta_r) = 1 - \left(\frac{\delta_r - r_{min}}{r - \epsilon - r_{min}}\right)^2$$

$$= 1 - \left(\frac{3-0.5}{4-0.05-0.5}\right)^2$$

$$= 1 - \left(\frac{2.5}{3.45}\right)^2$$

$$= 0.475$$

$E_a(\theta) = 1 - \left(\frac{2\theta}{\omega}\right)^2 \quad = 1 - \left(\frac{2*7}{30}\right)^2 \quad = 0.78$

$P^e_{i,j} = 0.475 * 0.78 = 0.37$

2.5 Conclusions

In this chapter various types of approaches of internal representation have introduced. Grid based internal representations are most widely used representations in the field of mapping. Various types of active and passive sensors are also shown with their advantage and disadvantage. Finally two sensor model, one is given by Robin R. Murphy and another one is given by Moravec and Elfes is presented.

Chapter 3

Sensor Data Fusion

3.1 Introduction

This chapter presents an overview of the potential advantages of integrating and/or fusing data as the multisensor fusion methods used in mobile robot applications. In order to achieve autonomy in an intelligent system like a mobile robot, it must be able to sense the environment. Sensing is achieved by sensors and these are required to provoid useful information about the robots surroundings, this information can be used to continuously learn about the world and the mobile robot can use it to creat its own world model and update it. It will be impossible for a mobile robot to interact with its surroundings without sensing and all the tasks the robot has to perform must be programmed a-priori. It is also impossible for a single sensor to fully and accurately determine the current status of the environment all by itself. In this situation, the system must gather information from different source as well as from different views and rely on the fusion of sensor data to increase its capability and make a final decision.

Some definitions of sensor data fusion are given below:

M. Kam defines “sensor fusion as the process of integrating data from distinctly different sensor for detecting objects and for estimating parameter and states needed for robot self-localization, map making, path planning, path computing, motion planning and motion execution” [15].

Luo and kay[11,46 and 47] give the following definition: Multisensor integration refers to the synergistic use of information by multiple sensory devices to assist in the accomplishment of the task by the system. Sensor fusion refers to any stage in the integration process where there is an actual combination of different sources of sensory information into one representational format. Potential advantages in integrating and/or fusing information correspond to the nations of redundancy, complementry, timeless and cost of information provided by the system. Characterisation of multisensor fusion can take place at signal, pixel, feature, and symbol level.

In my words sensor fusion is a process of integrating sensory information from different sources as well as different views into one representation to make a final decision.

3.2 Multisensor Integration:

As above mentioned that multisensor integration is the synergistic use of the information provided by multiple sensory devices to assist in the acomplishment of a task by the system. In the sense, the external sensor provide the system for different purposes. The potential advantages in integrating and/or fusing data from multiple sensors correspond respectively to the nations of of redundancy, complementry, timeless and cost of information [11].

- Redundant: It is redundant because the same feature in the environment is perceived by a singular sensor over time or by a group of sensors. The fusion of redundant information can reduce the overall uncertainty and thus increase the accuracy which the system perceives the feature from the environment.
- Complementary: It is complementary because different features in the environment are perceived by different sensors. This means that the information from multiple sensors allows features in the environment to be perceived that are impossible to perceive using a single sensor working seperately.
- More timely: It refers to the speed of the information.Multiple sensors can

achieve the same speed as a single sensor or the processing parallelism that could be achieved as a part of the integration process.

- Less costly: It refers to the context of a system with multiple sensors where the information can be obtained at lesser costs compared with the information of a single sensor. In other words, the total cost of the information of a single sensor must be compared with the total cost of the information of the integrated multisensor system.

Autonomous system like a mobile robot can have different approaches for multisensor integration depending of the sensors to be used and the task to be done. But, certain basic functions are common to most implementations. Luo [11 and 46] describe a general system for multisensor integration, can be depicted in figure 3.1. The diagram shown in figure is a composite of the basic functions which will described below.

- Sensors: A group of n different sensors that provoid an input to the integration process.

- Sensor Model: The data from the sensors must be effectively modelled before it is used by the integration. A sensor model represents the uncertainty and the error in the data from each sensor. This uncertainty can often be adequately modelled as a Gaussian distribution.
- Sensor Registration: Before the data is fused it has to be commensurate both in spatial and temporal dimensions i.e. the data refers to the same location in the environment over the same time period.
- Sensory Processing: Fusion is done at the symbol, feature, pixel and signal level and this step is described in next section. If the data of the sensor is significantly different from the other sensor will be used to guide the operation of the other sensors of the system.
- World Model: The data gathered from the different devices can be used to either construct a world model dynamically during operation or to be added to a predefined model of the world. The world model is usually defined in terms of high level representation for multisensor mobile robot navigation and control.

- Sensor Selection: Selects the best sensor configuration when the robot is performing a task.

3.3 Multisensor Fusion

As it is stated above, multisensor fusion refers to any stage in the integration process where there is an actual combination of different sources of sensory in formation into one representational format. The fusion of the data from multiple sensor can take place at different levels in the representation process. A useful categorisation is to consider multisensor fusion as taking place at the signal, pixel, feature and symbol level [11 and 47].

- Signal Level Fusion: The sensor in this fusion level must have a great degree of registration. This means that the sensor must be in temporal as well as in spatial registration i.e. the signal can be registered spatially by having the sensors co aligned in the same plateform. The resulting signal from the fusion process is usally of the same form as the original signal but with greater quality. Random variable can be used to model the signals from the sensors.

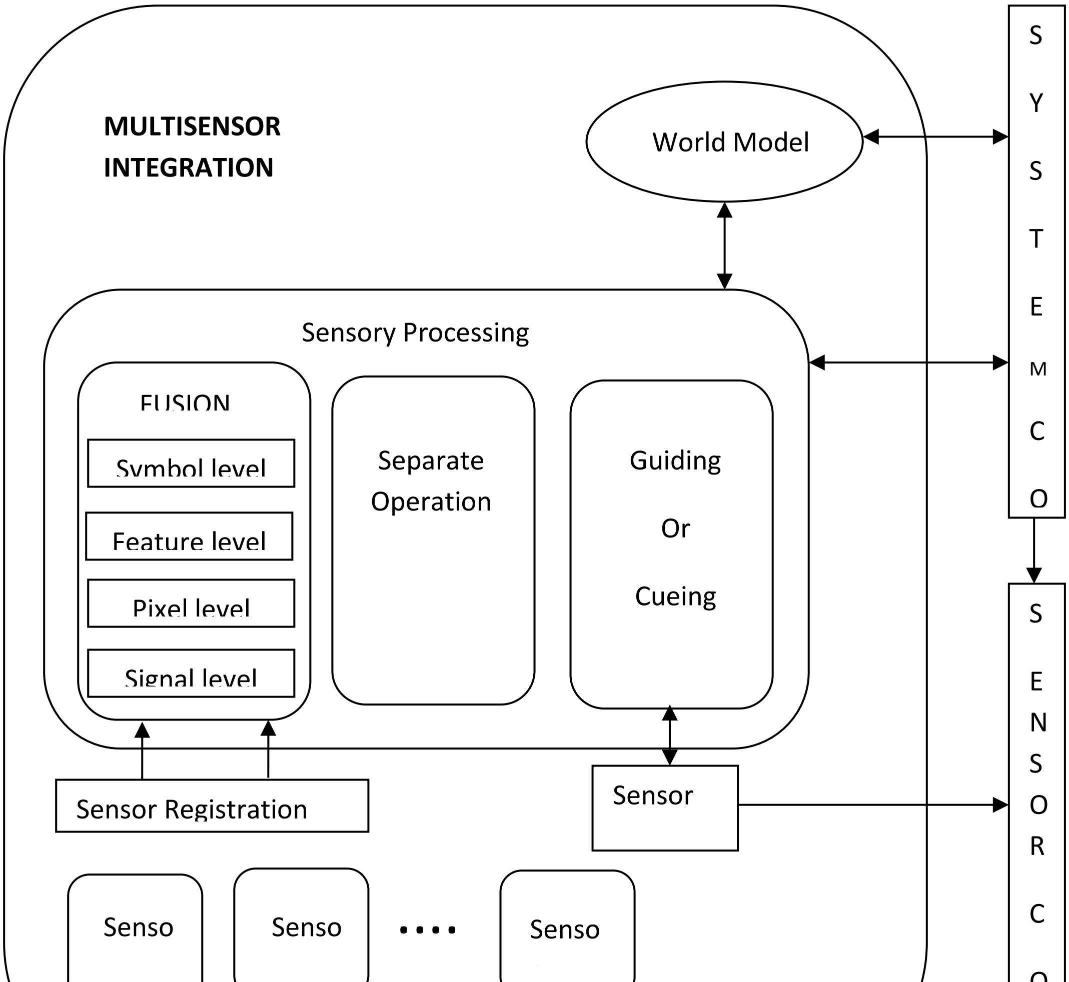


Figure 3.1 Fuctional diagram of multisensor integration and fusion.

- Pixel Level Fusion: The sensor in this level of fusion come from a single image sensor (like a CMOS or CCD camera), or a group of sensors (like a stereo pair of cameras). The fused image can be created either by the fusion of pixel by pixel or by the fusion of associated local neighourhoods of pixels in each of the emages. The sensor registration can be achieved if either a single sensor or multiple sensors provide the same resolution and share the same optics and mechanics. Tasks like segmentation or feature extraction can be improved by the fusion of multisensor data at a pixel level. A sequence of images from a single sensor and images from a stereo pair are good candidates for pixel level fusion.
- Feature Level Fusion: When multiple sensors report similar features at the same location in the environment, the likehood that the features actually present can be increased and the accuracy with which they are measured can be improved. Additional features can be created as the result of the fusion process and they can be a composite of the component features. In order to achieve the feature-level fusion, the sensor can be placed either in the same plateform or in a different one, making the sensor registration less stringent than those for signal and pixel level fusion.Typical features extracted from an image and used for fusion include edges and regions of similar intensity or depth.
- Symbol-Level Fusion: A symbol come from matching sensory information into a model, and it also associates the degree sensory information into a

model. Symbol-level fusion can be done at the highest level of abstraction and can be the only means of sensory information that can be fused if the information provided by the sensor is very dissimilar or corresponds to different regions in the environment. A statistical inference can be used for symbol level fusion, the symbols to be fused are represented as conditional probability expressions and their uncertainty measures corrospond to the probability measures associated with the expressions.

3.4 Sensor Fusion Techniques

.Different Sensor Fusion techniques are used[11,46 and 47], they are following:

- Bayesian Techniques
- Dempster-Shafer Techniques
- HIMM
- Kalman Filter Localization Techniques
- Probabilistic Grid-based Map (PGM**)**
- Fuzzy Logic Algorithm
- Neural Network Algorithm

(A) Bayesian Techniques

The most popular evidential method for fusing evidence is to translate sensor readings into probabilities and to combine probabilities using Bayes' rule. Elfes and Moravec at Carnegie Mellon University pioneered the probabilistic approach in the early 1980's. Later Morava turned to a form of Bayes' Rule which uses probabilities expressed as likelihoods and odds. This has some computational advantages and also side-steps some of the problems with priors. The likelihood/odds formulation is equivalent to traditional approach presented here. In a Bayesian approach, the sensor model generates conditional probabilities of the form P(s/H). These are then converted to P (H/s) using Bayes' rule. Two probabilities, either from two different sensors sensing at the same time or from two different times, can be fused using Bayes' rule.

(B) Dempster-Shafer Theory

An alternative theory of evidence is Dempster-Shafer theory which produces results similar to Bayesian probabilities. It is a much newer theory, originating in the work of A.P. Dempster, a mathematician at Harvard, during 1960's with extensions by Glen Shafer in 1987. Whereas Bayes' rule relies on evidence being represented by probability functions, Dempster-Shafer theory represents evidence as a possibilistic belief function. Possibilistic means that the function represents partial evidence. For example, a reading may provide direct evidence for an event H, but, due to occlusions, not be perceiving the entire object. Therefore, there is a possibility evidence could be higher than was reported. The possibilistic belief functions, also called Shafer belief functions, are combined used Dempster's rule of combination. The rule of combination is very different from Bayes' rule, although they provide similar results. Unlike Bayes' rule, Dempster's rule has a term which indicates when multiple observations disagree. This conflict metric can, be used by the robot to detect when its occupancy grid may be subject to errors.

(C) HIMM (Histogrammic in Motion Mapping)

The Histogrammic in Motion Mapping (HIMM) algorithm developed by Borenstein and Koren at the University of Michigan provides a different approach to scoring whether a particular element in an occupancy grid is occupied or empty. The motivation for HIMM stemmed from a need to improve obstacle avoidance. In order to safely run their Cyber motion robot CARMEL, at its top speed of 0.5 meters/ sec, they needed to update an occupancy grid on the order of 160ms. The processor was a 20MHz 80386, fast for 1992, but still quite slow. The Bayesian model at that time was well accepted but the sheer number of computations involved in updating prevented the algorithm from a satisfactory fast execution. HIMM was developer quasi-evidential technique where it scores certainty in a highly specialized way suitable for sonars.

The University of Michigan's robotics team entered CARMEL in the 1992 AAAI Mobile Robot Competition. CARMEL resoundingly won first place that year in the task of navigating between a se waypoints marked by landmarks on posts. By using HIMM in conjunction with the vector field histogram (VFH) obstacle avoidance algorithm, CARMEL was able to navigate at velocities an order of magnitude higher than the entries, and avoided obstacles of all sizes more reliably. After that, HIMM and occupancy grids became standard on many platforms.

(D) Traditional Kalman Filter Algorithm

The Kalman filter, is an optimal linear estimator based on an iterative and recursive process. It is used in a wide variety of applications, and applies particularly well to sensor fusion. It recursively evaluates an optimal estimate of the state of a linear system. The Kalman filter process consists of two sub processes, repeated iteratively at a time step of dt: the time update and the measurement update. An easy way to explain the Kalman filter is to say that it incorporates two sets of sensor measurements, one in each sub-process. In the time update process, a prior estimate Xprior (k) is computed based on the previous state estimate X (k-1) and sensors indirectly related to the state (e.g. accelerometers or velocity sensors when the state is position, also called dead-reckoning sensors). Then, in the measurement update process, this prior estimate is blended with direct measurements of the state (position) coming from other sensors, thus obtaining the new updated state estimate X (k).

(a) Time Update Process

The optimal estimate from the previous iteration, noted X(k-1) is projected in time through the state transition matrix A, and the noisy inputs U(k) (indirect sensors) are fed to the system through matrix B, relating the inputs to the state.

$$Xprior(k)=A.X(k1)+B.U\,(k) \tag{3.1}$$

$$Pprior(k)=A.P(k-1).A+Q \tag{3.2}$$

(b) Strengths and Weaknesses

The Kalman filter is popular in sensor fusion applications because its formulation makes it easily adaptable to sensor fusion. Dead-reckoning sensors (velocity or acceleration) can be blended with sensors measuring direct position to give an optimal estimate. This combination brings out the good sides of all the sensors position estimate errors over time for different combinations of noisy sensors. The noise from the accelerometer is accumulated over every iteration because the acceleration is integrated to obtain position. This results in a diverging position error and therefore this sensor is unusable over long time periods.

(E) Probabilistic Grid-based Map (PGM)

A new method for exploring and navigating autonomously in indoor environments is described. This method merges a local strategy, similar to wall following to keep the robot close to obstacles, within a global search frame, based on a dynamic programming algorithm. The approach considers information from its sensors, to learn and use a Probabilistic Grid-based Map (PGM) of an environment. A PGM is a, two dimensional map where the environment is divided in square regions or cells of the same size that have occupancy probabilities associated to them.

The two principal components for building a PGM are exploration and position estimation. Research on exploration strategies has developed two general approaches, reactive and model based. By far the most widely-used exploration strategy in reactive robotics is wall following. Model based strategies vary with the type of model being used but they are based on the same underlying idea: go to the least-explored region.

Exploration

The PGM building process does the following general steps:

1. Process the readings taken by all the sensors and update the probability of occupancy of the cells in the PGM (Sensor Fusion Step).
2. Update the travel space accordingly with the changes in the PGM.
3. Choose the next movement using value iteration. If the movement is not valid then the map is complete.
4. Execute the movement.
5. Get readings from the sensors and correct udometric error (Position Tracking Step).
6. Go to the first step.

The general idea for exploration is to move the robot on a minimum-cost path to the nearest unexplored grid cell. The minimum-cost path is computed using value iteration, a popular dynamic programming algorithm. In the cost for traversing a grid cell is determined by its occupancy value, while in the cost is determined by the distance between cells.

(F) Fuzzy Logic Algorithms

The fuzzy logic algorithm for sensor fusion related to mapping the environments. The algorithm deals with unknown a-priori sensory distributions and with asynchronous update of the sensors. Feedback is calculated every time each logical sensor sends new data and is used to measure on-line the logical sensors performances. Analysis in an indoor mobile robot experiment indicated superior performance of the adaptive fuzzy logic algorithm when compared to three other sensor fusion algorithms.

In this algorithm, each logical sensor makes decisions asynchronously and independently and fusion is conducted asynchronously. Feedback is calculated every time each logical sensor sends new data and is used to measure the logical sensors performances. This is conducted on-line according to sensory and environmental conditions.

(G) Neural Network Algorithms

For many navigation tasks, a single sensing modality is sufficiently rich to accomplish the desired motion control goals; for practical autonomous outdoor navigation, a single sensing modality is a crippling limitation on what tasks can be undertaken. Using a neural network paradigm particularly well suited to sensor fusion. It is successfully performed simulated and real-world navigation tasks that required the use of multiple sensing modalities.

3.5 Sensor fusion Framework

3.5.1 Bayesian Inference

Bayesian is a statically inference method in which observations are used to update or infer the probability that a hypothesis may be true. Bayesian inference is an approach to the statistics in which all forms of uncertainty are expressed in the form of probability. Bays theorem is also known as Bays Rule.

Theorem 3.1. (Bays Rule) If the event $B_1, B_2, \dots \dots . B_K$ constitute a partition in the sample space s, where p$(B_J \neq 0\,, for\ J = 1,2,3, \dots \dots, K.$ then for any even A in S such that P(A) ≠0.

$$P(B_i|A) = \frac{P(B_i \cap A)}{\sum_{j=1}^{k} P(B_j \cap A)} = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)} \qquad \text{for} \qquad i=1,2,\dots\dots,k$$

(3.3)

- B_i is one of *i* mutually exclusive (disjoint) event to be estimated.
- *A* is the evidence event.
- *P(B_i)* is the prior probability of the event B_i. It is prior in the sense that it does not take into account for any information about *A*.
- $P(B_i|A)$ is the conditional probability of B_i given *A*. It is also called the posterior probability because it is derived from or dependent upon the specified value of A
- $P(A|B_i)$ is the conditional probability of *A* given B_i
- $\sum_{j=1}^{k} P(B_j \cap A)$=$\sum_{j=1}^{k} P(B_j)P(A|B_j)$ is the total probability of A, and acts as a normalizing factor.

3.5.2 Recursive Bays Update Rule

Bays rule provides a way of computing a posteriori probability of a hypothesis being true giving supportive evidence [9,11] have successfully used bays rule to update the occupancy grid for multiple sensor readings ($s_i, \dots\dots, s_n$). The eqn. (3.3) transformed to eqn.(3.4) and (3.5) to the occupancy grid framework for multiple sensor readings.

$$P_{i,j}^{o|s} = \frac{P_{i,j}^{s|o} P_{i,j}^{o}}{P_{i,j}^{s|o} P_{i,j}^{o} + \left(1 - P_{i,j}^{s|o}\right)\left(1 - P_{i,j}^{o}\right)}$$

(3.4)

$$P_{i,j}^{e|s} = \frac{P_{i,j}^{s|e} P_{i,j}^{o}}{P_{i,j}^{s|e} P_{i,j}^{o} + \left(1 - P_{i,j}^{s|e}\right)\left(1 - P_{i,j}^{e}\right)}$$

(3.5)

The following statements are defined

- The relevant evidence of true parameter B_i is given by $P_{i,j}^{o}$ and $P_{i,j}^{e}$, meaning that they are the prior probabilities of the cell $C_{i,j}$ being occupied or empty they are taken from the existing map.
- The conditional probability $P(Ai \,|\, B_i)$ is given by $P_{i,j}^{s|o}$ and $P_{i,j}^{s|e}$, which are the conditional probabilities that a sensor reading will exit given the state of the cell $C_{i,j}$, being occupied or empty. This conditional probability is given by the probabilistic sensor model.
- The conditional probability $P(B_i|A)$ is given by $P_{i,j}^{o|s}$ and $P_{i,j}^{e|s}$, which is the conditional probability that a cell is occupied based on the past sensor readings it is the new estimate.

A new sensor reading s interduces additional information about about the state of the cell $C_{i,j}$ This information is done by sensor model $P_{i,j}^{s|o}$ and it is combined with the most recent probability estimate stored in the cell. This combination is done by the recursive bays rule $P_{i,j}^{o|s}$ based on the current set of readings to give a new estimate $P_{i,j}^{o|s}$. It is worth noting that when initializing the map an equal probabibilty to each cell Cij must be assigned. In other words the intial map cell prior probabilties are $P_{i,j}^{o}$ = $P_{i,j}^{e}$ = 0.5.

3.5.3 Fusion of sensors with two occupancy grids

Fusion of two or more occupancy grid is carried out by constructing occupancy grid for each sensor as a local grid map of individual sensor and these grids are used to build up the resulting grid. Each cell in both grids is evaluated before fusing the overlapped area of both the grids. In this sense, three rules are applied to obtain the resulting probability for each grid.

- If a cell $C_{i,j}$ in the source grid has higher probability of representing occupied space then a pre defined threshold T_0 , then the probability of the resulting cell being occupied is said to 1.
- A single cell $C_{i,j}$ is evaluated in the interval {1/2 , T_0}.
- If the value that represents occupancy from a single cell does not fall in the former two rules the value in the cell is keep it.

3.5.4 Evaluation criterion for a single cell $C_{i,j}$

Let $C_{i,j}$ be a single cell in one of the source grids to be evaluated and $P_{i,j}^{Os}$ the probability of a single cell being occupied based on the sensor reading s, so that

$$P_{i,j}^{Os} = 1 \qquad \text{for} \quad P_{i,j}^{Os} > T_0 \tag{3.6}$$

$$P_{i,j}^{Os} = \frac{P_{i,j}^{Os} + T_0 - 1}{2.T_0 - 1} \qquad \text{for} \quad P_{i,j}^{Os} \in \{1/2\,, T_0\}$$

$$P_{i,j}^{Os} \qquad \text{otherwise}$$

The computed values are than applied in Bayes 'rule to obtain probabilities in the resulting grid:

$$P_{i,j}^{0_{S_1,S_2}} = \frac{P_{i,j}^{S_1} P_{i,j}^{S_2}}{P_{i,j}^{S_1} P_{i,j}^{S_2} + \left(1 - P_{i,j}^{S_1}\right)\left(1 - P_{i,j}^{S_2}\right)} \tag{3.7}$$

Where $P_{i,j}^{S_1}$ is the modified probability of occupancy from the sonar sensor 1 and $P_{i,j}^{S_2}$ is the modified probability of occupancy from the sonar sensor 2. $P_{i,j}^{0_{S_1,S_2}}$ is the probability of occupancy after data fusion.

The fusion of two sensors in two different grids has the advantage of not affecting the data between the two representations. A single grid is suitable for fusion of data from single sensor. Two separate grids are required for mapping the area in front of the robot furnished by each sensor. The final fusion takes place on another part of the computer memory in different grid. It is used for data fusion process and is not used for integration of range from individual sensor [9].

3.6 Conclusions

This chapter deals with an overview of sensor data fusion and integration with its function diagram. Various sensor fusion techniques are also presented. Finally sensor fusion frame work is shown. In which bayesian inference, recursive bayes update rule for updating the sensor model and for fusion process as well as evaluation criteria of a single cell and fusion of sensor with two occupancy grid are also presented.

Chapter 4

Experimental Setup and Occupancy Grid Mapping Results

4.1 Introduction

There are a large number of algorithms available for disparity map using stereo vision system. The correspondence problem in stereo vision system is the problem of finding which element in the right image corresponds to the left image. The two main approaches in finding the solution to the correspondence problem in a stereo pair of images are correlation-based and feature based. The correlation-based correspondence algorithms usually find the correspondence between pixels in both images creating a dense disparity map were the 3D position of each pixel can be computed. The feature based correspondence algorithms find correspondence between features from both images. These features are edges, lines, points etc. Certain criteria must be satisfied in order to find the correspondence between features; for instance criteria for matching a line could be the length of the line, its orientation, the coordinates of a midpoint and the average contrast along the line edge. The disparity maps created by the feature-correspondence algorithms are more sparse than the ones created by the correlation-based, which makes the correspondence search less time consuming. The approach used in this paper is the scale invariant feature transform, because SIFT- feature based correspondence reduces the number of matches between left and right images. And examined effects of threshold value on disparity matching using SIFT.

This chapter deals with image acquisition and various steps of SIFT algorithm for disparity matching. It includes an empirical technique for optimization of threshold ratio and a distance measurement method to measure the distance to an object which is located outside the axes of stereo cameras. The simulation results of the measured distance of the objects in the dynamic environment are shown by using Vision Lab Feature Library (VLfeat) toolbox in MATLAB.

In this chapter, SIFT feature model has been implemented for mapping the occupancy grid of indoor environment the mobile robot has to deal with. Acquisition of priory information to the environment is performed by moving the mobile robot having a sonar sensor mounted on it. The range information given by sonar sensor and its orientation gives the polar coordinate of the occupied cell. The coordinates of occupied cell is calculated by the transformation of polar coordinates given by sonar sensor to the Cartesian coordinates.

Moravec- Elfes model has been used for generating the occupancy grid by SIFT feature model. And the final occupancy grid has been generated by updating the occupancy grid generated by SIFT by using Recursive Bays Rule. The priory to the environment is used for updating the occupancy grid. The chapter is concluded by discussing the results obtained and applications in this field.

4.2 Work Architecture

The environment the mobile robot has to deal with is an indoor environment. There is a variety of approaches to the integration and fusion of information from the combination of different types of sensors. Sonar sensor has been used for acquiring the prior information from the static environment.

The vision system is one of the most powerful sources of information. The vision algorithms used in the field of mobile robots are mainly based on feature extraction, where the features are edges or corners. This fact led to the exploration of new alternatives for extracting features from the scene to use them in the field of sensor fusion in mobile robot. It does provide a good basis for matching images of different sizes. SIFT algorithm is used to extract the features from the stereo snapshots taken from the robot during its path.

Acquisition of priory information to the environment is performed by moving the mobile robot having a sonar sensor mounted on it. The range information given by sonar sensor and its orientation gives the polar coordinate of the occupied cell. The coordinates of occupied cell is calculated by the transformation of polar coordinates given by sonar sensor to the Cartesian coordinates.

SIFT algorithm for disparity matching has been implemented for the measurement of distance of the objects. This is performed by a proposed empirical technique for optimization of threshold ratio and a distance measurement method to measure the distance to an object which is located outside the axes of stereo cameras.

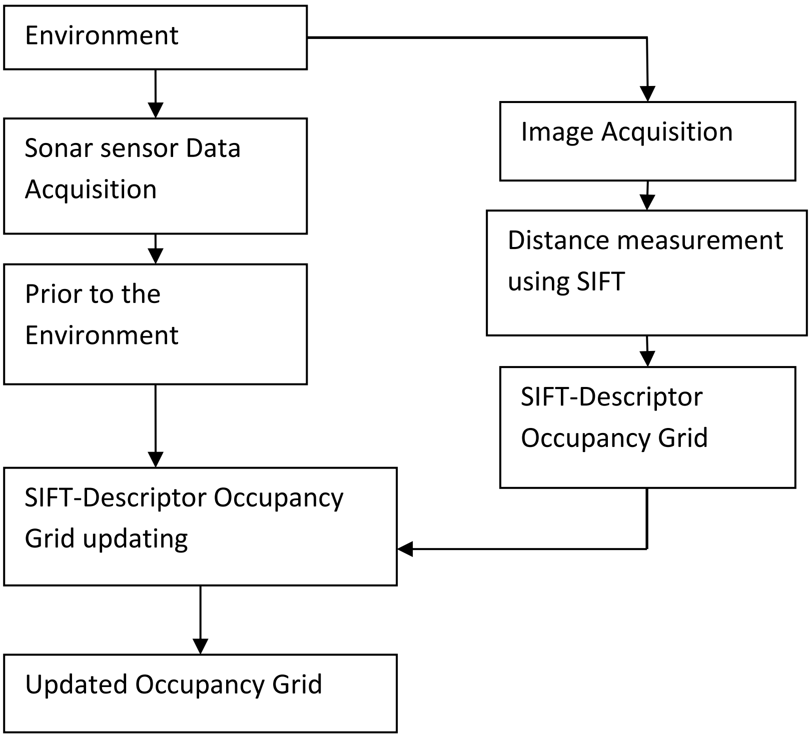

Environment
Sonar sensor Data Acquisition
Prior to the Environment
Image Acquisition
Distance measurement using SIFT
SIFT-Descriptor Occupancy Grid
SIFT-Descriptor Occupancy Grid updating
Updated Occupancy Grid

Figure 4.1 Work Architecture

Mathies and Shafer [66] successfully modelled the triangulation error in stereo matches with normal distributions. This approach has been taken into account on modelling the occupied area of the SIFT-descriptor. The Elfes approach has been considered to model the empty areas from the cameras to the boundary of the 3D normal distribution. These two models led to a SIFT- descriptor probabilistic model. The SIFT- descriptors identified by the SIFT algorithm in a snapshot is modelled and registered into occupancy. The priory information of the environment is used to integrate the SIFT- descriptor occupancy grid by using Recursive Bays formula.

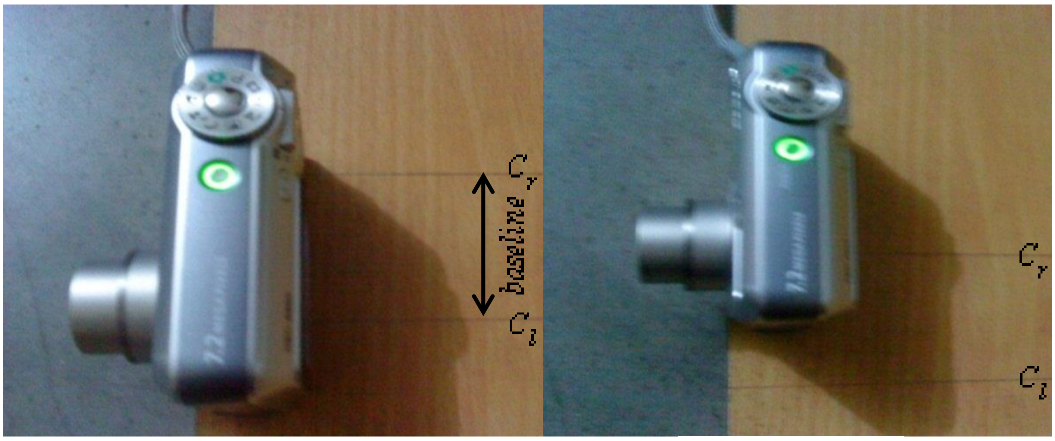
4.3 Image Acquisition

Stereo pair of images is being taken by using single CCD camera, Sony cyber shot (DSC-S650). The camera is placed on a slider at two different position separated by a distance of 5cm keeping the axis of the cameras parallel. Parallel camera setup has been shown in fig. 4.2.

The features of the camera are given in table 4.1 and the fig. 4.3 shows the acquired stereo pair of images.

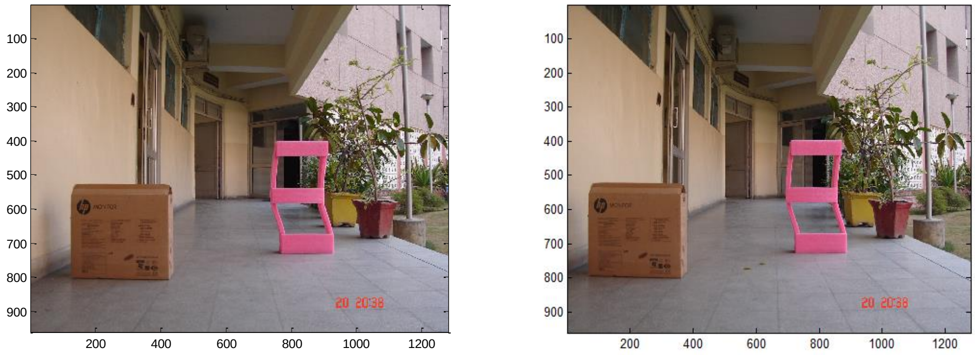
Table 4.1 Features of Camera (DSC-S650) for Simulation

Item			**Feature**
Setup Method			Parallel Setup
Distance between two cameras			50mm
Focal Length			5mm
CCD Camera	Size	1 Pixel	8.6μm X 6.6μm
		Whole	1280pixel X 960pixel
	Resolution	Horizontal	72dpi
		Vertical	72dpi



Left Camera Right Camera

Figure 4.2 Parallel Camera setup



Left Image Right Image

Figure 4.3 Stereo pair of Images.

4.4 Experimental setup and Occupancy Grid Mapping Results

The whole experimental work has been performed in four different experiments. First experiment is about acquisition of prior information from the static environment the mobile robot has to deal with. Second experiment discusses the method for distance measurements of the objects in the dynamic environment using SIFT. The third experiment deals with modelling the occupied area and empty area using SIFT feature descriptor model. And the fourth experiment is integrating the occupancy grid with prior information about the environment by Recursive Bays Rule.

4.4.1 Acquisition of prior information

4.4.1.1 Experimental setup

The static environment in which the mobile robot has to deal with is shown in fig. 4.4. The environment is known and the position of static objects in the environment is as shown in figure. The area beyond right of the transparent line is at the depth of 0.6m from the level of plane where mobile robot has to move. Some assumption has been taken while mapping the environment according to the priory. While mapping the occupied area we assumed that the cells lying on position of the transparent line are occupied.

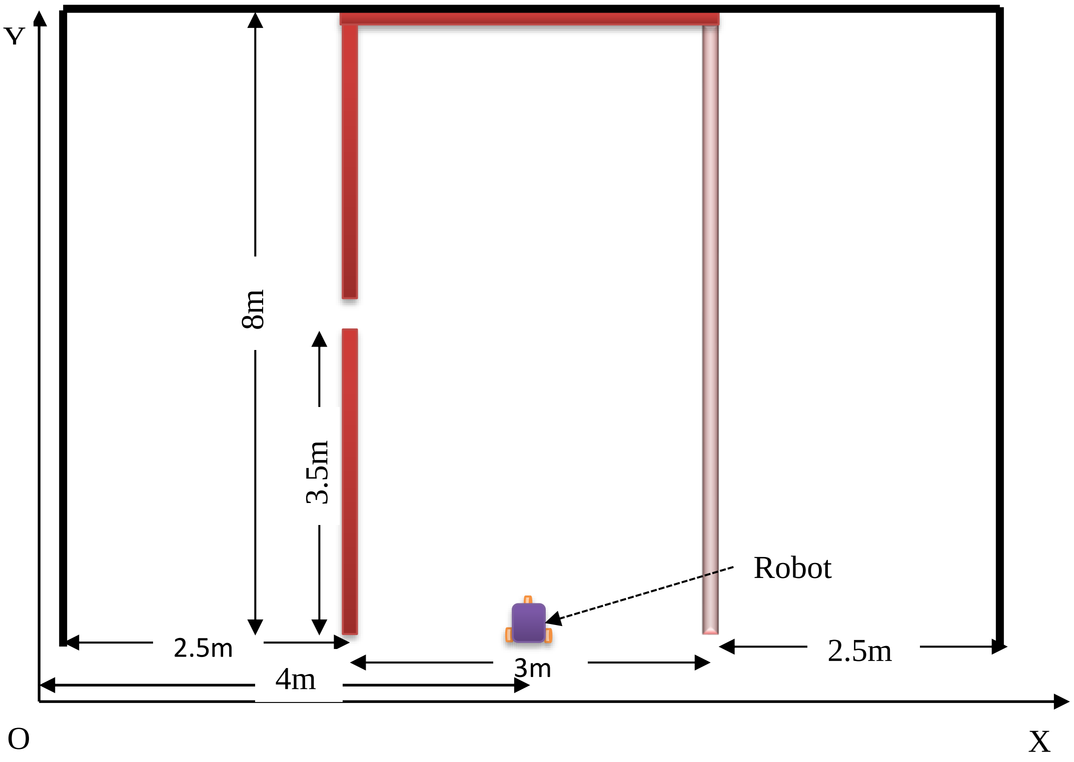


Figure 4.4 Geometric map of the static environment

Acquisition of priory information to the static environment is performed by moving the mobile robot having a sonar sensor mounted on it. The range information given by sonar sensor and its orientation gives the polar coordinate of the occupied cell. The coordinates of occupied cell is calculated by the transformation of polar coordinates given by sonar sensor to the Cartesian coordinates.

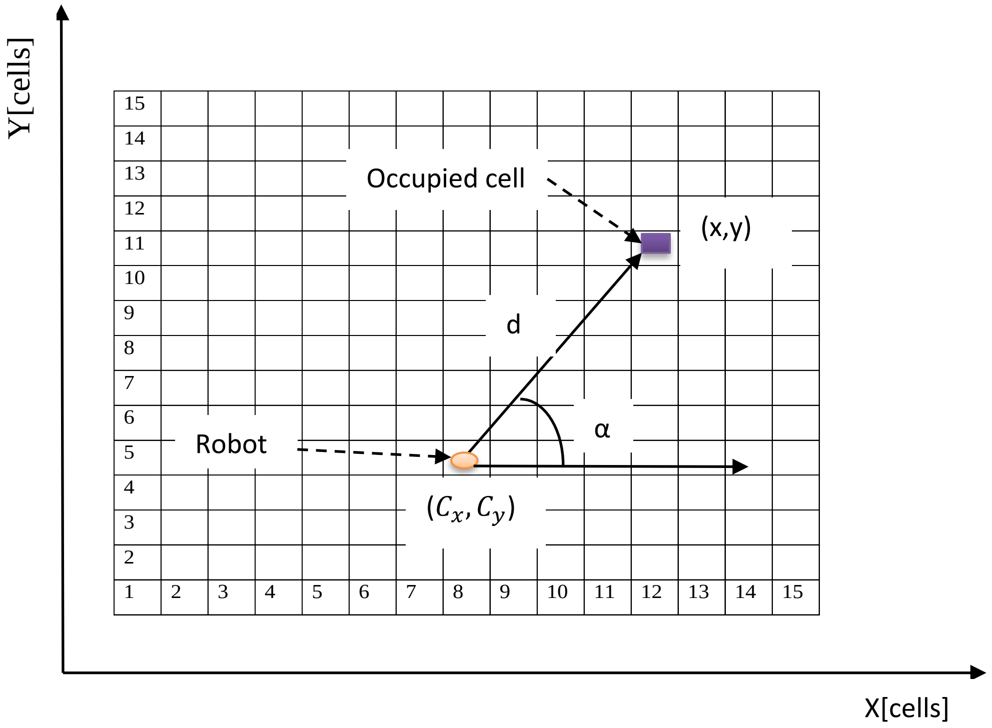


Figure 4.5 Occupancy grid of 15x15 cells with robot position (C_x, C_y) and occupied cell having polar coordinate (d, α)

The Cartesian coordinates of the occupied cell has been calculated by using equation 4.1 and 4.2.

$$x = C_x + dcos\alpha \qquad (4.1)$$

$$y = C_y + dsin\alpha \qquad (4.2)$$

4.4.1.2 Simulation results

The range information of the static environment has been acquired by moving the mobile robot throughout the environment that the mobile robot has to follow. The mobile robot having a sonar sensor mounted on it, sonar sensor is connected with a laptop by using Doctor DAQ data acquisition device as shown in fig. 4.6.

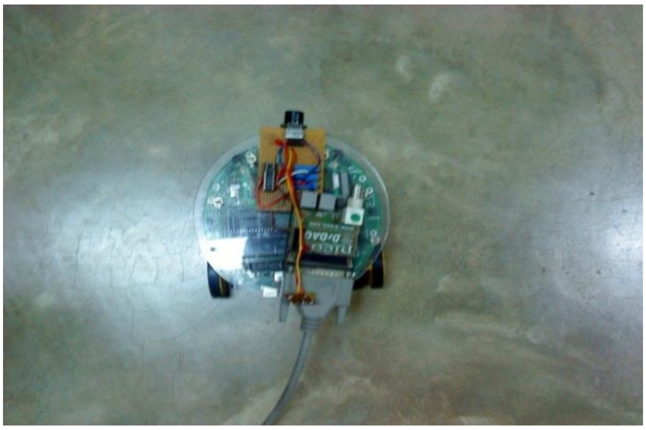


Figure 4.6 Mobile robot having sonar sensor mounted on it

The range information given by sonar sensor is stored in a MAT file. The range information is set as input to a Simulink programme for the transformation of polar coordinates of the occupied cell to the Cartesian coordinates. Whole static environment has been mapped by using an occupancy grid of 100x100 cells, each cell having size of 8cmx8cm. According to acquired prior information the occupancy grid map of the static environment is shown in fig. 4.7.

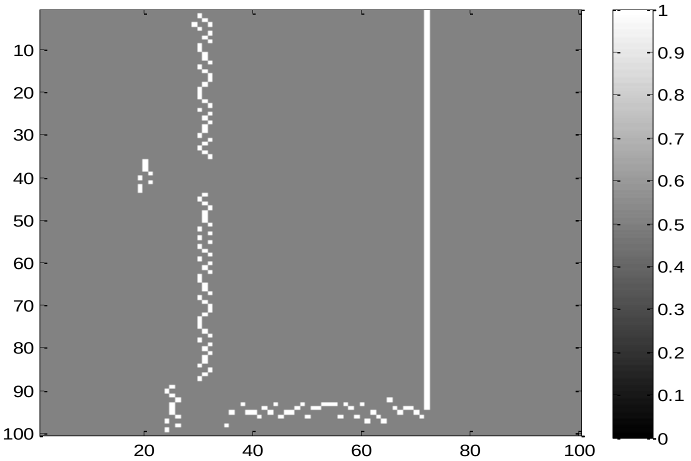


Figure 4.7 Occupancy grid map of the environment according to the priory.

4.4.2 Distance measurement using Scale Invariance Feature Transform (SIFT)

4.4.2.1 The Scale Invariance Feature Transform

The Scale Invariance Feature Transform (SIFT) method was originally introduced by Lowe [55] and refined by the same author [57].

It is a method for extracting distinctive invariant features from digital images, in which the features are invariant to scale and rotation. They also provide a robust matching across a substantial range of affine distortion, change in 3D view point, addition of noise and change in illumination.

The SIFT algorithm consists of the following major steps:

- Scale-space peak detection
- Accurate key-point localisation.
- Majority orientation assignment
- Computation of the local image descriptor.
- SIFT descriptor matching.

(A) Scale-space peak detection: This is the stage where interest points, which are called key-points in the SIFT, are detected. For this, the image is convolved with Gaussian filters at different scales, and then the differences of successive Gaussian-blurred images are taken. Key-points are then taken as maxima/minima of the Difference of Gaussian (DoG) that occur at multiple scales.

A Scale-space function is defined as;

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

(4.3)

Where: $\mathrm{G}(\mathrm{x}, \mathrm{y}, \sigma)$ be the Gaussian kernel and $I(x, y)$ be the input image.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2}\ e^{-(x^2+y^2\ )/2\sigma^2}$$

(4.4)

The scale parameter σ just indicates which scale level is being defined.

The Difference of Gaussian function is defined as;

$$DoG(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma)$$

(4.5)

The Difference of Gaussian image $D(x, y, \sigma)$ is defined as;

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (4.6)$$

D.G. Lowe [57] Proposes to find potentially interesting key-point locations by detecting extrema in the scale-space representation obtained from convolving the image with the DoG function.

In the implementation, before applying the formula (4.6), the original image, $\mathrm{I}(x, y)$, is up sampled by a factor of two using linear interpolation. [57] Mentions that, by up sampling, the image has the effect of increasing the number of stable key-points by almost a factor of 4. Afterwards the up sample image is incrementally blurred by a Gaussian kernel. Blurring the original image by a Gaussian kernel has the equivalent effect of low-pass filtering, discarding the higher frequencies. The creation of images separated by a constant factor k in scale space,$L(x, y, k\sigma)$, brings the concept of an octave.

An octave O_c is defined as the creation of incrementally convolved images with a Gaussian kernel in scale space. An octave in scale space is divided into a fix integer number s_u of intervals. The maximum of intervals are chosen as doublingσ, so $k = 2^{\frac{1}{s_u}}$.

$$O_c = L(x, y, k^i\sigma) = G(x, y, k^i\sigma) * I(x, y),\ \ i = 0, .., s_u + 1$$

(4.7)

$s_u + 1$ Images must be produced in the stack of blurred images for each octave, so the final extrema detection covers a complete octave. Lowe's implementation uses $\sigma = \frac{1}{2}$

After the first octave has been created, the Gaussian image $I(x, y)$that has twice the initial value of σ (it is the second image) is down sampled by a factor of two i.e. by taking every second pixel in each row and each column. A new octave is created at half resolution. This process is repeated until the image reaches a predetermined minimum size. This process produces an octave pyramid, OP, consisting of different octaves, as seen in equation (4.8).

$$OP = \begin{bmatrix} L_1(x, y, k^i\sigma) & i = 0 \ldots., s_u + 1 \\ L_2(x, y, k^i\sigma) & i = 0 \ldots., s_u + 1 \\ & \vdots \\ L_n(x, y, k^i\sigma) & i = 0 \ldots., s_u + 1 \end{bmatrix} \qquad (4.8)$$

Adjacent images in each octave in the pyramid are subtracted to produce the DoG functions' pyramid of octaves at different scales, as seen in equation (4.9).

$$DP = \begin{bmatrix} L_1(x, y, k^{i+1}\sigma) - L_1(x, y, k^i\sigma) & i = 0 \ldots., s_u \\ L_2(x, y, k^{i+1}\sigma) - L_2(x, y, k^i\sigma) & i = 0 \ldots., s_u \\ \vdots & \\ L_n(x, y, k^{i+1}\sigma) - L_n(x, y, k^i\sigma) & i = 0 \ldots., s_u \end{bmatrix} \qquad (4.9)$$

Local extrema detection is achieved by comparing each pixel in the current image with its eight neighbours and nine neighbours in the image above and below in the current DoG octave. More specifically each pixel is compared to its eight neighbours at the scale k^i and nine neighbours at the scales k^{i-1} and k^{i+1} respectively. A pixel is selected only if it is larger than all of its neighbours or smaller than all of them.

(B) Accurate key-point localisation: In the previous section, key-point candidates were found by a process called scale-space extrema detection. In this process too many key-point candidate were produced, some of which were unstable. The next step in the algorithm is to perform a detailed fit to the nearby data for accurate location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast or poorly localised along an edge.

As in [63], where it is mentioned that the location of the extrema to a sub-pixel Laplacian. This approach uses Taylor expansion (up to quadratic terms) of the scale space function $D(x, y, \sigma)$ which is shifted so the origin is at the interest point. The Taylor expansion of $D(x, y, \sigma)$ around the detected key-point location can be seen in the equation (4.10).

$$D(z) \approx D(z_0) + \frac{\partial}{\partial z} D^T(z)\Big|_{z_0} (z - z_0) + \frac{1}{2}(z - z_0)^T \frac{\partial}{\partial z} D(z)\Big|_{z_0} (z - z_0)$$

(4.10)

The function $D(x, y, \sigma)$ and its derivatives are evaluated at the sample point $z_0 = [x_{0,}y_{0,}\sigma_0]^T$,and $\Delta z = [\Delta x, \Delta y, \Delta\sigma]^T$ is the sub-pixel offset from this point.

The approximate sub-pixel location of the exterma is computed by taking the derivative of the equation (4.10) with respect to z and setting it to zero, and then evaluated at z* which is the location at extremum.

$$\frac{\partial^2}{\partial z^2} D(z)\Big|_{z_0} z^* = \frac{\partial^2}{\partial z^2} D(z)\Big|_{z_0} z_0 - \frac{\partial}{\partial z} D(z_0) \qquad (4.11)$$

Equation (4.9) is a linear system of the form A z*=b, where $\frac{\partial^2}{\partial z^2} D(z)$ is the Hessian matrix.

[63] Suggests approximating the derivatives of $D(z)$ by using differences of neighbouring sample points.

If the final offset z*- z_0 is larger than 0.5 in any dimension. This means that the extremum lies closer to a different sample point than z_0. In that situation, the sample point is changed to the closest one, and the process of interpolation is repeated for that sample point.

[63] Mentions that locating interest points at sub-pixel accuracy is especially important close to the top of the pyramid, where the resolution of the image is very low. This is because the samples spaces at levels in the pyramid correspond to large distances relative to the base image.

(C) Eliminating unstable key-points: The extrema locations that exhibit a low function value of $|D(z)|$ are sensitive to additive noise and therefore unstable. In order to reject interest points with low contrast, the function D (z*) at extremum is evaluated. This is done by substituting equation (4.11) into equation (4.10), giving equation (4.12). In [57] all extrema values with a value of $|D(z^*)| < 0.03$ are discarded, assuming image pixel values in the range [0, 1].

$$\mathrm{D}\ (z^*) = D(z_0) + \frac{1}{2}\frac{\partial}{\partial z} D^T(z)\Big|_{z_0} (z^* - z_0) \qquad (4.12)$$

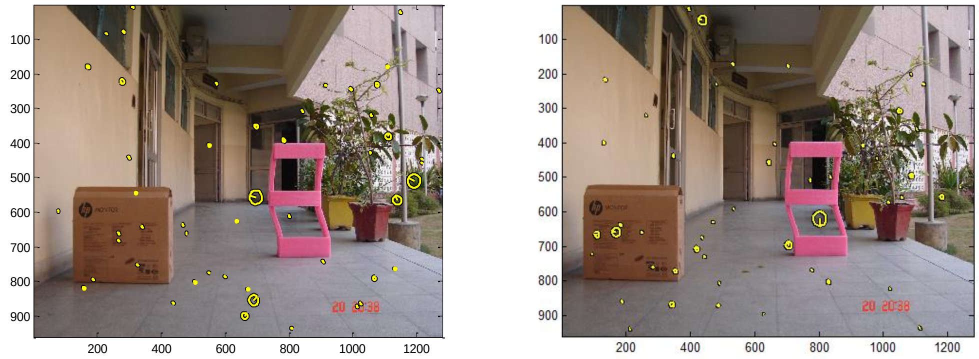
Another set of locations that may have stability problems are those located along edges of the DoG function. According to [57], the DoG function has the property that it produces strong responses along edges, even if the location along the edge is poorly determined. This property makes the locations sensitive to small amounts of noise and therefore unstable. For this reason, it is not sufficient to eliminate interest points with low contrast. In order to get better stability further precautions must be taken. A poorly defined peak in the Dog will have the property that it will have a larger principal curvature across the edge and a small one in the perpendicular direction. Finding these principal curvature amounts to solving for the eigenvalues of the second order Hessian matrix (H),

If $\lambda_1 and\, \lambda_2$ be the eigenvalues with the largest and smallest magnitude respectively. Then, defining $r_a = \lambda_1/\lambda_2$ .Thus,

$$R_a = \frac{(r_a+1)^2}{r_a}$$

(4.13)

R_a is minimum when the Eigenvalues are equal to each other. Therefore, the higher the absolute difference between the two eigenvalues, which is equivalent to a higher absolute difference between the two principal curvatures of D (z), the higher the value ofR_a. It follows that, for some threshold eigenvalue ratior_{th}, if R_a for a candidate key-point is larger than $\frac{(r_{th}+1)^2}{r_{th}}$, that key-point is poorly localised and hence rejected.



Key-points for left image Key-points for right image

Figure 4.8 Accurate key-points localisation

(D) Majority orientation assignment: This step is done by assigning a consistent orientation to each key-point. This step of the algorithm involves the scale at which the key-point was detected.

First, a gradient is defined which provides two pieces of information, magnitude and direction; it is natural to encode this information in a vector. The length of this vector provides the magnitude of the gradient, while its direction gives the gradient direction. Because the gradient may be different at every location, it is represented with a different vector at every image location.

The gradient magnitude m_g and the gradient orientation $\theta_d$of a Gaussian-smoothed image $L(x, y, \sigma)$ at the key-point's scale σ are computed as follows:

$$m_g(x, y) = |L(x, y, \sigma)| = \sqrt{\left(\frac{\partial L(x,y,\sigma)}{\partial x}\right)^2 + \left(\frac{\partial L(x,y,\sigma)}{\partial y}\right)^2} \tag{4.14}$$

$$\theta_d = \angle L(x, y, \sigma) = tan^{-1}\left(\frac{\partial L(x,y,\sigma)/\partial y}{\partial L(x,y,\sigma)/\partial x}\right) \tag{4.15}$$

Second, an orientation histogram is described in the following:

Histogram; from the gradient orientations of the sample points around a key-point, an orientation histogram is formed.

Quantisation; the orientation histogram is quantised into an even-sized bin N_b to reduce the data that has to be processed. A higher value of bins results in a higher resolution, but also a higher computation load, [57] suggests $N_b = 36$.

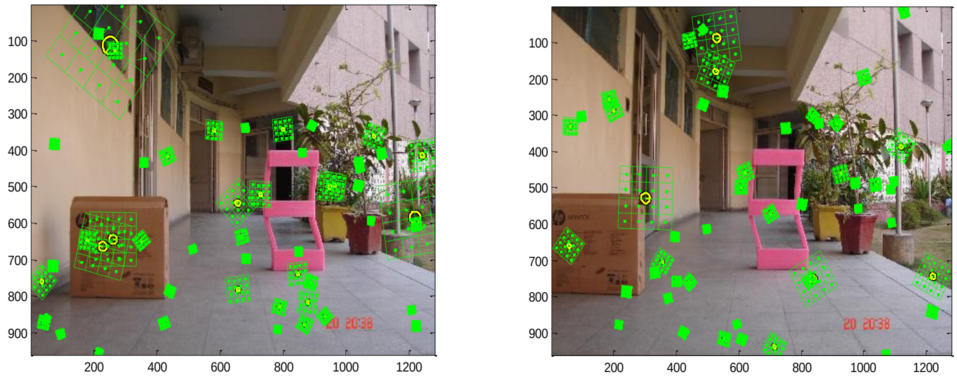
Weighted; each sample is weighted by a gradient magnitude and a Gaussian-weighted circular kernel that is placed on the centre of the sample point, with $\sigma,$ 1.5 times the scale of the interest point. This has the effect of giving higher weight to the samples near the centre of the window.

Creation of new interest points; the peaks in the orientation histogram correspond to dominant directions of the local gradients surrounding the interest point. The highest peak in the histogram is detected and if there is any other peak which falls within 80% of the highest peak, a new interest point is created with that orientation. So, there will be multiple key-points with the same location and scale, but different orientations. To obtain a better precision, a parabola is fitted to the three histogram values closest to each peak, in order to interpolate the peak position.

(E) Computation of the local image descriptor: Previous steps found key point locations at particular scales and assigned orientations to them. This ensured invariance to image location, scale and rotation. The next step is to compute descriptor vectors for these key-points such that the descriptors are highly distinctive and partially invariant to the remaining variations, like illumination, 3D viewpoint, etc. This step is pretty similar to the orientation assignment step.

An interest point descriptor is created by first sampling the gradients magnitudes $m_g(x, y)$ and orientation $\theta_d(x, y)$ in the surroundings of the key-point location. The gradient magnitude and orientation are computed using equation (4.14) and (4.15). Then, a rotation matrix is applied to the descriptor in order to get orientation invariance. Afterwards, the key-point descriptor is weighted and boundary effects are reduced. Finally the SIFT-descriptor is created. The effect of change in image contrast and nonlinear illumination can affect the SIFT-key-point descriptor.

A SIFT-descriptor vector is defined as a vector with 128 elements that uniquely identifies a key-point. The key-point descriptor is arranged into 16 histograms. Each histogram shows eight direction bins. Each bin in each histogram is formed by adding up the gradient magnitudes in the current direction. The value of the orientation histogram constitutes the 128-dimentional vector (8 orientations X 16 histograms). The 128 dimensional vector is the SIFT feature vector or SIFT descriptor as shown in fig. (4.9).



SIFT-descriptor for left image SIFT-descriptor for right image

Figure 4.9 SIFT- descriptor

4.4.2.2 SIFT-Descriptor Matching

Once the descriptors from the left and right images are found, the next step is to find the best candidate match for each key-point in the left image which corresponds with the right image.

[57] Suggests a method for comparing the descriptors. The method consists of looking at the relation between the two shortest Euclidean distances, as seen in equation (4.16).

If the ratio (r_{tt}) between the second-shortest (d_{scn}) and the shortest (d_{cn}) Euclidean distances is close to 1, it implies that two key-points in the reference set match the image key point equally well. This ambiguous match significantly reduces the probability that the match is correct.

This ratio is used to get rid of incorrect matches. It is done by setting a proper threshold(t_s) for this ratio, thus all matches above this threshold are considered false matches (f_m) and are rejected, and all matches below this threshold are considered correct matches (*cm*) and are accepted. That threshold has been determined empirically to 0.8. It is, of course, inevitable that this procedure will discard some of the correct matches, as it is stated in equation (4.17).

$$r_{tt} = \frac{d_{cn}}{d_{scn}}$$

(4.16)

Correct matches if $r_{tt} < t_s$

False matches if $r_{tt} \geq t_s$ (4.17)

The Euclidean distance d between two descriptors is computed as stated in equation (4.18)

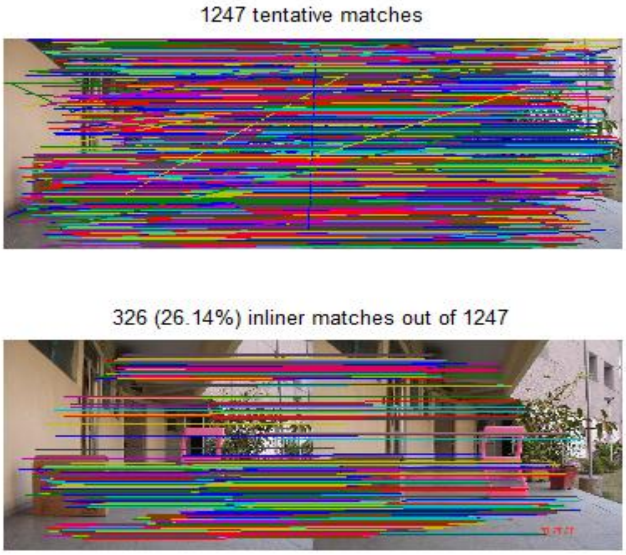
$$d = \sqrt{\sum_{i=1}^{128} \left(p_{1_i} - p_{2_i}\right)^2}$$

(4.18)

4.4.2.3 RANSAC (RANdom SAmple Consensus)

The RANSAC algorithm is proposed by Fischler and Bolles [64] is a general parameter estimation approach designed to cope with large proportion of outliers in the input data. It is a resampling technique that generates candidate solutions by using minimum number of observations (data points) required to estimate the underlying model parameters.

After the feature matching pairs are determined, we apply RANSAC to estimate the homography. The RANSAC algorithm is being used to perform this computation under the possible presence of outlier feature matches. Fig. 4.10 shows the results of matching interest points in an image stereo pair, tentative matches are the total matches determined by matching SIFT descriptor and inlier matches are the more accurate matches generated after applying RANSAC.



1247 tentative matches

326 (26.14%) inliner matches out of 1247

Figure 4.10 Descriptor matches are connected by coloured lines

4.4.2.4 Optimization of Threshold Ratio

Selection of threshold ratio for matching the descriptors is an important step while applying SIFT algorithm. Because if the selection of the threshold ratio is less than a certain limit then there is a large number of tentative matches found and that will lead to increase in the possibility of wrong matches. And if the selection of threshold ratio is more than certain limit then the number of tentative matches are very less. So there is a possibility to miss some objects in the environment.

Table 4.2 Variation in matching results with threshold ratio

Threshold value	Tentative matches	In liner matches	% of In liner matches
0.10	1247	326	26.14
0.20	972	295	30.35
0.30	716	235	32.82
0.40	551	196	35.57
0.50	460	172	37.39
0.60	390	160	41.03
0.70	355	149	41.69
0.80	323	147	45.51
0.85	308	145	47.07
0.90	298	135	45.30
0.95	106	21	20.1

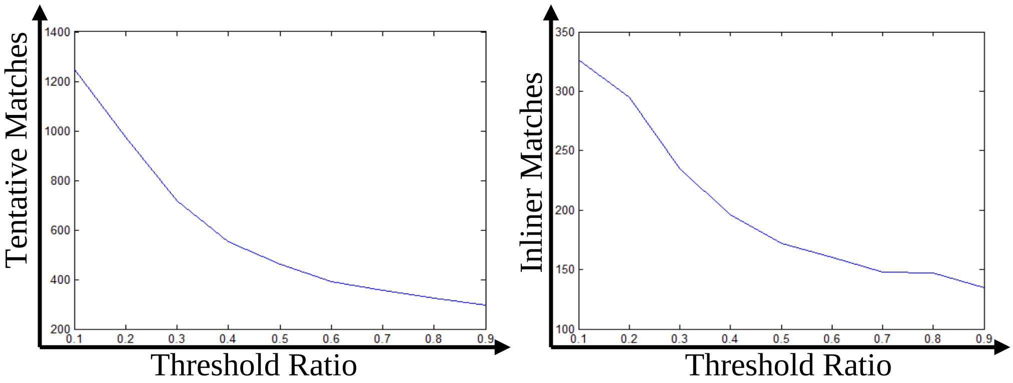
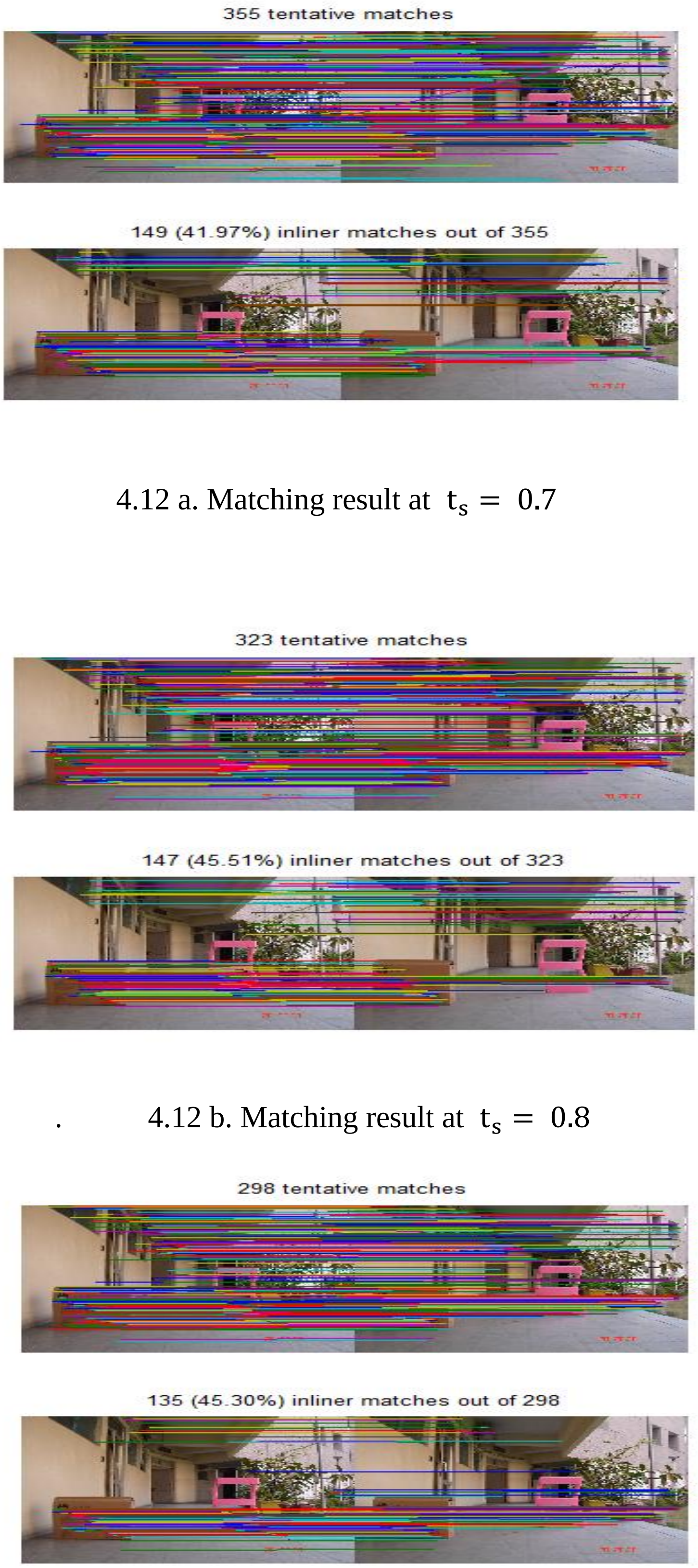


Figure 4.11 Variation of tentative matches and inlier matches with t_s


355 tentative matches
149 (41.97%) inliner matches out of 355
323 tentative matches
147 (45.51%) inliner matches out of 323
298 tentative matches
135 (45.30%) inliner matches out of 298

4.12 a. Matching result at $\mathrm{t_s} = 0.7$

. 4.12 b. Matching result at $\mathrm{t_s} = 0.8$

4.12 c. matching result at $\mathrm{t_s} = 0.9$

Figure 4.12 Descriptor matching with different t_s

For, selection of optimal value of threshold ratio we applied RANSAC after applying SIFT with varying the value of threshold ratio. After this experiment we found that the number of tentative matches given by SIFT decreases and percentage of inlier matches given by RANSAC increases by increasing the value of threshold ratio up to $t_s = 8.5$ but when we further increased the value of t_s , the percentage of inlier matches decreased rapidly. Some results of tentative matches and inlier matches are shown in Fig.4.12, and complete experimental results are given in table 4.2. Variation of tentative matches and inlier matches are shown in Fig.4.11.

By this experiment we conclude that optimal value of threshold ratio t_s can be selected by optimization of percentage of inlier matches.

4.4.2.5 Distance measurement

Distance measurement method used in this work was suggested by Hai-Sung Baek [65]. The whole stereo vision system area consists of three areas: area *A* between the two optical axes, area *B* outside the two optical axes, and a block section to be seen by only one camera which consist of area *RC* which is seen by the right camera only and area *LC* which is seen by the left camera only. Other proposed ways to measure distance to an object by SVS is only applicable if object is in area *A*.

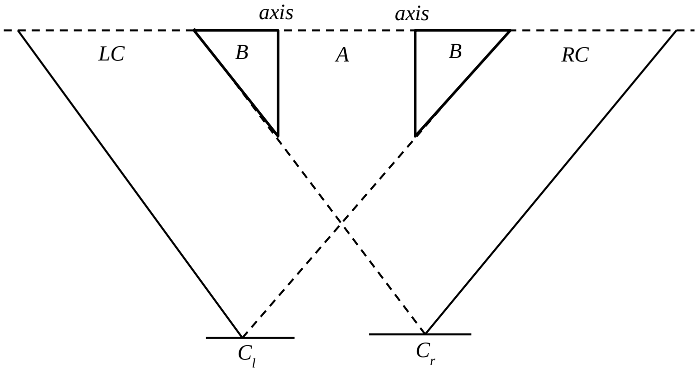


Figure 4.13 Stereo Vision Systems area

Distance measurement method proposed in [65] can be used when the object is located even in the outside area of the two optical axis as long as the object is in overlapping area of the two cameras. The Fig. 4.14 is picturing that object expresses case in outside area of optical axis P_l and P_r of Fig. 4.14 amounts to negative number and D_1, D_2 is calculated in equation (4.19) and (4.20).

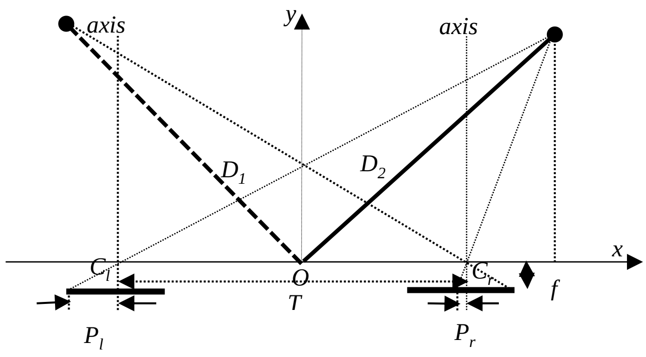


Figure 4.14 Object beyond the optical axis of right and left camera

$$D_1 = \frac{T}{2(P_l - P_r)}\sqrt{(P_l + P_r)^2 + 4f^2}$$

(4.19)

$$D_2 = \frac{T}{2(P_r - P_l)}\sqrt{(P_l + P_r)^2 + 4f^2}$$

(4.20)

4.4.2.6 Simulation Results

The simulation of these systems is all performed in MATLAB and SIMULINK. To find the correspondence relation between left and right images SIFT and RANSAC has been implemented by using VLfeat toolbox in MATLAB developed by Vedalid at UCLA [61]. This gives the disparity value in a MAT file and the MAT file is being set as the input to the Simulink programme as suggested by Hai-Sung Baek [65] for the distance measurement. Output of this Simulink programme has been saved in workspace that gives the distance of each point whose descriptors matches. And finally we compared the distance given by the system for the point around each objects with actual measured distance of the objects. Table 4.3 gives the specific experimental data.

Table 4.3 Measured distances by using SIFT

The actual distance (m)	The measured distance (m)	The error (%)
1.9	1.956	2.94
2.8	2.91	3.92
3.12	3.07	1.60
3.5	3.42	2.28

3.62	3.73	3.03

4.4.3 SIFT feature model

4.4.3.1 Experimental setup

The experiment is performed for the mapping of an indoor environment using SIFT feature descriptor model. Four objects have been placed on the path that mobile robot has to follow. Stereo pair of images has been taken by using single CCD camera, Sony cyber shot (DSC-S650). The camera is placed on a slider at two different position separated by a distance of 5cm keeping the axis of the cameras parallel. The fig. 4.15 shows the geometric map of the experimental setup. Whole environment has been mapped by using an occupancy grid of 100x100 cells, each cell having size of 8cmx8cm.

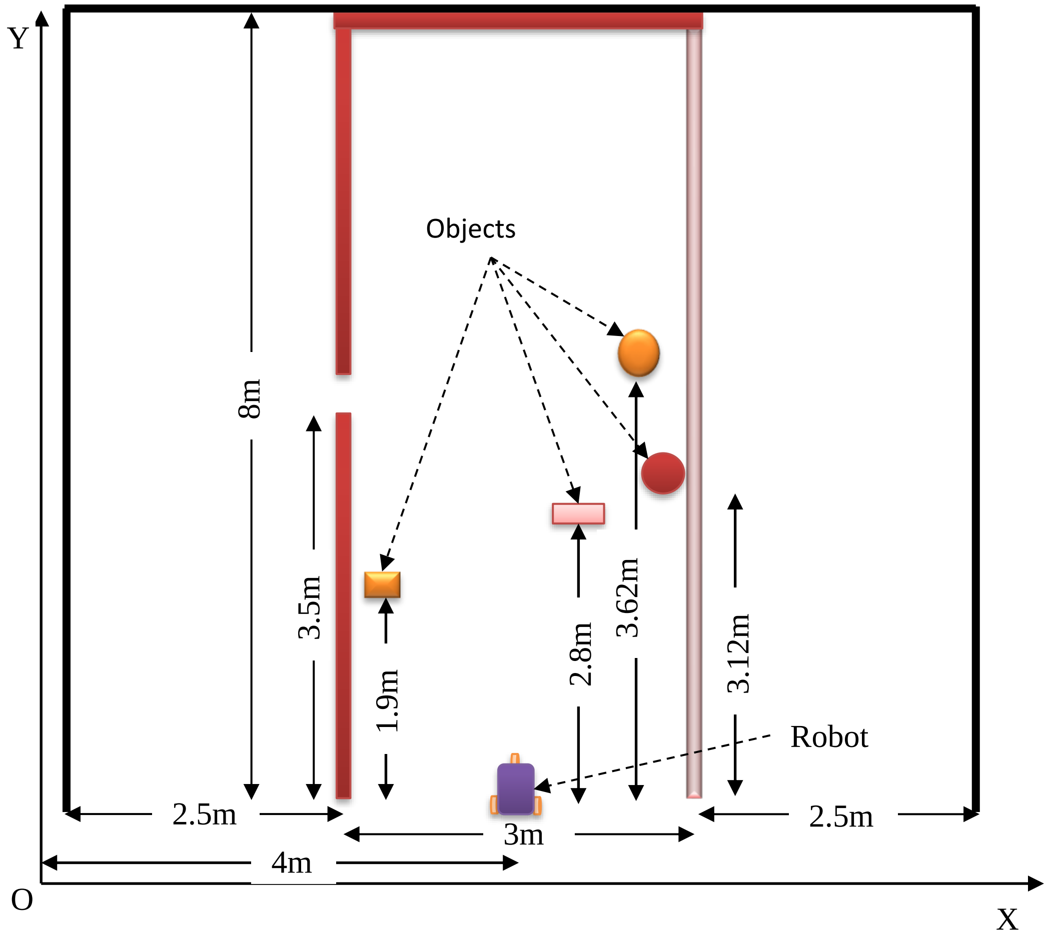


Figure 4.15 Geometric map of the dynamic environment

Position of robot from where stereo pair of images has been taken is (50, 1) and the positions objects are (34, 23), (66, 35), (69, 39) and (68, 45).

4.4.3.2 SIFT- Feature Model

One of the tasks of a stereo vision system in mobile robot is the extraction of features from a pair of digital images. These features can be used for map making. Feature based mapping has been implemented in the past by researchers. [66] For instance used the canny edge detector to extract edges from the environment, matching edges in the pair of digital images that are used to construct the map. [70] Suggests the use of visual techniques such as the SIFT algorithm to identify features common to each camera view, and thereby infer the geometry.

Sensor readings are uncertain by nature and the stereo vision system is not the exception. In this chapter a probabilistic sensor model is proposed. The model takes into account the uncertainties inherited by the stereo vision system readings. The model is divided in two areas; the occupied area which is in fact the model of the quantification error and the modelling of empty area. Combination of the two models into one gives the final SIFT-descriptor model.

4.4.3.3 Modelling the Occupied Area

Stereo triangulation is needed in order to get the depth from the stereo system to the features extracted from the SIFT. And, due to the factors of quantification and calibration errors, a certain degree of uncertainty must be expected in the triangulation. This uncertainty must be modelled. In [66], Mathies and Shafer show three approaches to model such uncertainty. They are: discrete tolerance limits, scalar weights and multidimensional probability distribution.

The multidimensional probability distribution will be explained in the following. The geometry of stereo triangulation is shown in figure 4.16(a). The trick marks on the image planes denote pixel boundaries as well as the lines which are radiating to the space from this tick marks. Assuming a point M in the space which is projected onto the left and right image planes at $x_l$and $x_r$respectively. Because of errors in quantification, the stereo system will determine $x_l$and x_r with some error, which in turn causes error in the estimated location of M. Thus causing the point M lying on the region surrounds the true location. It can also be seen in figure 4.16(a) that the uncertainty is skewed and orientated (diamond around the point M). This uncertainty can be captured by using 3D Gaussian distributions. This distribution can be depicted as in figure 4.16(b), where the ellipse represents the contour of the error model and the diamond represents the quantisation error.

The mean values and the covariance matrix of each point in the space are needed in order to get a proper shape of the Gaussian distribution. For that, the triangulation error for the general case of 3D points projecting onto 2D images is described in the following.

Consider 3D point M=(X, Y, Z) or a vector= $[X, Y, Z]^T$, which is projected onto the left and right image planes respectively as $m_l = [x_l, y_l]^T$ and $m_r = [x_r, y_r]^T$ as depicted in figure 4.17.

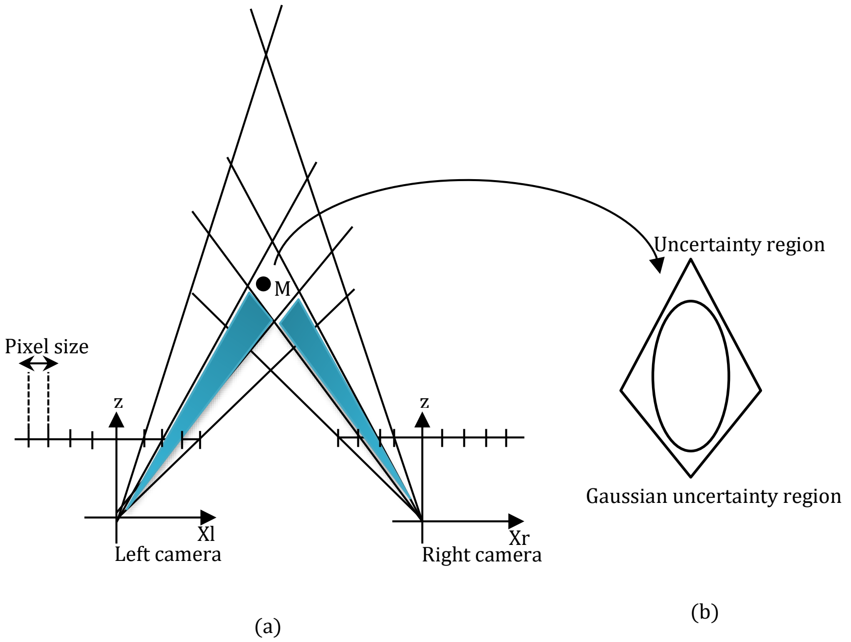


Figure 4.16 (a) Stereo Geometry showing triangulation uncertainty as a diamond around a point *M*. It also shows the empty region uncertainty from the pair of cameras to the uncertainty region of the point *M*. (b) 2D Gaussian distribution uncertainty region.

The vectors m_l and m_r are considered to be normally distributed with means μ_l and μ_r and covariance matrices V_l and V_r [66]. This mean and the covariance matrix of the point *M* are μ_M and V_M. μ_M and *M* are functions of m_l and m_r meaning that $\mu_M = f(x_l, y_l, x_r, y_r)$ and $M = [X, Y, Z]^T = f(x_l, y_l, x_r, y_r)$ as stated in [66]. Equation 2.6 and 2.7 from the triangulation process explained in chapter2 can be used to estimate the coordinates of the vectors$M = [X, Y, Z]^T$. These equations need to be changed in order to use them in the estimation of the point *M*. these equations deviate from the one used in this section in that, MATLAB calibration tool box uses the right camera as a reference frame instead of the left reference frame as stated in the chapter2. Equation 2.6 and 2.7 get the shape stated in equation 4.21 and 4.22.

$$s_M = am_r + \frac{1}{2}cn \qquad (4.21)$$

$$am_r - bR^T m_l + c(m_r \times R^T m_l) = T$$

$$[m_r - R^T m_l \quad m_l + c(m_r \times R^T m_l)] \begin{bmatrix} a \\ b \\ c \end{bmatrix} = T$$

(4.22)

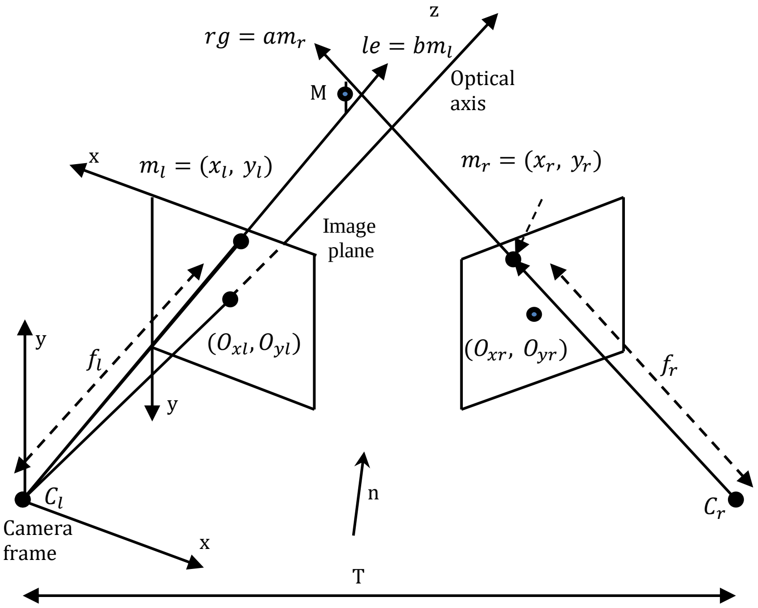


Figure 4.17 Figure shows in the left part the perspective or pinhole model and also the triangulation.

The vectors s_M in equation 4.21 defines the coordinates of the point M as seen in equation 4.23, where $n = m_r \times R^T m_l$.

$$M = [X, Y, Z]^T = am_r + \frac{1}{2}c(m_r \times R^T m_l)$$

(4.23)

Making the following definitions $m_l(x_l) = m_{lx}$, $m_l(y_l) = m_{ly}$, $m_r(x_r) = m_{rx}$, and $m_r(y_r) = m_{ry}$. The vectors m_l and m_r can be defined as vectors with 3 components each one such as $[m_{lx}, m_{ly}, f_l]^T$ and $[m_{rx}, m_{ry}, f_r]^T$ respectively, where f_l and f_r are the focal length of the left and right cameras respectively. $m_{lx} = (O_{xl} - x_l)S_x$, $m_{ly} = (O_{yl} - y_l)S_y$, $m_{rx} = (O_{xr} - x_r)S_x$, and $m_{lx} = (O_{yr} - y_r)S_y$, where S_x is the pixel size in mm in horizontal and vertical direction respectively. O_{xl} , O_{yl}, O_{xr}, and O_{yr} are the coordinates in pixels of the left and right image centres (The principal point). R is a (3x3) rotation matrix.

A proper substitution of the definition mentioned above into equation 4.23 and expanding this equation into individual equations gives equations stated in 2.25.

$$[X, Y, Z]^T = a\left[m_{rx}, m_{ry}, f_r\right]^T + \frac{1}{2}c\left(\left[m_{rx}, m_{ry}, f_r\right]^T \times R^T\left[m_{lx}, m_{ly}, f_l\right]^T\right) \qquad (4.24)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = a\begin{bmatrix} m_{rx} \\ m_{ry} \\ f_r \end{bmatrix} + \frac{1}{2}c\begin{bmatrix} m_{ry}R^T f_l - R^T m_{ly} f_r \\ m_{rx}R^T f_l - R^T m_{lx} f_r \\ m_{rx}R^T m_{ly} - R^T m_{lx} m_{ry} \end{bmatrix} \qquad X =$$

$$a(O_{xr} - y_r)S_x + \frac{1}{2}c\left[(O_{yr} - y_r)S_y R^T f_l - R^T\left(O_{yl} - y_l\right)S_y f_r\right] = f_1(x_l, y_l, x_r, y_r)$$

$$Y = a\left(O_{yr} - y_r\right)S_y + \frac{1}{2}c[-(O_{xr} - x_r)S_x R^T f_l + R^T(O_{xl} - y_l)S_x f_r] = f_1(x_l, y_l, x_r, y_r)$$

$$Z = af_r + \frac{1}{2}c\left[(O_{xr} - x_r)S_x R^T\left(O_{yl} - y_l\right)S_y - R^T(O_{xl} - x_l)S_x\left(O_{yr} - y_r\right)S_y\right]$$

$$= f_3(x_l, y_l, x_r, y_r) \qquad (4.25)$$

The parameters *a* and *c* in equations stated in 4.25 can be obtained by solving the linear system as seen in equation 4.22 where *a, b* and *c* are scalars. This linear system is defined as follows $(A_x = b_l)$, where $A = [m_r, -R^T m_r, m_l \times R^T m_r]$, $x = [a, b, c]^T$ and$b_l = T$.

The mean value μ_M and the covariance matrix V_M of the point *M* can be computed as follows. V_M is calculated as stated in [66]as seen equation 4.26

$$V_M = J\begin{bmatrix} V_l & 0 \\ 0 & V_r \end{bmatrix} J^T \qquad (4.26)$$

Where J is the Jacobian of first partial derivatives of $f_{1n}(x_l, y_l, x_r, y_r)$ with n= 1, 2, 3 as seen in expression 4.25. The jacobian, which is a matrix of all first-order partial derivatives of a vector-valued function, can be computed as in equation 4.27

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_l} & \frac{\partial f_1}{\partial y_l} & \frac{\partial f_1}{\partial x_r} & \frac{\partial f_1}{\partial y_r} \\ \frac{\partial f_2}{\partial x_l} & \frac{\partial f_2}{\partial y_l} & \frac{\partial f_2}{\partial x_r} & \frac{\partial f_2}{\partial y_r} \\ \frac{\partial f_3}{\partial x_l} & \frac{\partial f_3}{\partial y_l} & \frac{\partial f_3}{\partial x_r} & \frac{\partial f_3}{\partial y_r} \end{bmatrix}$$

(4.27)

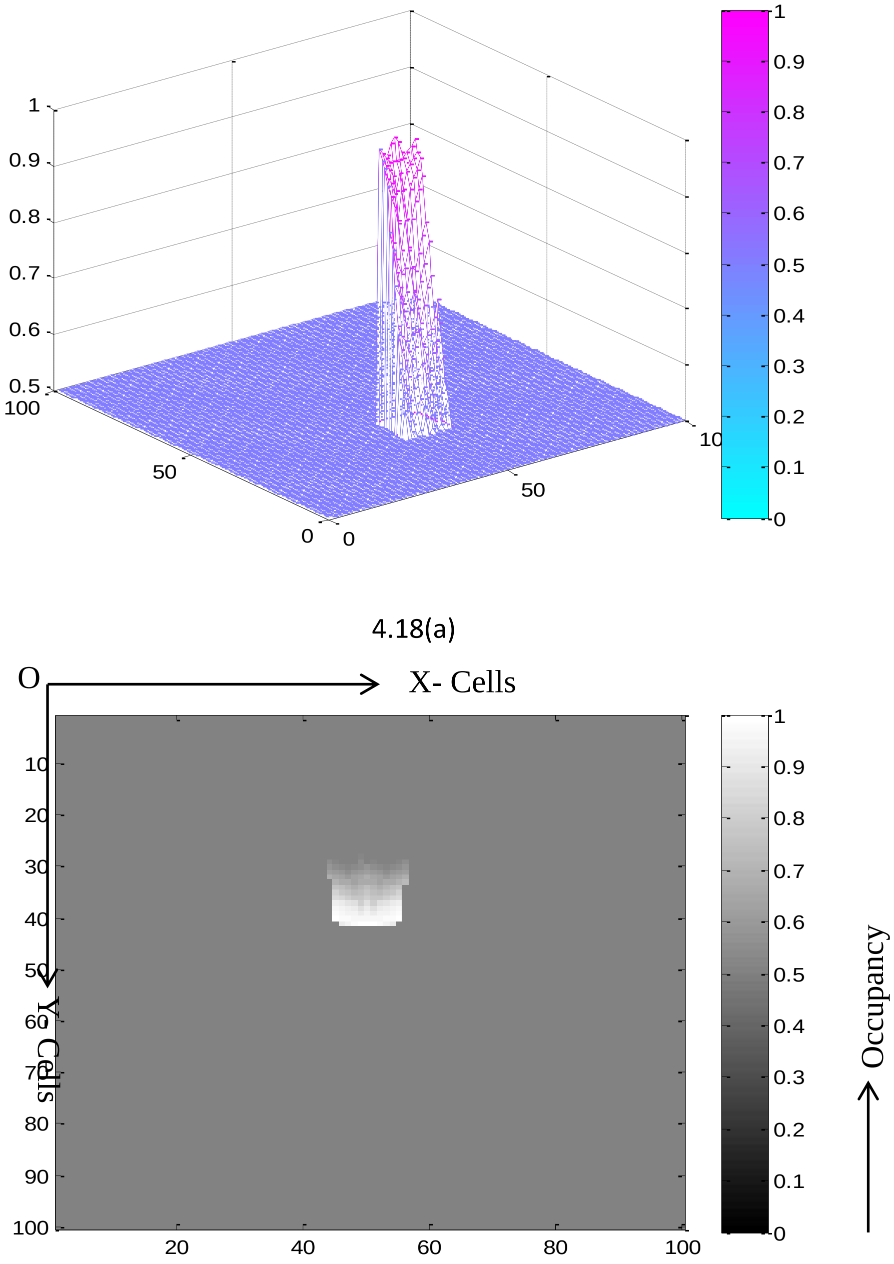
The Jacobian matrix stated in 4.27 can be solved by means of numerical analysis, ans an example of such numerical approach is shown in equation 4.28.

$$\frac{\partial f}{\partial x_1} \approx \frac{1}{2}(f(x_l - 1, y_l, x_r, y_r) - f(x_l + 1, y_l, x_r, y_r)) \qquad (4.28)$$

Mathies and Shafer [66] suggest to approximate the means (μ_l and μ_r) with the coordinates returned by the stereo matcher and the covariances with the identity matrices. This is equivalent to treating the image coordinates as uncorrelated with variances of one pixel.

Fig. 4.18 (a) shows a three dimensional probabilistic model of the area by the SIFT – descriptor. The colour- bar indicates the lowest and the highest probabilities of being occupied. Fig. 4.18 () is a two gray-scale dimensional representation of the occupied area. The

0 in the gray-scale bar indicates the lowest probability of occupancy, e.g. $P^{o}_{i,j} = 0$.Whereas the 1 indicates the highest probability of occupancy, e.g. $P^{o}_{i,j} = 1$.



4.18(a)

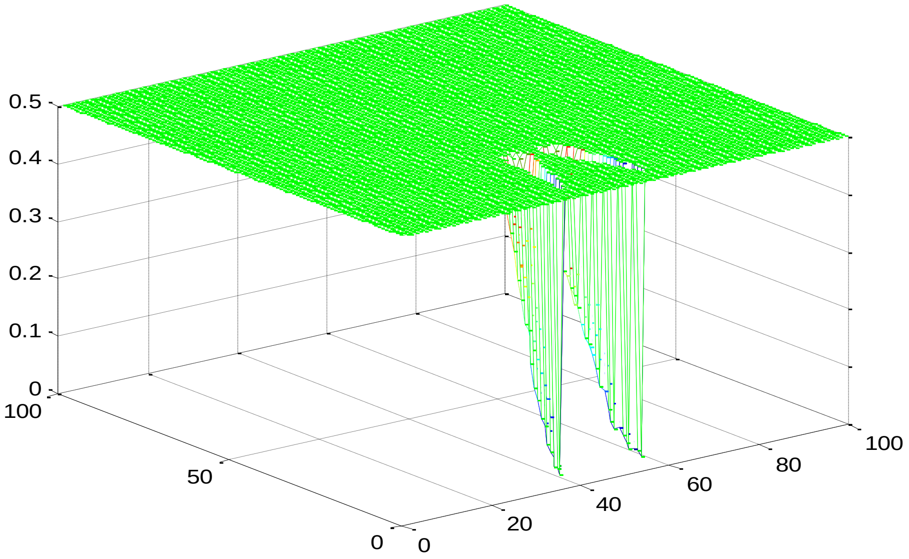
4.18(b)

Figure 4.18 (a) is a 3D representation of the occupied area by the SIFT-descriptor. (b) Is a 2D representation of the occupied area by the SIFT- descriptor.

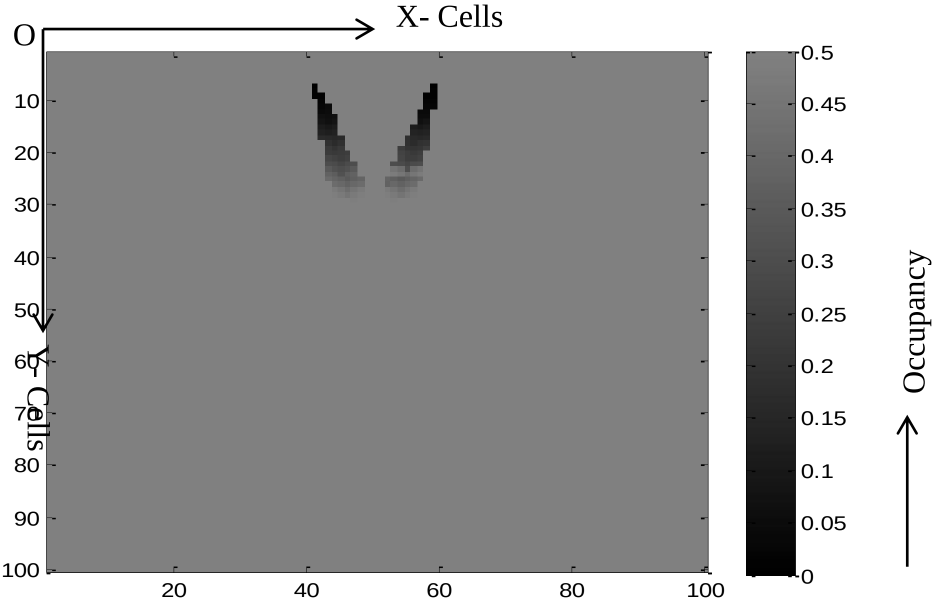
4.4.3.2 Modelling the Empty Area

The empty regions from the left and right cameras, as shown as shadow areas in fig. 4.16 (a), also need to be modelled. A search concerning the modelling of the uncertainties of these two regions has been carried out in the literature with unsuccessful results.

The approach taken by Elfes [2] (equation 2.19 and 2.20) to model the empty region of the sonar beam has been taken into consideration to solve the problem of modelling these two empty regions. The implementation came up with satisfactory results as depicted in fig. 4.19. Fig. 4.19 (a) shows the three dimensional probabilistic model of the two empty areas of the SIFT- descriptor, whereas fig. 4.19 (b) shows a two dimensional grey scale representation of the empty area, the more black the region is the more probability of emptiness.



4.19(a)



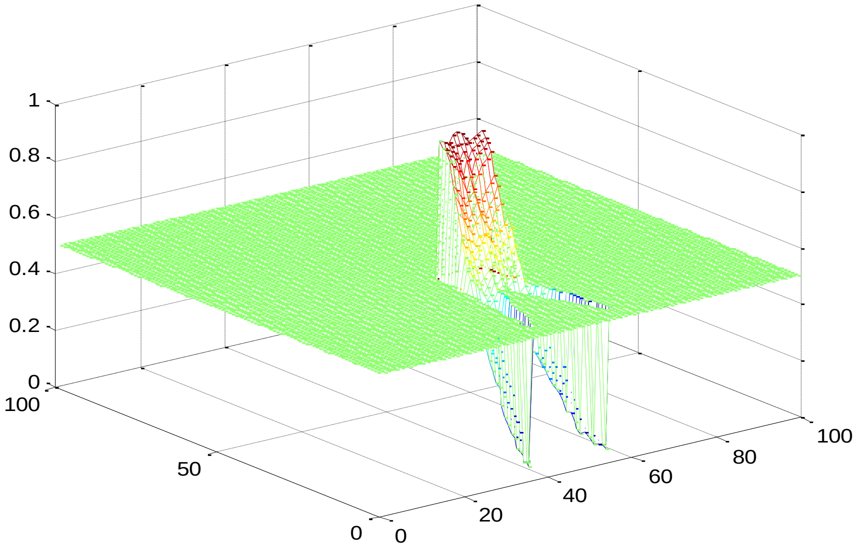
4.19(b)

Figure 4.19 (a) A 3D probabilistic representation of the empty area of the SIFT- descriptor.

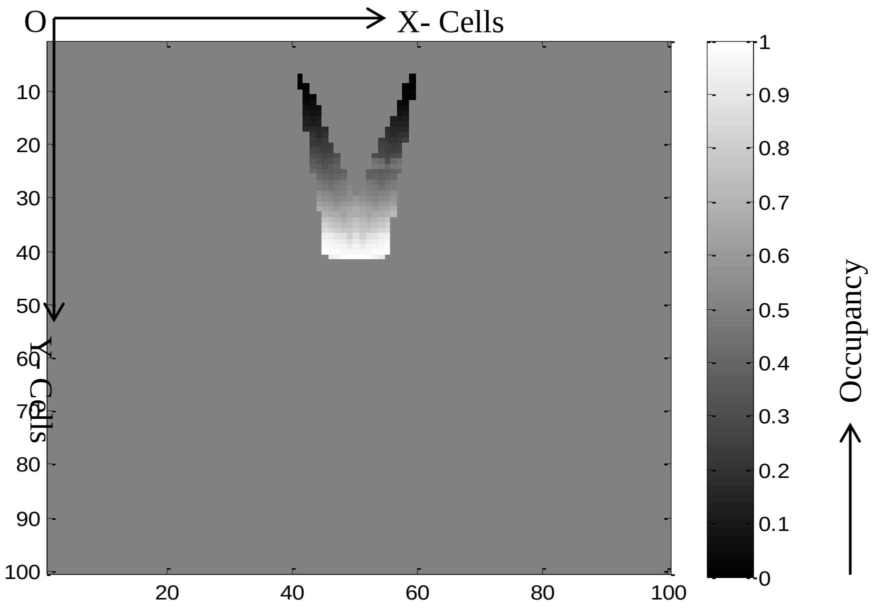
(b) A 2D grey- scale representation of the empty area of the SIFT- descriptor.

4.4.3.3 Experimental results

The occupied area as well as the empty one of the SIFT-descriptor model is put together bringing up the final probabilistic SIFT- descriptor model. Fig. 4.20 (a) shows a 3D model of the uncertainty triangulation together with the empty uncertainty region of the empty areas, which in fact is the 3D probabilistic model of the SIFT descriptor. Fig. 4.20 (b) shows a 2D view of the former 3D view of the uncertainties.



4.20(a)



4.20(b)

Figure 4.20 The SIFT- descriptor probability model. (a) 3D view of the SIFT- descriptor probability model. (b) 2D view of the SIFT- descriptor probability model.

4.4.4 Integrating the occupancy grid

4.4.4.1 Recursive Bayes Update Rule

The attraction of the Bayesian inference approach to map building stems from the fact that Bayes updating rule is recursive. When it is used to support sensor fusion, Bayes rule provides a way of computing a posteriori probability of a hypothesis being true giving supporting of evidence. [68], [49], [66], and [3] have successfully used Bayes rule to update the occupancy grid for multiple sensor readings$(s_1, \dots \dots, s_n)$.

Equation 4.29 and 4.30 are obtained when Bayes rule is transferred to the occupancy grid framework for multiple sensor readings.

$$P_{i,j}^{o|s} = \frac{P_{i,j}^{s|o}\ P_{i.j}^{o}}{P_{i,j}^{s|o}\ P_{i.j}^{o} + (1 - P_{i,j}^{s|o})(1 - P_{i,j}^{o})}$$

(4.29)

$$P_{i,j}^{e|s} = \frac{P_{i,j}^{s|e}\ P_{i.j}^{e}}{P_{i,j}^{s|e}\ P_{i.j}^{e} + (1 - P_{i,j}^{s|e})(1 - P_{i,j}^{e})}$$

(4.30)

The following statements are defined.

- The relevent evidence A is given by the sensor reading s.

- The certainty of the true parameter B_i is given by $P^o_{i.j}$ and $P^e_{i.j}$, meaning that they are the prior probabilities of the cell $C_{i,j}$ being occupied or empty. They are taken from the existing map
- The conditional probability $P(B_i|A)$ is given by $P^{s|o}_{i,j}$ and $P^{s|e}_{i,j}$, which are the conditional probabilities that a sensor reading will exist given the state of the cell $C_{i,j}$, being occupied or empty. This conditional probability is given by the probabilistic sensor model.
- The condotional probability $P(B_i|A)$ is given by $P^{o|s}_{i,j}$ and $P^{e|s}_{i,j}$, which is the conditional probability that a cell is occupied based on the past sensor readings. It is the new estimate.

A new sensor reading, introduces additional informational information about the state of the cell $C_{i,j}$. This information is done by the sensor model $P^{s|o}_{i,j}$, and it is combined with most recent probability estimate stored in the cell. This combination is done by the recursive Bayes'rule ($P^{o|s}_{i,j}$) based on the current set of readings to give a new estimate$P^{o|s}_{i,j}$. It is worth nothing that when initialising the map an equal probability to each call $C_{i,j}$ must be assigned. In other words, the initial map cell prior probabilities are $P^o_{i.j}$=$P^e_{i.j} = \frac{1}{2}$.

A sensor reading s in equations 4.29 and 4.30 changes to r when there is a sensor reading and to v when there is a vision reading, e.g. $s \rightarrow r$ and $s \rightarrow v$.

The graphical interpretation of equations 4.29 and 4.30 for a signal cell $C_{i,j}$ within a sonar model can be depicted as in fig. 4.21. In order to update the cell $C_{i,j}$, with equations 4.29 and 4.30, a prior probabilities $P^{o_r}_{i.j}$ and $P^{e_r}_{i.j}$ given by the existing grid are needed. These probabilities are combined with the probabilities given by the sonar model ($P^{r|o}_{i,j}$ and$P^{r|e}_{i,j}$) by means of Bayes'rule of combination to give a new estimate or posteriori probabilities ($P^{o|r}_{i,j}$ and $P^{e|r}_{i,j}$)

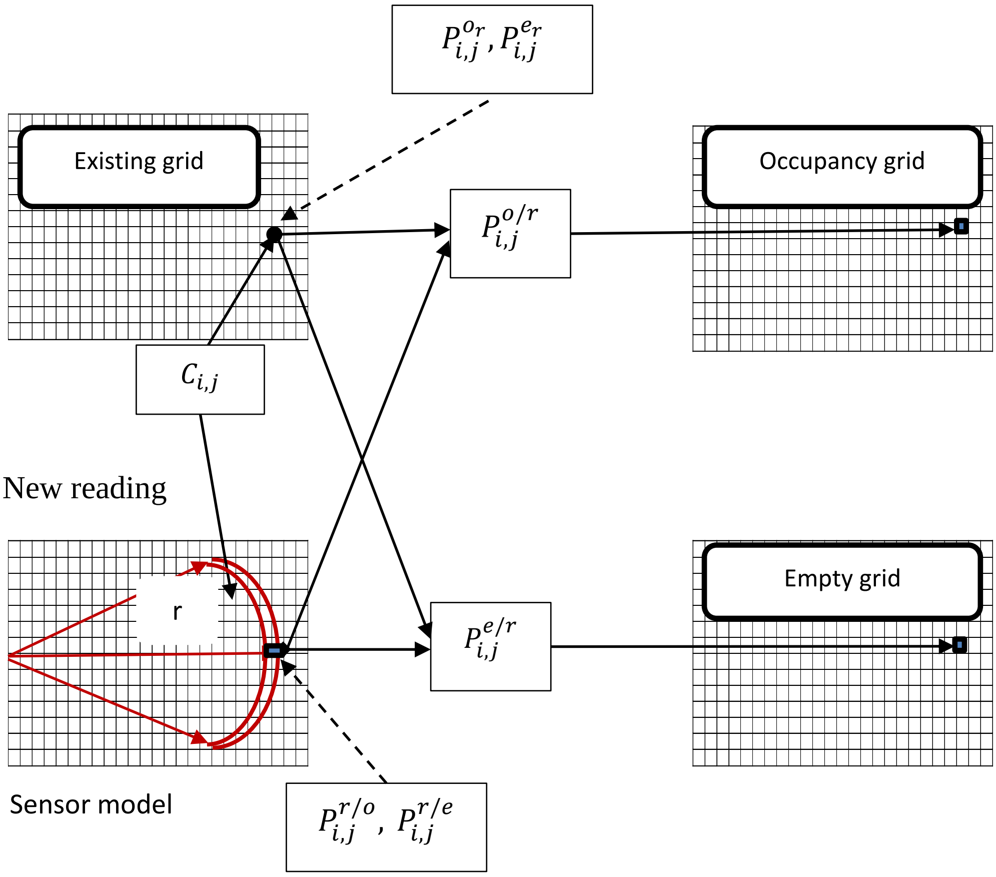


Figure 4.21 a graphical interpretation of equation 4.1 and 4.2 for a single cell $C_{i,j}$within the main lobe of the sonar model. An existing grid contains the old probabilities (prior) of the sonar model $P_{i.j}^{o_r}$ and $P_{i.j}^{e_r}$ of a single cell $C_{i,j}$ being occupied or empty respectively. A new sensor data interpreted by a sonar model ($P_{i,j}^{r|o}$, $P_{i,j}^{r|e}$) is used together with the existing probabilities in the grid to estimate the new state of the cell ($P_{i,j}^{o|r}$, $P_{i,j}^{e|r}$).

4.4.4.2 Experimental results

The occupancy grid obtained from the SIFT descriptor model is updated by using Recursive Bays Rule. For integration of the occupancy grid obtained from the SIFT descriptor model we have used the occupancy grid of the prior information about the static environment as an existing grid. The occupancy grid obtained from the SIFT descriptor model is updated with existing occupancy grid by using Recursive Bays rule. The integrated occupancy grid map of the indoor environment is shown in figure 4.22.

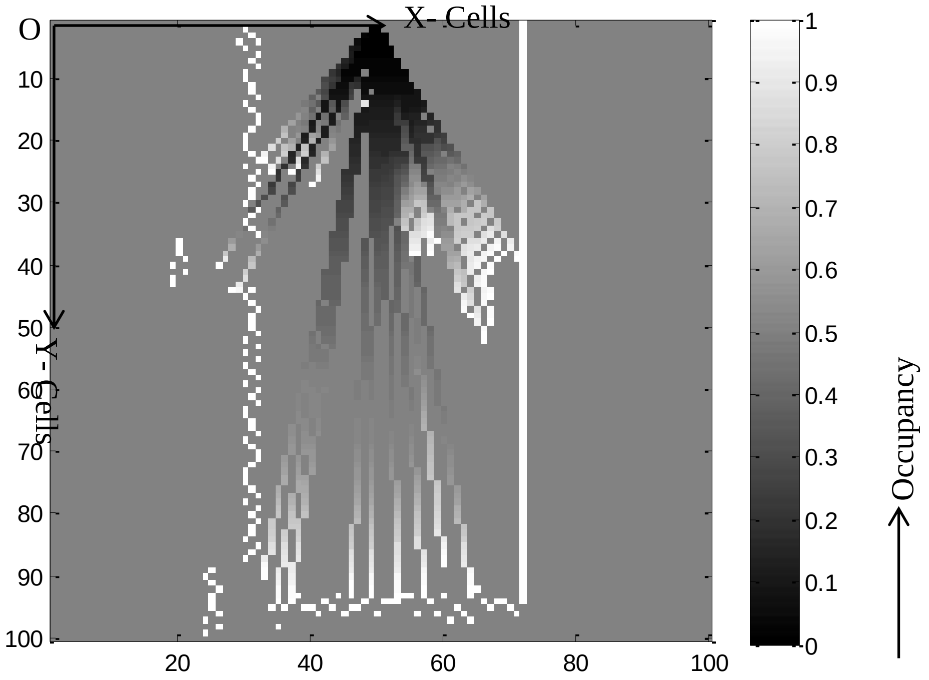


Figure 4.22 Integrated occupancy grid map of the environment

White cells represents occupied cell and the scale on the side bar shows the probabilistic value of occupancy according to the color of a particular cell. The more the color is black of a cell the less is the occupancy of the cell.

4.5 Conclusion

In this chapter, we have studied the distance measurement system by using Scale Invariant Feature Transform. Analyzed the key algorithm, brought out the image match method based on SIFT matching and introduced an empirical technique to optimize the system performance. And implemented the discussed algorithm for a distance measurement method to measure the distance of an object which is located outside the axes of stereo cameras, obtained the ideal results. In the following system, distance to an object in any location of overlapping area of stereo cameras can be measured. It is expected that the system will be widely applied in engineering measurement and automatic control.

This chapter has also proposed a SIFT-descriptor probabilistic sensor model. The model takes into account the uncertainty inherited in the stereo vision system when computing the SIFT-descriptors. To this end, the model has been divided in two regions; occupied and empty areas. The occupied area was modeled according to [66], where the multidimensional probability distribution approach is used to model the quantification error. The empty region was modeled according to the approach proposed in [1] for modeling the empty area of the sonar beam. The modeling came up with satisfactory results. The model has been used for internal representation of sensor fusion tasks. The occupancy grid obtained from SIFT-descriptor model is integrated by using Recursive Bays rule with occupancy grid of the prior information about the static environment obtained by using sonar sensor. This reduces the computational burden of the system for mapping the indoor environment by providing static information about the environment that cannot be changed. This system can be used for the multi sensor data fusion tasks.

Chapter 5

Conclusion and Future Scope of the Work

5.1 Conclusions

In this work, we have studied different sensors and selected vision sensors for mapping the environment because of its high accuracy for distant objects, sonar sensors for acquisition of prior information about the environment. Analysed the key algorithm, brought out the image match method based on SIFT matching and introduced an empirical technique to optimize the system performance. The number of tentative matches given by SIFT, number of inliner matches given by RANSAC and percentage of inliner matches are given in Table 4.2. By using this experiment we have selected 0.85 as a threshold ratio where the percentage of inliner matches is highest. It is found that percentage of accurate matches in inliner matches given by RANSAC is 89% which much better than other type of sensor like Sonar Sensor.

We improved the result for distance measurement by using a new method [65]. Using this method an object in any location of overlapping area of stereo cameras can be measured after implementing the discussed SIFT algorithm. The improved results are given in table 4.3.

The occupancy grid obtained from SIFT-descriptor model is shown in Fig. 4.20 and integrated by using Recursive Bays rule with occupancy grid of the prior information about the static environment obtained by using sonar sensor. The modelling came up with satisfactory results for occupancy grid, almost similar to the original environment as shown in Fig. 4.22.

5.2 Future Scope of the Work

In this work, we have pre assumed while occupancy grid mapping according to prior information about the environment that the area lying at some depth from the level of the floor of mobile robot are occupied. Further research can be done for modelling the region lying at some depth from the level of the floor of mobile robot by using stereo vision camera or any other sensors.

From results of sensor fusion it is clear that only vision sensor cannot detect objects accurately so for increasing accuracy of the system, combination of same sensor as well as combination of different sensor may be used.

The dissertation presented the Bayesian theorems for occupancy grid mapping. Experiments with fuzzy logic algorithms, Dempster-Shafer grid and Neural Network algorithms are a possible and interesting research area to work further and may be get even better estimations.

References

[1] Hans P. Moravec, Alberto Elfes, "High Resolution Maps from Wide Angle Sonar,"Proc. IEEE Int'l Conf. on Robotics and Automation, CS Press, Los Alamitos, Calif, March 1985, pp. 116-121.

[2] H. P. Moravec, "Sensor Fusion in Certainty Grid for Mobile Robots," AI magazine, vol. 9, 1988, pp. 61-74.

[3] A.Elfes. Using occupancy grids for mobile robot perception and navigation. IEEE Computer, 22 (6), 1989, pp. 46-57.

[4] Sebastian Thrun, "Learning Occupancy Grid Maps with Forward Sensor Models," Autonomous Robots, Vol.15, 2003, pp. 111–127.

[5] Robin R. Murphy, "Dempster–Shafer Theory for Sensor Fusion in Autonomous Mobile Robots", IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, April 1998, pp 197-206.

[6] Robin R. Murphy, "Introduction to AI Robotics", Prentice-Hall of India Private Ltd, 2004,pp. 1-466.

[7] Sv. Noykov, Ch. Roumenin, "Occupancy Grids Building by Sonar for Mobile Robot", Robotics and Autonomous System, Vol. 55, 2007, pp 162-175.

[8] Md. Abdel Kareem Jaradat, Reza Langari, "Line Map Construction using a Mobile Robot with a Sonar Sensor", Int'l Conf. on Advanced Intelligent Mechatronics, Monterey, California, USA, July, 2005, pp. 24-28.

[9] Alfredo C. Plascencia, "Sensor Fusion for Autonomous Mobile Robot Navigation" A Ph.D Thesis, Automation and Control, Aalborg University, Denmark, August 2007, pp 1-114.

[10] Kurt Konolige, "Improved Occupancy Grids for Map Building", Autonomous Robots, vol.4, 1997, pp 351–367.

[11] R. C. Luo and M. G. Kay, "A Tutorial on Multisensor Integration and Fusion," in Proc. 16th Annu. Conf. IEEE Ind. Electron., vol. 1, 1990, pp. 707–722.

[12] R. C. Luo, M.-H. Lin, and R. S. Scherp, "Dynamic Multi-Sensor Data Fusion System for Intelligent Robots," IEEE J. Robot. Automat. vol. 4, Aug. 1988, pp. 386-396.

[13] B. Siciliano, "Control in Robotics: Open Problems and Future Directions," in Proc. 1998 IEEE Int. Conf. Control Applicat., vol. 1, 1998, pp. 81–85.

[14] B. S. Y. Rao, H. H. F. Durrant-Whyte, and J. A. Sheen, "A Fully Decentralized Multi-Sensor System for Tracking and Surveillance," Int. J. Robot. Res., vol. 12, Feb. 1993, pp. 22-44.

[15] M. Kam, X. Zhu, and P. Kalata, “Sensor Fusion for Mobile Robot Navigation,” Proc. IEEE, vol. 85, no. 1, Jan. 1997, pp. 108-119.

[16] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The Development of Honda Humanoid Robot,” in Proc. IEEE Int. Conf. Robot. Automat. vol. 2, 1998, pp. 1321–1326.

[17] D. Hall, Mathematical Techniques in Multisensor Data Fusion. Boston, MA: Artech House, 1992, pp. 1-430.

[18] A. Filippidis, L. C. Jain, and N. Martin, “Multisensor Data Fusion for Surface Land-Mine Detection,” IEEE Trans. Syst., Man, Cybern. C: Applications and Reviews, vol. 30, Feb. 2000, pp. 145–150.

[19] R. R. Carson, Jr., M. P. Meyer, and D. J. Peters, “Fusion of IFF and Radar Data,” in Proc. 16th DASC AIAA/IEEE Digital Avionics Syst. Conf., vol. 1, 1997, pp.9–15.

[20] S. Singh and D. Smith. (2006, December). “Approaches to Multisensor Data Fusion in Target Tracking: A Survey.” IEEE Transactions on Knowledge and Data Engineering. Vol. 18, No. 12. pp. 1696-171

[21] T. E. Bell, “Remote Sensing,” IEEE Spectrum, vol. 32, Mar. 1995, pp. 24–31.

[22] B. Solaiman, L. E. Pierce, and F. T. Ulaby, “Multisensor Data Fusion using Fuzzy Concepts: Application to land-cover classification using ERS-1/JERS-1 SAR composites,” IEEE Trans. Geosci. RemoteSensing, vol. 37, May 1999, pp. 1316–1326.

[23] A.Chiuderi, F. Stafano, andV. Cappellini, “An Application of Data Fusion to Landcover Classification of Remote Sensed Imagery: A Neural Network Approach,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst., 1994, pp. 756–762.

[24] D. G. Goodenough, P. Bhogal, D. Charlebois, S. Matwin, and O. Niemann, “Intelligent Data Fusion for Environmental Monitoring,” in Proc. Geosci. Remote Sensing Symp., vol. 3, 1995, pp. 2157–2160.

[25] Robert C. Levy, Gregory G. Leptoukh, Ralph Kahn, Victor Zubko, AruGopalan, , and Lorraine A. Remer. A critical look at deriving monthlaerosol optical depth from satellite data. IEEE Transactions on Geosciencand Remote Sensing, 47(8), 2009

[26] O. Rud and M. Gade, “Monitoring algae blooms in the Baltic Sea: A multi sensor approach,” in Proc. OCEANS ’99 MTS/IEEE, vol. 3, 1999, pp. 1234–1238.

[27] T. I. Liu and K. S. Anantharaman, “On-line sensing of drill wear using neural network approach,” in IEEE Int. Conf. Neural Network, vol. 2, 1993, pp. 690–694.

[28] K.-N. Lou and C.J. Lin, “An intelligent sensor fusion system for tool monitoring on a machining center,” in Proc. IEEE/SICE/RSJ Int. Conf. Multisensor Fusion Integration Intell. Syst., 1996, pp. 208–214.

[29] G. Littlefair, M. A. Javed, and G. T. Smith, “Fusion of integrated multisensor data for tool wear monitoring,” in Proc. IEEE Int. Conf. Neural Networks, vol. 2, 1995, pp. 734–737.

[30] Woo, Alec, David Culler. A Transmission Control Scheme for Media Access inSensor Networks. Mobicom 2001

[31] N. Levy, M. C. Zhou, and Y. Quan, “Real-time tool wear identification using sensor integration with neural network,” in Proc. 33rd IEEE Conf. Decision Control, vol. 2, 1994, pp. 1050–1051.

[32] A. N. Steinberg, C. L. Bowman, and F. E. White, “Rethinking the JDL
Data Fusion Model”, to appear in NSSDF Conference Proceedings, JHAPL, June, 2004.

[33] A.I. Hernandez, G. Carrault, F. Mora, L. Thoraval, G. Passariello, and J. M. Schleich, “Multisensor fusion for atrial and ventricular activity detection in coronary care monitoring,” IEEE Trans. Biomed. Eng., vol. 46, Oct. 1999, pp. 1186–1190.

[34] F. Azuaje,W. Dubitzky, N. Black, and K. Adamson, “Improving clinical decision support through case-based data fusion,” IEEE Trans. Biomed. Eng., vol. 46, Oct. 1999, pp. 1181–1185.

[35] B. Solaiman, R. Debon, F. Pipelier, J.-M. Cauvin, and C. Roux, “Information fusion, application to data and model fusion for ultrasound image segmentation,” IEEE Trans. Biomed. Eng., vol. 46, Oct. 1999, pp. 1171–1175.

[36] A. Mirabadi, N. Mort, and F. Schmid, “Application of sensor fusion to railway systems,” in Proc. IEEE/SICE/RSJ Int. Conf. Multisensor Fusion Integration Intell. Syst., 1996, pp. 185–192.

[37] A. Mirabadi, N. Mort, F. Schmid, “Fault detection and isolation in multisensor train navigation systems,” in Proc. UKACC Int. Conf. Control, vol. 2, 1998, pp. 969–974.

[38] K. Kobayashi, F. Munekata, and K.Watanabe, “Accurate navigation via differential GPS and vehicle local sensors,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst., 1994, pp. 9–16.

[39] C. E. Smith, S. A. Brandt, and N. P. Papanikolopoulos, “Vision sensing for intelligent vehicle and highway systems,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst., 1994, pp. 784–791.

[40] Z. Korona and M. M. Kokar, “Multisensor integration in the tracking of landing aircraft,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst., 1994, pp. 771–778.

[41] J. Sato, K. Shigeta, and Y. Nagasaka, “Automatic operation of a combined harvester in a rice field,” in Proc. of IEEE/SICE/RSJ Int. Conf. Multisensor Fusion Integration Intell. Syst., 1996, pp. 86–92.

[42] C.-Y. Chong and M. Liggins, “Fusion technologies for drug interdiction,” in Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. syst., 1994, pp. 435–441.

[43] LV-MaxSonar®-EZ1™, “High Performance Ultrasonic Range Finder”, MaxBotix®Inc, 26 April 2010.

[44] RudraPratap, “Getting Started with MATLAB”, Oxford University Press, New York, 2002, pp. 1-245.

[45] Alberto Elfes, “Sonar-Based Real-World Mapping and Navigation”, IEEE Journal of Robotics and Automation, Vol.3, No. 3, June 1987, pp 249-265.

[46] Ren C. Luo, Chih-Chen Yih, KuoLan Su, “Multisensor Fusion and Integration: Approaches, Application, and Future Research Directions”, IEEE Sensors Journal, Vol. 2, No. 2, April 2002, pp. 107-119.

[47] Ren C. Luo, Michael G. Kay, “Multisensor Integration and Fusion in Intelligent Systems”, IEEE Transactions on System, Man, and Cybernetics, Vol.19, No.5, September/ October, 1989, pp. 901-931.

[48] G.N.DeSouza and A.C.Kak. Vision for mobile robot navigation: A survey. IEEE Transactions on Pattern Analysis and Machine Inteligence, 24(2):237-267, November 2002.

[49] A.Elfes and L.Matthies. Integration of sonar and stereo range data using a grid-based representation. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 727-733, 1988.

[50] W.D.Rencken. Autonomous sonar navigation in indoor, unknown unstructured environments. In Proceedings of the IEEE International Gonference on Intelligent Robots and Systems, volume 1, pages 431-438, September 1994.

[51] D.J. Kriegman, E. Triendl, and T.O. Binford. Stereo vision and navigation in buildings for mobile robots. IEEE Transactions on Robotics and Automation, 5(6):792-803, December 1989.

[52] C. Jennings and Murray. Stereo vision based mapping and navigation for mobile robots. In Proceedings of the IEEE International Society for Optical Engineering, Bellingham, Washington USA, July 2004.

[53] H. Moravec. Rover visual obstacle avoidance. In Proceedings of the seventh International Joint Conference on Artificial Intelligence, pages 785-790, August 1981.

[54] S.J. Henkind and M.C. Harrison. An analysis of four uncertainty calculi. IEEE Transactions on Systems, Man and Cybernetics, 18(5):700-714, 1988.

[55] D.G. Lowe. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision ICCV, Corfu, pages 1150-1157, 1999.

[56] D.G. Lowe. Local feature view feature view clustering for 3D object recognition. Cvpr, 01:682-688, 2001.

[57] D.G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91-110, 2004.

[58] S.Se, D.G. Lowe, and J.J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 2051-2058, Seoul, Korea, May 2001.

[59] Jean-Yves Bouguet. Camera calibration toolbox for matlab. http://vision.caltech.edu/bouguetj/. Technical report, 2007.

[60] EmanueleTrucco and Alessandro Verri. Introductory Techniques for 3-D Computer Vision. Prentice Hall, 1998.

[61] A. Vedaldi and B. Fulkerson. VLFeat library. http://www.vlfeat.org/, 2008.

[62] An opean implementation of the SIFT detector and descriptor, A. Vedaldi, UCLA CSD Tech. Report 070012, 2006.

[63] M. Brown and D.G. Lowe. Invariant features from interest point groups. In Proceedings of BMVC, Caerdiff, Wales, pages 656-665, 2002.

[64] M.A. Fischler and R.C. Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM, 24(6):381-395, 1981.

[65] Hai-Sung Baek and Jung-Min Choi. Improvement of didtance measurement algorithm on stereo vision system (SVS).

[66] A. Elfes and L. Matthies. Sensor integration for robot navigation: Combining sensor and stereo range data in a grid-based representation: Combining sonar and stereo range

data in a grid-based representation. In Proceedings of the 26th IEEE Conference on Decision and Control, Los Alamitos, Calif., 1987.

[67] A.Elfes. Sonar-based real-world mapping and navigation. IEEE Journal on Robotics and Automatation, 3:249-265, June 1987.

[68] A.Elfes. Atesselated probabilistic representation for spatial perception and navigation. In Proceedings of the NASA Conference on Space Telerobotics, volume N90, pages 341-350, 1989.

[69] A. Elfes. Occupancy grids: a stochastic spatial representation for active robot perception. In Proceedings of the sixth conference of Uncertainty in AI, pages 60-70, July 1990.

[70] A. Dankers, N. Barnes, and A. Zelinsky. Active vision-rectification and depth mapping. In Proceedings of the 2004 Australasian Conference on Robotics and Automation, 2004.