University of Alberta

Vision-based World Modeling Using a Piecewise Linear Representation of The Occupancy Function

by

Dmitry Olegovich Gorodnichy

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta
Fall 2000

0-612-59593-5

University of Alberta

Library Release Form

Name of Author: Dmitry Olegovich Gorodnichy

Title of Thesis: Vision-based World Modeling Using A Piecewise Linear Representation of The Occupancy Function

Degree: Doctor of Philosophy

Year this Degree Granted: 2000

Dmitry Olegovich Gorodnichy
Volgogradskaya str., 14, Apt. 6,
Kiev, Ukraine, 252 110

Date: May 8, 2000

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Vision-based World Modeling Using A Piecewise Linear Representation of The Occupancy Function** submitted by Dmitry Olegovich Gorodnichy in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

. . . Wm. W. Armstrong . .
William W. Armstrong

. . . .
Walter F. Bischof

. . . Xiaobo Li . . . .
Xiaobo Li

. . . David Lowe . . . .
David Lowe

. . . .
Max Meng

. . . .
Hong Zhang

Date: May 8, 2000

Abstract

This thesis considers the task of building world models from uncertain range data. We study the occupancy approach, which is one of the most popular approaches used for this task. We identify three problems of this approach which prevent it from being used for building 3D world models. The thesis aims to resolve these problems.

The first problem concerns the design of sensor models which assign the values of uncertainty to registered range data. Vision-based sensors are the most affordable sensors capable of registering 3D range data. However, their sensor models are not known or are very difficult to calculate using probability theory. In the thesis we propose a new approach for building visual sensor models which uses evidence theory. This approach allows one to efficiently build sensor models of unreliable, inexpensive video systems by employing stereo error analysis. We present the design of an inexpensive visual range sensor which consists of a single off-the-shelf video camera. This visual sensor is shown to be very suitable for world exploration problems.

The second problem deals with the combination rule, which combines uncertainty values obtained from different range data. Approximations of the Bayesian and Dempster-Shafer rules, which are the common rules used in the occupancy approach, in many cases assume the independence of range data, contrary to the usual situation. In the thesis, we develop a new technique for combining range data which is based on regression. This technique does not make independence assumptions about the data and can therefore be applied to combining such dependent range data as those obtained by a single-camera range sensor.

Finally, the third problem concerns the redundancy of stored and processed data, which results from using the grid representation of the occupancy function. In the thesis we establish a new framework for representing the occupancy function in a parametric way using piecewise linear surfaces. This framework, which is the major

thrust of the thesis, uses the techniques we have developed for registering and combining visual range data, and is tested on both simulated and real range data. The advantages and the limitations of the proposed framework are studied. Besides being closer to optimal space-wise, this framework is also shown to be more efficient for map extraction and world exploration.

While much remains to be done in the area we believe that the proposed strategies for building sensor models, combining uncertain range data, and using parametrically represented occupancy functions provide the basis for new applications of the occupancy approach and will promote the development of this approach in both world modeling and robot navigation.

To this unique source of harmony
called the Gorodnichy family

Résumé

Dans la présente thèse, nous considerons la tâche de construire des modèles de l'espace à partir de données télémetriques incertaines. Nous étudions l'approche d'occupation, qui est l'approche la plus populaire pour cette tâche. Nous identifions trois problèmes propres à cette approche qui l'empêchent de pouvoir être utilisée pour construire des modéles de l'espace tridimensionels. Le but de cette thèse est de résoudre ces problèmes.

Le premier problème est relié à la conception de modèles de capteur qui determinent les valeurs d'incertitude au données télémetriques registrées. Les capteurs qui sont basés sur les signaux optiques sont les capteurs les plus accessibles capables du registre des données télémetriques tridimensionelles. Cependant, leurs modèles de capteur ne sont pas connus ou sont très difficiles à calculer en utilisant la théorie des probabilités. Dans cette thèse, nous proposons une nouvelle approche pour construire des modèles de capteur optique basée sur la théorie de l'évidence. Cette approche permet de construire efficacement des modèles de capteur de systèmes optiques peux coûteux et peux fidèles en utilisant l'analyse de l'erreur du système stéréo. Nous présentons la conception d'un capteur télémetrique fait á partir d'une videocamera générique. On fait la démonstration de l'adéquation de ce capteur optique pour des problèmes d'exploration de l'espace.

Le deuxième problème a rapport avec la règle qui combine des incertitudes obtenues á partir de différentes données télémetriques. Des approximations aux règles bayésienne et de Dempster-Shafer, qui sont les règles générales utilisées dans l'approche d'occupation, assument souvent l'indépendence des données télémetriques, contrairement á la situation habituelle. Dans cette thèse, nous développons une nouvelle technique basée sur la règression pour combiner des données télémetriques. Cette technique n'assume pas l'indépendance des données et peut donc être appliquée pour

combiner des données télémetriques telles que celles obtenues par le capteur fait á partir d'une vidéocamera.

Finalement, le troisième problème est relié à la redondance des données enregistrées et traitées, qui résulte de l'utilisation de la représentation de grille de la fonction d'occu- pation. Dans cette thèse nous établissons un nouveau cadre pour représenter la fonction d'occupation d'une voie paramètrique en utilisant les surfaces linéaires. Ce cadre, qui est la principale contribution du présent travail, utilise les techniques développées pour régistrer et combiner des données télémetriques optiques, et est testé sur des données télémetriques simulées et réelles. En plus d'utiliser beaucoup moins de mémoire, on démontre aussi que ce cadre est plus efficace pour l'extraction de la carte de navigation.

Si bien il reste beaucoup á faire dans ce domain, nous pensons que les stratégies présentées pour construire des modéles de capteur, tout en combinant des donnes télémetriques incertaines, et en utilisant des fonctions d'occupation á représentation paramétrique, fournissent la base pour de nouvelles applications de l'approche d'occupation et stimuleront le développement de cette approche dans la modelisation de l'espace et la navigation robotique.

Аннотация

В диссертации рассматривается задача построения пространственных моделей на основе неточных телеметрических данных. Изучается один из наиболее употребляемых подходов для данной задачи – оккупационный подход. Мы показываем, что данный подход имеет три проблемы, препятствующие его применению для построения трехмерных моделей мира. Диссертация посвящена разрешению этих проблем.

Первая проблема касается построения сенсорных моделей, задающих уровни неточности регистрируемым телеметрическим сенсорным данным. Оптические сенсоры являются наиболее доступными телеметрическими сенсорами, способными регистрировать трехмерные данные. Однако модели этих сенсоров либо неизвестны, либо очень трудны для вычисления, если для их вычисления используется теория вероятности. В диссертации предлагается новый подход для построения моделей оптических сенсоров. Этот подход позволяет эффективно рассчитывать модели ненадежных недорогих видеосистем, используя анализ стерео погрешностей. Мы предлагаем дизайн телеметрического сенсора, состоящего из одной недорогой видеокамеры, и показываем, что данный видеосенсор очень подходит для задач исследования окружающего пространства.

Вторая проблема касается формул, использующихся для объединения значений неточности телеметрических данных. Приближения наиболее используемых в оккупационном подходе формул Баеса и Демпстера-Шефера предполагают, что сенсорные данные являются независимыми друг от друга, вопреки тому, что на самом деле эти данные чаще всего являются зависимыми. В диссертации предлагается новый подход для объединения значений неточности, базирующийся на линейной регрессии. Этот подход не делает предположений относительно независимости данных и потому может употребляться для объединения таких зави-

симых друга от друга данных, как те, что регистрируются сенсором, состоящим из одной видеокамеры

Наконец, третья проблема касается избыточности запоминаемых и обрабатываемых данных. Эта проблема является следствием использования матричного представления оккупационной функции. В диссертации предлагается новая стратегия для представления оккупационной функции с помощью кусочно-линейных поверхностей. Данная стратегия является основным вкладом данной диссертации. Она базируется на предлагаемых в диссертации подходах по регистрации и объединению телеметрических видеоданных. Тестирование данной стратегии проделано как на симулированных, так и на настоящих сенсорных данных. Преимущества и недостатки предлагаемой стратегии изучаются. Помимо того, что данная стратегия потребляет значительно меньше памяти, она также является более удобной для вычисления навигационных карт.

You ask me, how old am I?
I tell you, it depends how do you count —
If age is defined
By the years from birth, I'm younger than you no doubt.
But if by the years, which left to death ...
I am much older than you, I guess.

D.O. Gorodnichy (from "*He and She*" album)

Acknowledgement

I express my deepest gratitude to my supervisors Dr. William W. Armstrong and Dr. Xiaobo Li for their help and guidance throughout my research.

I thank Dr. Russ Greiner for numerous stimulating discussions concerning the uncertainty combination part of the thesis, Dr. Hong Zhang for his in-depth and insightful comments on the computer vision and robotics parts and Dr. Walter F. Bischof for his valuable feedback on the computer vision and machine learning parts of the research as well as on the dissertation in general.

I would like to acknowledge that it was Dr. W.W. Armstrong who first suggested to use one camera for building occupancy world models. The idea of using piecewise linear surfaces for the occupancy function approximation was also suggested by Dr. W.W. Armstrong.

I also gratefully acknowledge help of Kyle Palmer in designing the multithreaded software architecture of Boticelli and Monroe Thomas in designing the PTU software.

This research was partially supported by the Defense Research Establishment Suffield, which is acknowledged as well.

I want to thank the Department of Computing Science for that friendly environment we have in the Department and also the International Center at the University of Alberta for that great and very important job they are doing.

I would also like to acknowledge the influence of Sandro Botticelli, one of the leaders of the Renaissance, whose name we used for our robot, and who inspired me to treat such a multi-faceted problem as Vision-based World Modeling with the attitude of a Renaissance man: trying to open the world to new ideas.

Finally, I will always be immensely grateful to and proud of my parents for being who they are.

Contents

List of Figures

List of Tables

Notations

Chapter 3:	
(i, j)	the coordinates of a pixel in the image plane
F	focal length of the camera measured in rasters
$\vec{m} = [i, j, F]_{unit}$	unit vector of a feature
$\vec{r} = \vec{m}r = (X, Y, Z)$	3D vector of a point in space
r	the distance to a point $\vec{r}$
δr	the range error
h	the baseline
$\vec{h}$	the baseline vector, which is the translation vector of the camera
Φ	the angle of the camera tilt rotation
Ω	the angle of the camera support
R	the rotation matrix
N	the dimension of the vector obtained by scanning an image with a 5 by 5 window
$\vec{V}$	N-dimensional vector, the elements of which are the intensities of the pixels in the scanning window
E	match error used in tracking
$\vec{r}_s$	a 3D point registered by a sensor
m	the occupancy value of a point
$m_{occ}(\vec{r})$	the evidence that a point is occupied
$m_{emp}(\vec{r})$	the evidence that a point is empty

Chapter 4:	
m	the occupancy value of a point
r	a range measurement; also the distance to a point
o	event that the point in question is occupied
e	event that the point in question is empty
$m(o)$	the evidence that a point is occupied
$m(e)$	the evidence that a point is empty
$Bel(o) \equiv m(o)$	the amount of belief that a point is occupied
$Pl(o) \equiv 1 - m(e)$	the amount of plausibility that a point is occupied
E	error of approximation of the function
α	the pan angle of a point
h	the height of a point
ϵ	acceptable error level
L	the number of hyperplanes used in regression
Chapter 5:	
s_t	the system of the robot at time t
r_t	reinforcement obtained by the robot at time t
π	navigation policy of the robot
$V(s)$	total discounted reinforcement, aka the value function

Chapter 1

Introduction

1.1 Presentation of The Problem

1.1.1 The Challenge of World Modeling in Robotics

The era of computers and robots has come. Robots are replacing humans in many aspects of life, and first in environments which are hazardous for a human or where a human cannot be physically present. Such environments unclude mine fields, high-voltage workspaces, highly polluted and radioactive environments, sea bottoms, war battle frontiers, surfaces of other planets, and other environments in space missions.

In order to operate successfully in an unknown environment, a robot may need to learn a model of the environment. Robots however are not humans and they do not perceive the world as precisely as we humans do. Let us name the main practical limitations on a robot's ability to acquire accurate models of the world.

First, sensors are never perfect. They have limited resolution, they provide data corrupted by noise, and above all, they provide only a limited amount of data. More expensive sensors are able to register data more precisely than cheaper ones. Nevertheless, as we understand, there always will be an issue of time and costs involved in obtaining the data. There might be situations when we cannot afford expensive sensors, as, for example, in the case of using robots in hazardous places where the chance of breakage is high.

Second, in most applications a robot should operate in real time. That is, in addition to hardware limitations imposed by the cost of the sensor, there is also a time constraint which does not allow a robot to obtain enough or more precise data.

Third, the environment around a robot is complex and dynamic. This can result in contradictory sensor readings.

Finally, robot's motion is inaccurate due to the drift and slippage, which results in incorrectness in object location estimation. These odometric errors however can be reduced by limiting the mobility of the robot and in this work we do not consider them, focusing our main attention on treating the other limitations mentioned.

The major challenge of the research comes from the desire *to build adequate world models from inaccurate and incomplete range data*, where the adequacy of the model is judged from its suitability to a given task.

1.1.2 Occupancy Approach and Grids

The occupancy approach was formulated at the CMU Robotics institute in 1983 to turn wide angle range measurements from cheap sonar sensors into a 2D spatial map [MM96]. Since then, it has become one of the main approaches used in robotics for building world models from uncertain range data, especially in situations

a) when there is no *a-priori* knowledge of the geometry of the environment, and

b) when low-cost and not very reliable range sensors are used.

The founders of the occupancy approach believed that the approach had a great future, if applied to building 3D models from 3D range data [MM96]. The approach however has an intrinsic problem which prevents it from being used for constructing 3D and large scale models. This problem is the grid representation of the world used in the approach. Everything in the approach: sensor models, which assign the degree of uncertainty to registered data; combination rules, which are used to combine the data; map extraction methods, is based on this representation. This is even why the occupancy approach is mainly known under the name of the *occupancy grids* approach. The grid representation however may result in storing and processing hundreds of millions of voxels, which makes constructing and using models of 3D environments in real time practically impossible. And this is not the only problem incurred by using grids. Grid models are not suitable for radial range data, which is the most frequent case, and they are very inefficient for map extraction.

1.1.3 Vision-based Range Sensors

In addition to the representation problem mentioned above, there is also a problem of designing low-cost and fast 3D range sensors. Vision-based sensors appear to be the most suitable candidates for that, since sonar sensors do not provide accuracy sufficient for 3D modeling, while laser-based range finders are expensive and/or slow. Sensors that are not based on vision are also indescriptive of registered data, meaning that they do not have control over the strategy of selecting the features, whereas vision-based sensors allow one to choose features depending on the picture observed.

The development and application of vision-based range sensors however is greatly impeded by the unavailability of video camera sensor models which assign values of uncertainly to uncertain visual data. Using expensive highly calibrated or digital stereo system resolves this problem by making all observed visual data certain. But then these stereo systems are no longer low-cost and/or fast.

1.2 Statement of The Thesis

Objectives of the work

The goal of the dissertation is to provide solutions to the problems of vision-based occupancy world modeling described above. This includes two main objectives. The first objective is related to computer vision and is

> to design an affordable 3D range sensor based on an inexpensive off-the-shelf video camera, which will be capable of registering 3D range data in real time, and then to provide an approach for building sensor models of

unreliable visual sensors.

The second objective is related to world modeling and is

to develop a new framework for combining, representing and using occupancy data using a parametric piecewise linear representation of the occupancy function which represents the occupancy model of the world.

Importance of the work

The techniques proposed in the thesis for building sensor models, combining uncertain range data, and using parametrically represented occupancy functions provide solutions to the problems occurring in building 3D occupancy world models, thereby providing the basis for new applications of the occupancy approach and promoting the development of this approach in both world modeling and robot navigation.

Contents of the work

The dissertation contains the results of the author which appeared previously in six publications of the author: five appeared in proceedings of refereed conferences [Gor99, GA99b, GA99a, ACG99, GA00c] and one appeared in a journal [GA00a]. These results have been reported and discussed at various computer science forums, including International Symposium on Robotics (ISR/PRECARN'2000), Montreal, 14-17 May 2000 (planned); International Joint Conference on Neural Networks, Washington DC, July 21-23, 1999; Vision Interface conference, Trois-Riviéres, Canada, May 18-21, 1999; Quality Control by Artificial Vision Conference, Trois-Riviéres, Canada, May 18-21, 1999 Canadian Conference on Electrical and Computer Engineering, Edmonton, Canada, May 9-12, 1999.

1.3 Organization

The problems covered in the thesis belong to different areas of computing science, and in the literature, in many cases, they are studied by different research communities.

For example,

- The design of a stereo rig and stereo error analysis are studied by researchers in Image Processing and Computer Vision;

- Manipulations with evidence and probability values are of interest to researchers in Uncertainty in AI and Belief Networks. This includes assigning values of evidence to uncertain data, as well as rules used to combine these data.

- Learning from data is a forte of scientists from Machine Learning and Neural Network community as well as statisticians who deal with regression or other estimation and induction methods.

- A new approach for world modeling is of interest to World Modeling and Virtual Environment research community.

- Finally, most of the work presented in the thesis has applications in Robotics, where one finds the greatest number of publications dealing with occupancy grids, map extraction and world exploration.

The organization of the thesis in chapters is done according to the areas the problems in question belong to.

First, in Chapter 2 we describe the world exploration problem, which is the main problem the occupancy approach is applied to, and which is the problem we make use of throughout the thesis. In this chapter we describe the occupancy approach, and focus our attention on the problems of this approach. In particular, we identify the following problems. The first problem concerns the visual range sensor, the sensor model of which has to be developed in order to be used in fusing. The second problem deals with the combination rule which in many cases assumes the independence of range data, contrary to the usual situation. Finally, the third problem concerns the redundancy of stored and processed data, which results from using the grid representation of the occupancy function. The outline of the solutions proposed in the thesis to the above problems concludes the chapter.

Then, in Chapter 3 we concentrate on the vision part of the research. We describe the concept of a sensor model and explain why it is difficult to build a sensor model of a vision-based sensor. We introduce a single camera 3D range sensor and show the advantages of it, the major of which are the following: it is fast and inexpensive, and it is convenient for building a sensor model. We describe the design of the sensor, present the stereo error analysis, build the sensor model of it and show the application of it to mobile robot world exploration. It is also in this chapter when we first argue in favor of evidence theory over probability theory. We show how evidence theory allows one to efficiently build sensor models of vision-based sensors using the error analysis of the vision system.

Chapter 4 studies the issue of combining uncertain range data in the occupancy approach. We show the deficiencies of the existing combination rules. In particular, we show that Bayesian and Dempster-Shafer-based rules, which are the main rules used in the approach, may produce non-sensical results when applied to fusing data obtained by a single-camera stereo. We then propose a new technique for combining uncertain range data which is based on regression. In order to apply regression to sample points, which are range measurements along with their evidence values, we draw upon the evidence approach, thereby showing one more advantage of this approach over the probabilistic approach. We describe the proposed technique for a general case of using an arbitrary regression technique, and then we show how it can be efficiently implemented using Adaptive Logic Networks as a regression tool.

In Chapter 5, we demonstrate that occupancy models can be represented parametrically and show the advantages of representing the occupancy function using piecewise linear surfaces. These advantages are 1) efficiency in representing the model; 2) high speed of constructing the occupancy function; 3) efficiency for map extraction. This chapter is dedicated to showing how to use parametrically represented occupancy models for the world exploration problem, which is the problem the occupancy approach is commonly applied to. In particular, we describe an approach which uses the information extracted from the constructed occupancy models in order to make navigation decisions. This approach uses a reinforcement learning technique where the reinforcements are obtained from the knowledge of the goal location as well as from the knowledge of unexplored area and the likelihood of obstacles in the exploration area, which is extracted from the occupancy models.

A framework for building occupancy models using visual range data and parametrically represented occupancy functions is presented in Chapter 6. This framework makes use of the techniques described in the preceding chapters and consists of a set of stages which lead from the first step of grabbing a video-frame to the final step of making a navigation decision based on the constructed models. Special attention is given to the implementation of the proposed framework. We describe the software and hardware architectures of robot Boticelli designed for purpose of demonstrating the approaches proposed in the thesis. The data obtained from running the robot are presented.

The last chapter summarizes the contributions of the dissertation, lists directions for further improvement of the techniques proposed and presents our vision for future work in the area of vision-based world modeling using occupancy functions.

Chapter 2

Occupancy Approach

2.1 Introduction

The occupancy approach was proposed as a solution to the world exploration problem. This problem, which deals with the making navigation decisions from small, noise-prone quantities of sensor data, has been of interest to the researchers in mobile robotics for almost two decades. As early as 1980, it was realized that direct sensor measurements should not be used in planning navigation [Mor81]. Instead, multiple sensor measurements should be taken into account and world models should be built in order to be used in making intelligent navigation decisions.

In this chapter, we describe the world exploration problem and overview the work done by various robotics labs on this problem. The focus of our attention is vision-based mobile robotics and applications of the occupancy approach, which are the areas of our contribution. The occupancy approach is introduced and the properties of this approach which make it indispensable for mobile robotics are described. The applications of the occupancy approach for world modeling problems are presented.

The emphasis of the chapter is on the problems of the occupancy approach. In order to resolve these problems, a new paradigm has to be proposed. The outline of this paradigm concludes the chapter.

2.2 World Exploration Task

The world exploration task, in a general form, is described as follows.

> An agent equipped with range sensors is put in an absolutely unknown environment. The agent tries to understand what is around it, by building a model of the environment, in order to accomplish a given task.

As understood, the quality of the world models that an agent wants to build is governed by two factors:

1. the tasks the agent has to fulfill, and
2. the quality of the available range sensors.

For example, precise 1 mm resolution models, such as the ones used for building virtual environments [EHBR98, GMB98] or surgical applications [Gri99], cannot be built from unreliable data registered by inexpensive range sensors. On the other hand, there might be no need of very precise models, as, for example, in the case when the models are used for guiding a robot to a goal or in exploring large-scale environments. Thus, when designing a world modeling technique, it is important to know how the models will be used and what the properties of the registered range data are.

2.2.1 Two Types of World Modeling

Along the lines written above, we can distinguish two types of modeling. The first type is concerned with building as precise a model of the world as possible. The issue of time is not a concern for this type of modeling. The range data are obtained by expensive and reliable range sensors, which are well calibrated and which operate in a calibrated environments using turn-tables [RGEZ99] or predefined positions of camera [GMB98]. This type of modeling is used in building virtual representations of real environments.

The object of investigation of this dissertation however is another type of modeling, which we may refer as fast-but-crude modeling as opposed to the precise-but-slow modeling described above. This type of modeling is the one used in mobile robotics, where the issue of time efficiency is of prime concern and where it is common to use inexpensive range sensors, which in many cases are not very reliable. Table 2.1 summarizes the features of two types of modeling.

In the next section, we present an overview of the reliability characteristics of different range sensors used in robotics and next we present the testbed problem for fast-yet-crude modeling.

Table 2.1: *Two types of world modeling.*

	Type 1: precise-but-slow	Type 2: fast-but-crude
Nature of range data	localized in one area, dense	distributed over large range, collected all around
Time requirement	may take days to calculate	real-time
Required precision	as precise as possible, 1mm resolution	precise enough to navigate, 1dm–1m resolution
Applications	virtual environments, cyber-cities, surgeries	exploring unknown environments, finding objects, collecting new data
Sensors used	laser scanners, multiple-cameras calibrated video system	sonars, cameras, laser range finders, infrared sensors

2.2.2 "Find an Object" Problem

Consider the following world exploration problem. A robot is put in an absolutely unknown environment. The objective of the robot is to find a predefined target object as fast as possible, using data registered by its sensors. We will refer to this problem throughout the thesis and make of use it to test the approaches proposed in the thesis. Let us show what are the main challenges of this problem.

First, the fact that the environment is absolutely unknown means that there is no *a-priori* information about the environment, such as maps of the environment or any assumptions made about the environment concerning its scale, its shape, or its dynamics. Thus, the model of the world is built from scratch and without resorting to geometry-based techniques.

Second, the objective of finding an object fast implies two things. On one hand, the robot has to make intelligent navigation decisions based on the world models built from the acquired range data, where range data can be partial or unreliable. On the other hand, the robot has to build these models fast, preferably in real time, and it has to extract the information needed for navigation from these models also fast, where this information includes the knowledge of already explored areas and the areas where exploration is still required.

The time constraint of this exploration problem requires fast-but-crude modeling of the environment Another reason for fast-but-crude modeling comes from using unreliable range sensors and the desire to make robots affordable.

The efficiency and the quality of modeling in this problem is defined by the time needed for the robot to find the target and also by the costs incurred in doing the job.

2.3 Range Sensors

Range sensors are non-contact sensors that measure the distance to a point in space. Depending on the technique used to calculate the distance to a point, they have different ranges, depth and angular resolution, measuring rates and prices.

It should be mentioned that all range sensors have limited accuracy and rate of acquisition, though some are more accurate and faster than others; and as a rule, the more reliable a sensor is, the more expensive it is.

In the current dissertation we use an inexpensive single videocamera range sensor. We develop a sensor model of it and we use it in our experiments. However, the techniques proposed in the dissertation can as well be applied to other unreliable range sensors. So, let us review those sensors.

The most popular in robotics sensors are ultrasonic sensors [TFB98, BEFW97], time-of-flight lasers [For98], range cameras [RGEZ99, PHLG97] and videocamera-based sensors [JD97, YSA98, Mor96]. Other types of sensors include radar sensors [Mul98] and infrared cameras [Pol99, Mal93]. They however are not commonly used in robotics at the present moment.

Ultrasonic sensors are the most affordable range sensors. However, they have low measuring rates due to the low speed of sound and short measuring range (5–10 meters). They are also very susceptible to the angle of incidence and the reflectance properties of the objects. Range resolution of ultrasonic sensors is about 1 centimeter. Their angular resolution however is very low: 50–200 mrad, which makes them unsuitable for high-precision modeling.

Time-of-flight lasers have maximum range of 10–30 meters and high range resolution (1–5 centimeters) and angular resolution (1-20 mrad). Their acquisition rate is about 1 frame per second. They have very few spurious measurements. They have however many drop-outs. They are usually expensive. They also register depth data in one plane only or according to a predefined pattern [Lau99].

Range cameras are faster and cheaper than scanning time-of-flight lasers. Their range resolution is 0.01–10 millimeters and angular resolution is 0.2-02 mrad. They have limited useful range however (less than 5 meters) and are expensive.

Finally, vision-based sensors can register 3D range data with a wide range. They are generally fast and have high angular resolution. Rate of acquisition and range resolution of vision-based sensors depend upon the quality of the videocamera, stereo setup and depth acquisition algorithms. Tradeoffs between the quality of range measurements, the cost of the stereo setup and the complexity of depth acquisition algorithms are very apparent. The main source of unreliability of video-data comes from the limited resolution of the camera and the correspondence error.

2.4 Occupancy Approach

2.4.1 Definition

The occupancy approach was proposed in order to facilitate fusing of uncertain range measurements. According to this approach, each point in space is associated with what is called an *occupancy value of the point.* Mathematically, occupancy modeling can be written as mapping from 3D or 2D space into real values:

$$m = m(r), \quad r \in \Re^3 \tag{2.1}$$

Usually, *occupancy values* designate the likelihood that a point in space is occupied: the higher the value is, the more likely it is that the point is occupied. However, they can also designate the likelihood that a point in space is empty, in which case they are sometimes called *emptiness values* and the models are called *emptiness models.* Figure 2.1 shows an example of an occupancy model of a simple 2D environment consisting of two perpendicular walls.

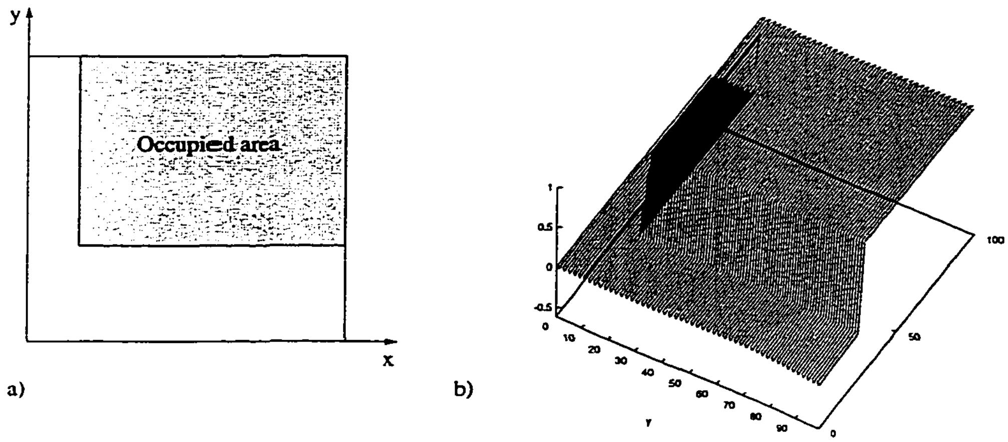


Figure 2.1: *Two dimensional occupancy model of the world. The higher the value of the point is, the more likely it is that the point is occupied.*

Though simply defined, the occupancy approach poses a lot of questions and problems.

2.4.2 Difficult Questions

The occupancy approach consists in calculating the occupancy values of all points in the explored space, according to the range data measured by a range sensor. In the original work of Elfes and Moravec [ME85], who are considered to be the founders of the occupancy approach, as well as in most follow-on works [Elf86, ME88, Elf89, MM96, Mor96, JD97, TFB98], the occupancy values are calculated as probabilities that a point in space is occupied, conditioned on all available range measurements. Designating the event that a point A is occupied as O_A and range measurements as $r_1, r_2, ..., r_M$, this can be written as

$$m_A \equiv P_A^{r_1 \ldots r_M} = P(O_A | r_1, r_2, ..., r_M) \tag{2.2}$$

This formulation allows one to use the probabilistic framework for fusing uncertain range data, and in particular, to use conditional probability tables to represent range data. The following example illustrates the concept.

First question

Let's say, according to the r-th measurement of a sensor, point A is occupied with probability 0.7, i.e. $P_A^r \equiv P_r(O_A) = 0.7$. Then, for each point α different from point A, the probability that the point is occupied P_α^r is expressed as the conditional probability

$$m_\alpha \equiv P_\alpha^r = P_r(O_\alpha | P_A^r = 0.7) \tag{2.3}$$

A couple of obvious questions arise.

Q1a: How should we assign a probability of occupancy to a registered point (e.g. the value of 0.7 to a point A in the example above)?

Q1b: How does one calculate and store conditional probability tables needed to calculate the occupancy probabilities of all points in space after an individual measurement is taken (i.e. all points α around point A)?

As can be seen, these conditional probability tables can be quite big, if there are many points in space and there are many possible occupancy values.

These two questions describe what is called the *sensor model design problem.* This problem deals with the assignment of occupancy values to the points in space after each individual measurement. Occupancy values are sometimes also called confidence, evidence, or uncertainty values and in this dissertation we use these notions interchangeably.

Assuming that this problem is resolved, there is yet another problem.

Second question

Let's say, there are M range measurements, according to which the probabilities that point A is occupied are $P_A^1, P_A^2, ..., P_A^M$, respectively.

Q2a: How should these M values of occupancy obtained from different measurements be combined to give the total occupancy value $P_A = f(P_A^1, P_A^2, ..., P_A^M)$ of a point?

In reality, this problem is aggravated by the fact that, in most cases, M *different* range measurements describe not the same point A, but M *different* points $A_1, A_2, ..., A_M$, which lie close to each other, yet are not the same. Then the question is

Q2b: How should one combine the occupancy values $p_{A_1}^1, p_{A_2}^2, ..., p_{A_M}^M$ of closely located points?

These questions refer to what is called as the *combination rule problem.*

Third question

Now, assuming that all points α in space have their occupancy values calculated, i.e. P_α have been calculated, the question is how to use them.

Q3a: How should occupancy models be used?

Q3b: Should all points with the probability higher than 0.5 be considered as occupied? What is the most appropriate way of using them? How should one extract maps needed for navigation, if occupancy models are used for world exploration?

These questions address the *application problem.*

Fourth question

When dealing with world modeling, one may wish to visualize the constructed world models either for the purpose of analyzing the quality of the models or for the purpose of computer-human interfacing. One major question is

Q4: How should occupancy models, which are four-dimensional (x, y, z, m) models, be visualized on a 2D screen?

This is the *visualization problem.* Finally, comes the main problem.

Fifth question

Q5a: How many points α in space should be used when modeling an environment? How one should store the occupancy model?

In the original work of Elfes [Elf86], as well as in all subsequent work, the environment is represented as a two- or three- dimensional grid. The resolution of the grid defines the number of points used to represent the environment. The occupancy values are stored as elements of a multidimensional array. A natural question one may ask is

Q5b: If one does not know anything about an environment, how can one tessellate it into a grid? How can one decide what the resolution of the grid should be? Should the resolution be the same on the whole environment?

It is understood, for example, that some parts of the environment, which contain elaborate details, may require a finer resolution, whereas others parts of the environ-

Table 2.2: *Advantages and deficiencies of the occupancy approach.*

Advantages	Deficiencies
Easy to understand	Requires a lot of memory and computation power
Doesn't require any extra knowledge about the environment	Inconvenient for map extraction
Easy to maintain	Requires simplification assumption for combination
Good for unreliable sensors	Difficult to visualize
Fast for small-scale models	Slow for large-scale models
Uses the knowledge of sensor models	Requires the knowledge of sensor models

ment, which contain nothing or big bulky objects, for example, can be modeled using large scale grids.

This is the nature of the problem which impedes significantly the development of the occupancy approach — the so called *representation problem.*

The last question

As can be seen, the occupancy approach has many problems. These problems will be reviewed once again later in the chapter when presenting the outline of the solutions to these problems. A question one may wish to ask now is

Q6: If there are so many problems with the occupancy approach, what makes it so valuable for world exploration?

Let us consider the advantages of the occupancy approach.

2.5 Advantages of Occupancy Modeling

The choice of the model to be used in the description of the world depends on three factors: 1) environment, 2) sensors and 3) task goals.

Recent research has produced three fundamental paradigms for world modeling for mobile robots: a geometrical (beacon recognition and tracking based) paradigm, a topological (graph-based) paradigm, and a metric (occupancy-based) paradigm [TB96].

The geometrical approach is based on extracting geometrical features of the environment. In most cases, the environment is assumed to consist of planar objects: planes and lines, which are then extracted from the range measurements. Such techniques as Hough transform, Gabor or Kalman filtering are the most common techniques used to extract lines and planes. The uncertainty of a range sensor is propagated to the model using the first two moments of the probability distribution of the spatial variables of the environment. The geometrical approach is well suited to modeling indoor and human made environments and it has found many applications in vision guided robots [UD95]. The geometrical nature of the approach makes the extraction of navigation maps easy. It can be seen however that, due to multiple matrix multiplications involved in updating covariance matrices, the geometrical approach is rather time-consuming and therefore cannot be used to model complex environments in real time. The assumption about the geometry of the explored environment required by the approach also makes it impossible to use the approach for arbitrary environments, especially for outdoor and dynamically changing environments.

The topological approach is often used for robot navigation planning. Its advantages are low space complexity, which is due to the fact that the resolution of the model in the topological approach depends on the complexity of the environment. It is convenient for problem solvers and natural language interfaces. Topological models however are difficult to construct and maintain, especially for large environments. They require recognition of places used as landmarks and are sensitive to the point of view and therefore cannot be used in arbitrary environments.

The occupancy approach, as can be now seen, has the major advantage over other approaches in that it can be used for modeling absolutely unknown environments. It does not require any assumptions about the shape of the environment. Another major advantage is its ability to deal with unreliable range data. It is also superior to

Table 2.3: *The conditions under which the occupancy approach is commonly used.*

1.	When no geometric constraints can be imposed on the environment
2.	When the environment is changing
3.	When range sensors are not reliable
4.	When computation time is critical (e.g. as in mobile robotics)

others for its simplicity in building, representing and maintaining the world model.

Table 2.2 summaries the advantages and deficiencies of the occupancy approach and Table 2.3 recapitulates the conditions under which the occupancy approach is commonly preferred to other approaches.

2.6 Problems and Solutions: Overview

The occupancy approach does not have as long a history as the geometrical approach, and many aspects of it still need to be developed and investigated.

In the current thesis we bring forward four main problems of the occupancy approach, which need more investigation. Some of these problems have already been identified and solved by other researchers, others have not. Let us review the problems and outline the solutions we propose in the thesis.

2.6.1 Sensor Model Design Problem

The sensor model design problem does not arise if range sensors are considered flawless. In this case, a registered point gets the probability of occupancy equal to one, and all points between the sensor and the point get their occupancy probabilities assigned to zero. This situation is idealistic since, as mentioned in Section 2.3, the more affordable the sensors, the less accurate are their measurements.

The sensor model design problem also does not arise if sensor models are given by a manufacturer of the sensor, as in the case of Polaroid sonars [PNDW95]. This explains why in occupancy approaches the most commonly used sensors are laser scanners, highly calibrated trinocular stereo systems and sonar sensors.

Video cameras provide an inexpensive and accurate source of 3D range data. However, because their sensor models are not known, it is difficult to use them for occupancy-based modeling.

In Chapter 3, which is based on our paper [GA99b], we present a single-camera-based 3D range sensor and show a way of calculating the model of this sensor. The design of this visual sensor is governed by the objective to provide in real time range data which will be sufficient for building 3D occupancy models of the required quality. The advantages of the single-camera sensor include a) acquisition of range data all around the sensor, b) high angular resolution (outliers lie on the ray of view only), c) possibility of using the camera for surveillance purposes concurrently with acquiring depth data, d) control over the number of selected features, e) high speed of data acquisition, and which is the most important h) its convenience for designing a sensor model (levels of uncertainty are associated with the registered data using stereo error analysis in accordance with evidence theory).

2.6.2 Combining Range Data

The rules used to combine uncertain data depend on the way the uncertainty is represented. There exist two paradigms for representing uncertainty. The first one is based on probability theory and the second one is based on evidence theory. Both paradigms are intensively used in robotics, and the question of which paradigm is better is widely discussed in the AI community.

In Chapter 4, which is based on papers [Gor99, GA00b, GA00a], we contribute to this discussion, arguing in the favour of evidence theory when applied to fusion of range data. We showed that in certain cases commonly used rules cannot be applied they way they usually are, because of the assumptions made. These include the cases when the registered range data are not independent. An example of this is when the data are registered by a single-camera stereo. Thus, we proceed to the proposal of a new range data fusion technique based on regression. This technique uses evidence theory in assigning occupancy values and builds the occupancy function by fitting the sample data provided by a sensor with a piecewise linear function.

2.6.3 Representation and Application Problems

The most difficult problem of the occupancy approach for world modeling deals with the grid representation of the world required by the approach. We consider this problem to be the most difficult, because grid representation is believed to be the basis of the occupancy approach and nobody ever seemed to challenge this representation because of that.

This representation results in storing and processing huge arrays of data. In robotics, this is the reason for building two dimensional models instead of three dimensional ones [MM96, BEFW97, JD97, YSA98, BBC+95, PNDW95]. In other areas of world modeling because of this problem, only models of small objects are constructed using the occupancy approach [EHBB+97, RGEZ99, PHLG97].

Octrees have been suggested to replace the uniform grid representation of space [PHLG97]. This however does not resolve the problem, since the construction of octrees is still based on grids, and, while final representation of the world takes less space, the amount of computations required only increases.

The solution to this problem is seen in replacing the grid representation of the world with another one. This however would require reconsideration of all stages of the occupancy approach, since everything in the approach: from assigning the values of uncertainty to combining and using these uncertainty values, is based on this representation. This is what we propose in Chapter 5, which is based on papers [GA99a, GA00a, ACG99], where we show how occupancy models can be represented using a min-max tree of combining linear functions. While being rather crude, parametrically represented occupancy models require little space, are fast to build and can easily be applied to navigation problems

2.6.4 Applying Occupancy Models

Grid-based occupancy models are very inconvenient for map extraction [TB96]. This is because in order to get the boundary of the area available for navigation, the models, which are 3D arrays of data, have to be ray-traced, then line extraction algorithms have to be applied to obtain a 2D boundary. This consumes a lot of computational power.

However, as we show in Chapter 5, parametrically represented occupancy models are efficient for map extraction and navigation planning. The validity and the promise of our techniques are demonstrated by implementing them on the mobile robot Boticelli, which searches for objects in an unknown environment using a single camera stereo range sensor. The implementation issues of building and using the occupancy models are covered in Chapter 6, which puts together the results obtained in the previous sections and which is based on papers [GA00c, GA00a].

2.6.5 A Historical Remark

The work under this dissertation started with only two objectives: 1) to design an affordable 3D range sensor, and 2) to attempt to introduce a new parametric representation for the occupancy models. It was then realized that in order to achieve this we had to reconsider the occupancy approach from its very origin. And this how we started exposing all problems of the occupancy approach and started asking difficult questions about the approach. We saw that there were no good answers to any of the described questions, "good answers" meaning answers that do not make fallacious assumptions and do not require unrealistic simplifications.

It became clear that the representation problem cannot be considered separately from other problems; that it cannot be resolved without other problems being resolved; that a completely new framework should be established for the occupancy approach — *a framework for 1) registering, 2) combining, 3) representing and 4) applying the occupancy values.*

Chapter 3

Visual Sensor for Mobile Robot World Exploration

3.1 Introduction

This chapter addresses two important issues related to the vision-based world modeling. The first issue belongs to the Image Processing and Computer Vision area of Computing science and deals with the hardware and software implementation of the visual range sensor. The second issue, on the other hand, is more of an Uncertainty in Artificial Intelligence nature and studies the problem of calculating the levels of uncertainty of the range data registered by unreliable sensors.

We start with the presentation of a new single-camera-based 3D range sensor. The design of the visual sensor is governed by the objective to provide in real time range data which will be sufficient for building 3D occupancy models of the required quality. The advantages of the sensor for mobile robot world exploration are shown.

We then proceed to the study of the reliability characteristics of the sensor and propose a new approach based on evidence theory which allows one to calculate the levels of uncertainty of registered data using the stereo error analysis of the sensor. The advantages of the proposed approach will be clearly seen in the next chapter where the uncertainty values of registered data are fused to build a plausibilistic occupancy model of an environment.

In this chapter we show the applications of the proposed techniques for mobile robot navigation. Data obtained by running a single camera mobile robot are presented. Directions for further improvements of the proposed visual sensor conclude the chapter.

3.2 Previous Work

3.2.1 Reasons for This Research

In world exploration, the occupancy model of the world is one of the most commonly used [FBE94, TFB98, JD97, YSA98]. Originally developed for building 2D maps [MM96], the occupancy approach has recently been extended to build 3D models of the world [Mor96, PHLG97], which provide much more information about the environment. However, as discussed in the previous chapter, there are serious problems with building 3D occupancy models, and one of them concerns range sensors.

Sonar sensors are not expensive and their models are known [FBE94, PNDW95]. That is, usually a manufacturer of the sonar sensor provides all information needed for calculation of the reliability of the registered data. However, because of their wide angular resolution, sonar sensors are not suited for 3D modeling [Mor96].

On the other hand, laser range sensors and highly calibrated stereo systems have resolution which makes them suitable for building 3D models. These sensors can calculate depth very precisely, which makes it easy to build their models. This is however also what makes them so expensive and why they cannot be used in many situations because of the costs involved. Another problem with laser and stereo-based range sensors is that they are too slow, either because of too many calculations being involved or because they provide too much data.

Thus, there is a need for an inexpensive 3D range sensor which would be capable of acquiring depth data around a robot in real time. An off-the-shelf video camera can be used for the design of such a sensor, as soon as the model of this sensor can be calculated.

3.2.2 Vision Based Mobile Robot Projects

Video cameras are used intensively in mobile robot navigation [JD97, BBS+97, MS97, UD94, WW94]. However, as applied to occupancy-based robot navigation, the best known visual sensors are the ones designed at UBC (Spinoza robot) [JD97], at the University of Bonn (Rhino robot) [BBC+95, TFB98], CMU CS (Xavier and Amelia robots) [Thr98], CMU RI by Moravec [MM96], and by Yamauchi (Ariel Project) [YSA98].

The sensors in the above mentioned robot projects are designed in such a way that the visual data they provide are considered to be precise and correct. In other words, there is no uncertainty associated with registered visual data. This precision of data is achieved either by using a trinocular stereo as in [JD97], which eliminates outliers or by rectifying each image according to a precalculated rectification matrix with 12 parameters as in [SRG99, Mor96], by using Hough or Gabor transforms [BBC+95]. These techniques result in extra cost of the stereo setup and an extra amount of computations needed in calculating the depth of a point.

It should be noted that in other world modeling approaches, like geometrically based ones, uncertainty of the video data is often used in computation, where it is gradually refined by using Kalman-filter-based approaches. For occupancy-based approaches however, uncertainty of visual range data is not calculated, nor it is used.

Moravec was the first who used the occupancy approach to realize that even in highly calibrated stereo systems there is a chance of registering spurious points. He suggested associating a level of confidence with the range date registered by a video camera. This idea however has not been developed, because of the problems involved in building a sensor model of video systems. This dissertation offers a way to resolve these problems.

3.3 Exploration Task and Goals of Sensor Design

Let us make it clear what we want to achieve. It is understood, the design of a visual sensor depends on tasks the sensor is used for. In this research that range sensors are considered for the purpose of building occupancy models of the environments surrounding a robot. This implies two things:

- first, that the sensor has to register range data all around the robot, and
- second, it should do it fast, so that a robot can operate in real time.

This guides the design of the visual range sensor. In order to decide what the precision of the registered data should be, we have to ask ourselves: *What are the world models built on the basis of the visual data going to be used for?* To answer this question, let us consider an example of the problem the proposed visual sensor will be applied to.

a) b)

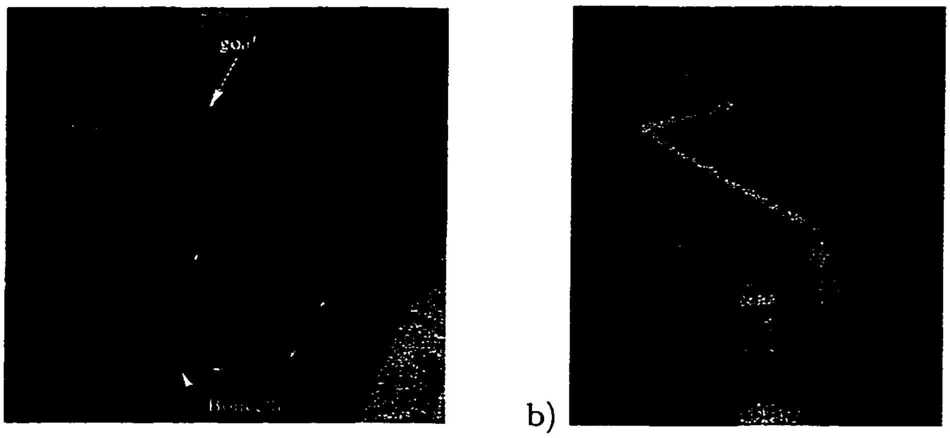


Figure 3.1: *An environment to explore (a) and a single videocamera range sensor used in exploration (b).*

We consider the task of exploring an environment for the purpose of locating a hidden target, as described in Section 2.2.2. Figure 3.1.a shows the room which is used as a testbed environment in the project. Starting from an arbitrary position, a robot has to find a target, which is chosen to be a corner of a green triangle glued to white paper seen on the back wall in the figure. The figure also approximately shows the field of view of the camera and the moves of the robot on its way to the target.

The decision how to explore is determined by 1) the knowledge of already observed obstacle points, 2) the knowledge of exploration points, i.e. points where no information is obtained yet, and 3) the knowledge of the target location, if available. This determines a three-module architecture of the robot, which we refer to as the "Look-Think-Drive" architecture. The entire architecture of this robot, which we call the robot Boticelli, is presented later in Chapter 6 and here we consider the first module of the architecture which is the vision module.

During the operation of the vision module, the robot tries to locate the target and collects range data around itself. The amount of these data should not be very large so as not to impede the mobility of the robot. On the other hand, it should suffice to build a precise enough 3D occupancy model of the world, where the precision is measured by the ability to navigate using the maps extracted from the model.

3.4 Stereo Rig Design

Many problems in world exploration by mobile robots are attributed to the odometric errors of the robot. Therefore, it is desirable to get as much information around the robot without having the robot move. This is achieved with a camera which has enough degrees of freedom to capture the entire environment.

While the idea of getting depth data with one camera seems straightforward: grab images from two different camera positions and triangulate, — it takes quite a few steps to design a good stereo rig. The problem is that there are a lot of ad-hoc parameters involved in such a design, such as the position of the camera, the angle and the direction of camera rotation, the resolution and the quality of the image and image preprocessing techniques used. These parameters frequently constitute the know-how parts of commercial products [Mar97], as they play a crucial role in the success of the design.

Figure 3.1.b shows the configuration of the camera-arm setup which has been selected by us as the most optimal for acquisition of depth data in the desired range. We describe it in more detail now.

In order to achieve stereo with one camera, we mounted it on an L-shaped support on a Direct Perception pan-tilt unit (PTU). This shape of the support was chosen to maximize the base line of the stereo, while making the stereo system compact, and allowing the robot to see close to itself. The angle and the length of the support are chosen in such a way that the camera can observe completely the part of the world from the floor to the height of the robot, within a range from one decimeter to infinity.

A grabber grabs 640x480 colour (RGB) images, which are then preprocessed with an averaging filter to produce 160x120 pixel images. The Matrox Imaging Library (MIL) is used as an image grabbing and processing tool. The resolution of the image is one of the important factors contributing to the success of the range data acquisition. The size of 160x120 of an image has been found optimal not only by us, but other researchers [JD97]: lower resolution results in losing too much visual information, while higher resolutions produce images which are very susceptible to changing light conditions and imperfections of the camera, and result in more calculations. Depth acquisition is done on these low 160x120 resolution images.

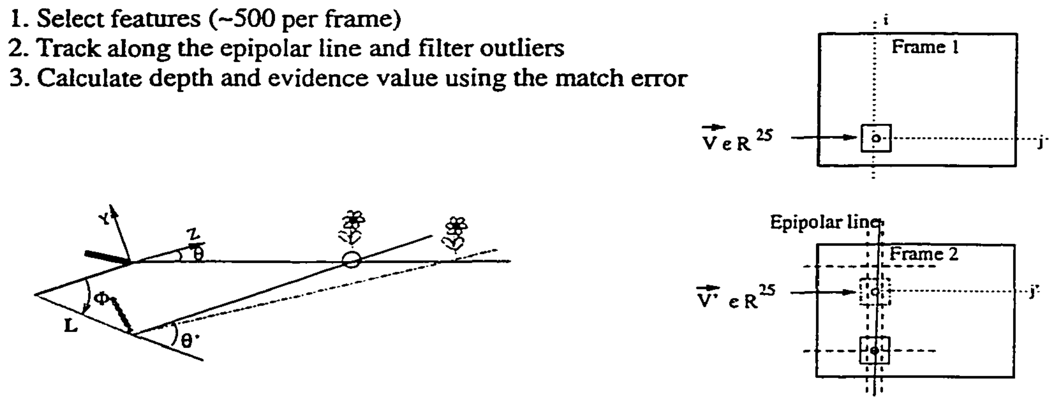


Figure 3.2: *Three-step depth registration procedure.*

3.4.1 3D Data Registration

The readings from a single camera stereo, which are 3D depth data, are obtained by the following three-step procedure (see Figure 3.2).

Step 1: A set of features is selected in the first frame;

Step 2: Each feature is tracked along the epipolar line in the second frame, which is grabbed after the camera has moved, and the best match is obtained;

Step 3: The depth to those features which are selected and successfully tracked is calculated on the basis of the disparity of the features in two frames.

We describe each of these steps in more detail now.

According to the objective to build a world model just good enough for exploration and in order to make fusion and world modeling faster, we select only about 500 features per image. The features are chosen so that it will be easy to track them. In particular, the pixels with a high intensity derivative in the vertical direction are selected as features. This choice of features is explained by the vertical motion of the camera.

The second frame is grabbed after the camera moves down. The angle of the camera tilt rotation (see Figure 3.2) is chosen as $\Phi = 7.7^o$ and the lever length $L = 21$ cm, which results in the baseline $h \approx 3$ cm. This produces the range of visibility of about four and a half meters, which is the same as in [JD97]. In the second frame, each feature is tracked along the epipolar line, which is chosen 3 pixels wide to account for warping of the image, using a 5 by 5 scanning window centered on a pixel. The equation of the epipolar line is derived in the Appendix.

The match error E is calculated as the Euclidean distance between the normalized N-dimensional vectors ($N = 25$) obtained by using the scanning window:

$$E = ||\vec{V} - \vec{V}'||^2 = \sum_{n=1}^{N}(V[n] - V'[n])^2, \tag{3.1}$$

which is a standard approach in feature tracking [GAL97]. The values $V[n]$ in the above formula are the intensities of pixels in the scanning 5 by 5 window minus the intensity of the pixel the window is centered on.

A feature is considered successfully tracked if error E between the best match and the original feature is lower than a certain threshold E_{thresh}. By lowering the threshold E_{thresh}, we can reduce the amount of uncertain data. This filters away approximately 60% of features.

Finally, the depth r to those features which are selected and tracked is calculated, using triangulation based on the projective camera model [Kan93]:

$$R\vec{m}'r' = \vec{m}r - \vec{h} \tag{3.2}$$

where $\vec{m} \doteq [i, j, F]_{unit}$ and $\vec{m}' \doteq [i', j', F]_{unit}$ designate unit vectors determined by the positions of a feature in the first and the second frame respectively. F is the focal length of the camera measured in rasters, which is known from the camera specifications or calculated in advance using the vanishing point technique described in [Kan93]. The focal length of the camera used in the experiments $F = 150$, and $i \in [-53, 53]$ and $j \in [-40, 40]$. R is the rotation matrix and $\vec{h}$ is the translation vector of the camera. Both are known, since only the pan-tilt unit moves and not the robot during depth acquisition.

Using the coordinate method (see Figure 3.3) and the fact that motion of features is close to vertical, formula Eq. 3.2 can be approximated by

$$\begin{cases} r' \sin(\Phi + \Omega - \theta') = Z \tan(\Omega - \theta) - h \cos\frac{\Phi}{2} \\ r' \cos(\Phi + \Omega - \theta') = Z + h \sin\frac{\Phi}{2} \end{cases} \tag{3.3}$$

where $\tan(\theta') = \frac{j'}{F}$, $\tan(\theta) = \frac{j}{F}$ and Ω is the angle of the camera support.

Dividing the first equation by the second one yields the formula for (X, Y, Z) coordinates of a feature in the coordinate system centered on the first location of the camera:

$$\begin{cases} Z = \frac{h\cos\frac{\Phi}{2} + h\sin\frac{\Phi}{2}\tan(\Phi+\Omega-\theta')}{\tan(\Omega-\theta) - \tan(\Phi+\Omega-\theta')} \\ X = Z\tan\theta \\ Y = Z\tan\theta_x, \quad \text{where} \quad \tan\theta_x = \frac{i}{F} \end{cases} \tag{3.4}$$

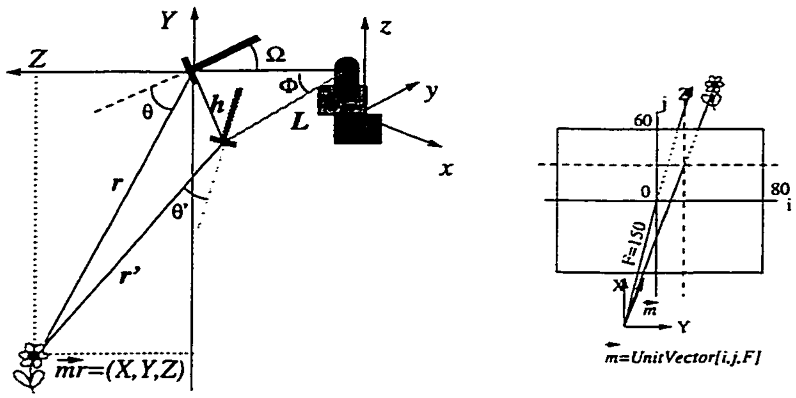


Figure 3.3: *A single camera range sensor: depth calculation procedure.*

To obtain the coordinates (x, y, z) of a feature in the PTU-centered coordinate system, vector $\vec{mr} = (X, Y, Z)$ is multiplied by a homogeneous matrix describing the current position of the camera, which is a function of camera pan and tilt angles.

After depths are calculated for the current pan angle of the camera, the camera is panned on the PTU clockwise and the procedure is repeated for the new angle of view, until finally all parts of the world around the robot are observed.

A thing to be mentioned about the single camera vision system is the parallelism of its operations — the depth is calculated, while the camera is moving. Because of that, the time needed to acquire depth information about the surrounding environment is just equal to the time needed to complete the full rotation of the camera. It takes 15 different pan positions of the camera to observe the whole environment and the whole process of a building a sparse depth map of the entire environment takes about one minute.

3.4.2 Searching for A Target

As opposed to a stereo setup with a fixed camera configuration [Mor96], a single camera stereo allows arbitrary motion of the camera. This gives more flexibility not only in tracking the features but also in searching for a target.

In this project, we are not concerned with the issue of target recognition. In our experiments we choose the target to be invariant to the distance so that it can be detected at any distance from the camera as soon as it is the field of view of the camera. This explains why we chose a corner of a green triangle on a white paper background (indicated in Figure 3.1.a with an arrow) as a target goal.

The target is sought by checking each image frame for the existence of a pattern previously stored in memory. MIL has a function which can do this operation efficiently. If the target is found, the same depth calculation routine which is used for features is used again to produce the location of the target with respect to the current position of the robot. This location of the target will later be used in making the navigation decision. This is described in more detail in Chapter 5 and here we just want to emphasize that single-camera stereo sensor allowed us to search for an object concurrently with acquiring range data, which we see as another advantage of this visual sensor.

3.5 Building Sensor Models

In sensor fusion, the concept of the *sensor model* is of prime importance. Using a probabilistic framework [vDKG95, MM96] the sensor model can be defined as the conditional probability

$$P(O_{\vec{r}}|\vec{r}_s) \tag{3.5}$$

that a point $\vec{r}$ in space is occupied, given a range sensor measurement $\vec{r}_s$.

It has been argued however that the probabilistic framework is not valid in building a sensor model when a sensor is not reliable [Voo95, PNDW95, MSG91]. For example, if a sensor works properly only three times out four (because of power failures or other problems), then a measurement $\vec{r}_s$, which, we may say, is 75% reliable, provides some information about the occupancy of a point, but it does not give any data about the negation, i.e. about the emptiness of this point.

Also, as described in Chapter 2, calculating conditional probabilities in Eq. 3.5 may result in calculating and storing huge conditional probability tables.

Therefore, another way of defining sensor model which uses the evidential theory has been suggested to circumvent these problems [Voo95, Wan94, PNDW95]. Rather than dealing with the probability that a point is occupied, the evidential approach considers two values of evidence: the evidence $m_{occ}(\vec{r})$ that a point is occupied and the evidence $m_{emp}(\vec{r})$ that a point is empty, which it calculates using the parameters describing the reliability of the measurement. These parameters are usually the functions of the intrinsic characteristics of the sensor.

The evidential approach has other advantages over the probabilistic approach which we will talk about later in Chapter 4. Below we show how this approach can be used for building visual sensor models.

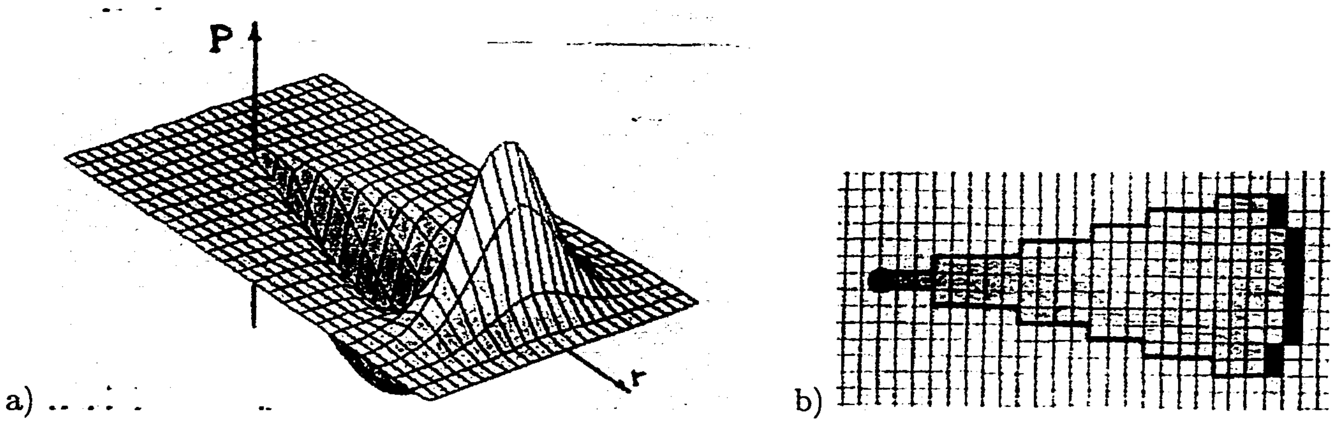


Figure 3.4: *Sonar sensor model according to probabilistic (a) and evidential (b) approaches.*

3.5.1 Sonar Sensor Model

In building a sensor model for a single camera stereo we utilize an approach which is similar to the approaches used by other authors for building sonar sensor models.

Let us see how sensor models are built for a sonar sensor. Figure 3.4 shows sensor models of a Polaroid sonar sensor built according to probabilistic and evidential approaches. These models are taken from [MM96] and [PNDW95], respectively.

In the first approach, the occupancy values are obtained using the Gaussian probability density distribution. In the second approach, the occupancy and the emptiness values are calculated as

$$
\begin{array}{ll}
m_{occ}(\vec{r}) = 1/n, m_{emp}(\vec{r}) = 0, & \forall \text{cells } \in \text{ arc, } n \text{ is the number of cells in the arc} \\
m_{occ}(\vec{r}) = 0, m_{emp}(\vec{r}) = 1/S, & \forall \text{cells } \in \text{ sector where } S \text{ is the area of the sector} \\
m_{occ}(\vec{r}) = 0, m_{emp}(\vec{r}) = 0, & \forall \text{ other cells}
\end{array} \tag{3.6}
$$

As can be seen, the evidential approach provides much simpler way of calculating the evidence values.

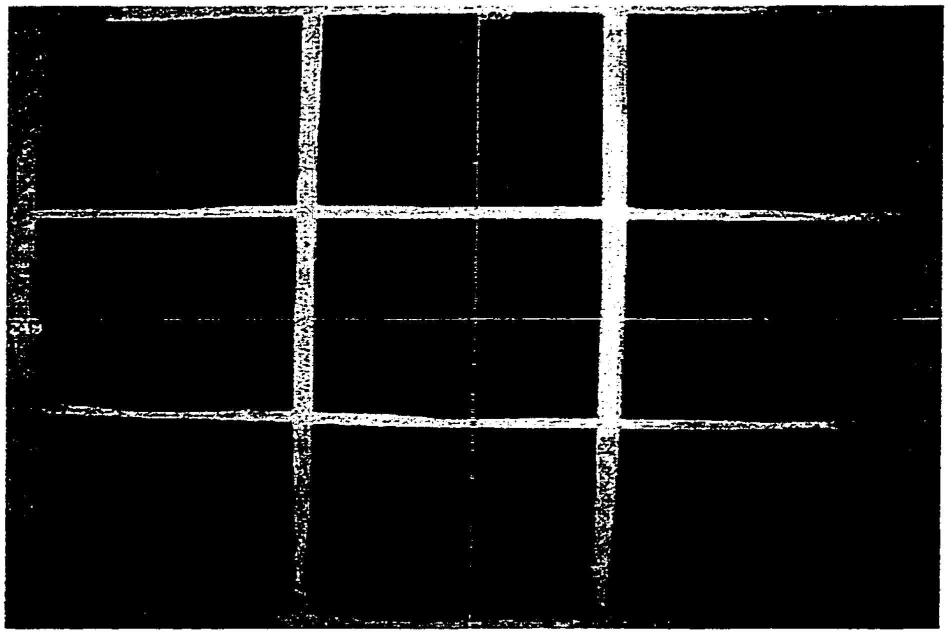


Figure 3.5: *The corruption of an image by a camera: Monochrome green rectangles as observed by a camera.*

3.6 Reasons for Uncertainty

As a result of camera warping, changing light conditions and incorrect registration of features, the 3D information obtained is not certain.

The best way to understand where the uncertainty comes from is to look at the Figure 3.5. It shows an image of monochrome constant-colour green rectangles as observed by our camera. The warping of the picture and different intensities of the same uniformly green colour can be clearly seen.

In addition to the imperfections of the camera, the observed objects themselves are a source of uncertainty. Figure 3.6 shows a 160x120 image of a camouflage cloth, the distance to which is going to be calculated, just before the tracking stage of the depth acquisition procedure. Many visual features look very much alike, when scanned by the scanning window, along the same epipolar line. This increases the chance of misregistration.

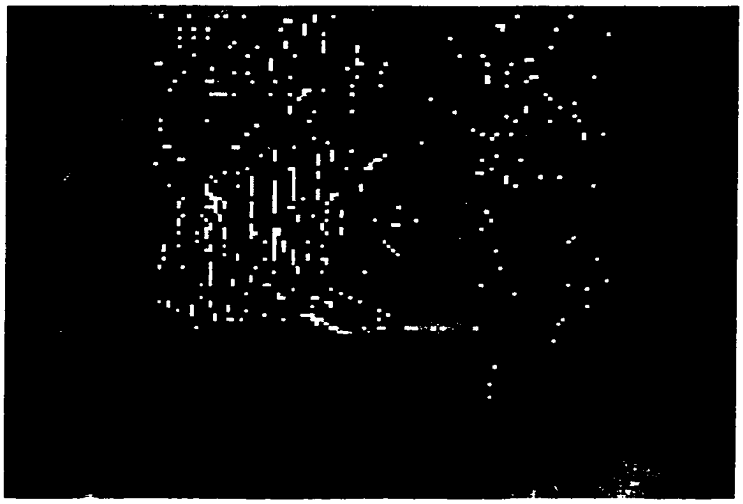


Figure 3.6: *A regular image observed by a camera - the one to be used for depth registration.*

These are two main reasons for imperfect tracking and matching of features, which, in turn, results in under- or over-estimating of depth values corresponding to the features. This is illustrated in Figure 3.7. The figure also shows another major reason for uncertainty in depth estimation — limited resolution of the camera. All these sources of uncertainty have to be taken into account when building a visual sensor model.

At the same time, in order to decrease the depth estimation error, we resort to the following two techniques. First, we disregard the marginal features (as in [YSA98, JD97]), since they introduce high error not only because of the image quantization but also because of the warping of the image. Second, when the robot views the surrounding world, we make sure that each selected feature is observed at least twice, so that it appears at least once in the middle of the image, where its error is low. This is achieved by adjusting the angle of pan rotation.

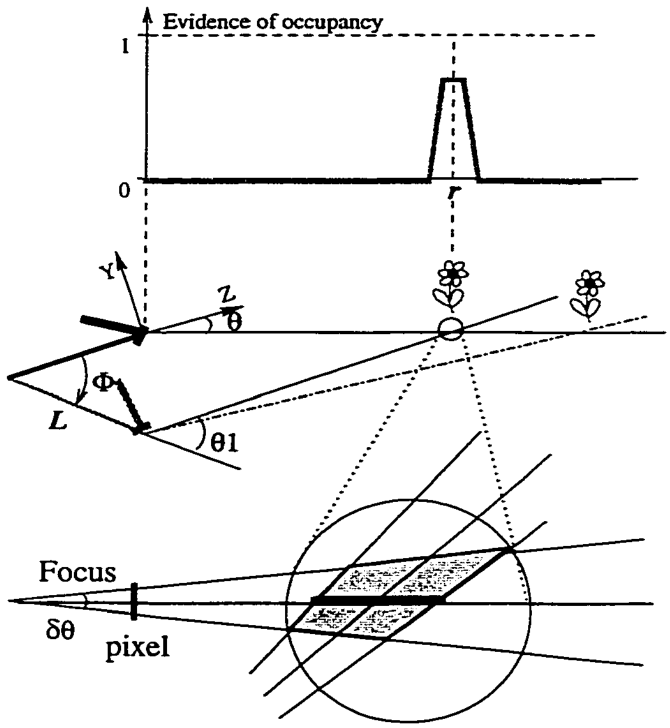


Figure 3.7: *Building a sensor model for a single camera stereo.*

3.7 Visual Sensor Model

Industrially manufactured sonar sensors [FBE94] and laser range finders [PHLG97] have well defined sensor models which are provided by a manufacturer. However, there is no general sensor model of a visual range sensor, which is due to the diversity of the visual system setups. Thus, we have to design our own model of the single camera range sensor.

3.7.1 Taking into Account Quantization Error

As mentioned above, the depth data obtained by a vision system is not certain for many reasons.

Due to the finite resolution of the image, the angle θ' in the Eq. 3.4 is known only with the precision $\delta\theta' = \frac{1}{F}$ (see Figure 3.7). This results in the range error δr, which can be estimated by taking a derivative of $r = (X, Y, Z)$ with respect to θ' in Eq. 3.4.

Another way of estimating the range error is to use the results obtained for nonconvergent dual camera stereo systems. The analysis of the uncertainty due to image quantization has been done in [BH87, MS87, RA90] and using the result obtained in [RA90], we get the following estimate of the range error:

$$\delta r = \frac{2r^2}{hF + r}, \tag{3.7}$$

where F is the focal length in raster units and h is the baseline of the stereo system.

3.7.2 Taking into Account Match Error

Calculation of the evidence values assigned to the registered range data is based on the following idea. If we are 100% confident in the range data, then the range data should get the evidence value one. On the other hand, if the sensor is completely unreliable, then the range data should get the evidence value zero.

In the case of the single camera stereo, the measure of confidence of registered depth data $\vec{r}$ is provided by the match error E obtained during the depth calculation procedure (Eq. 3.1). In particular, we obtain the evidence of a 3D point $m_{occ}(\vec{r})$, by applying the Tuckey by-weight to the error E:

$$m_{occ}(\vec{r}) = \begin{cases} (1 - (\frac{E}{E_{max}})^2) & \text{if } E < E_{max} \\ 0, & \text{otherwise} \end{cases}, \tag{3.8}$$

which is a common approach in robust estimation [MMRK91, RL87]. E_{max} is a constant which is chosen in agreement with the threshold value E_{thresh} used in filtering the outliers in Section 3.4.1

This approach is different from that of [MS97] and resembles that of [Mor96]. It produces the value of evidence in range [0,1], which can be used in fusing the range data.

3.7.3 Linear Representation

In the case of the ideal visual sensor, all points between the camera and the observed point will be given the evidence values $m_{emp}(\vec{r})$ and $m_{occ}(\vec{r})$, as illustrated in Figure

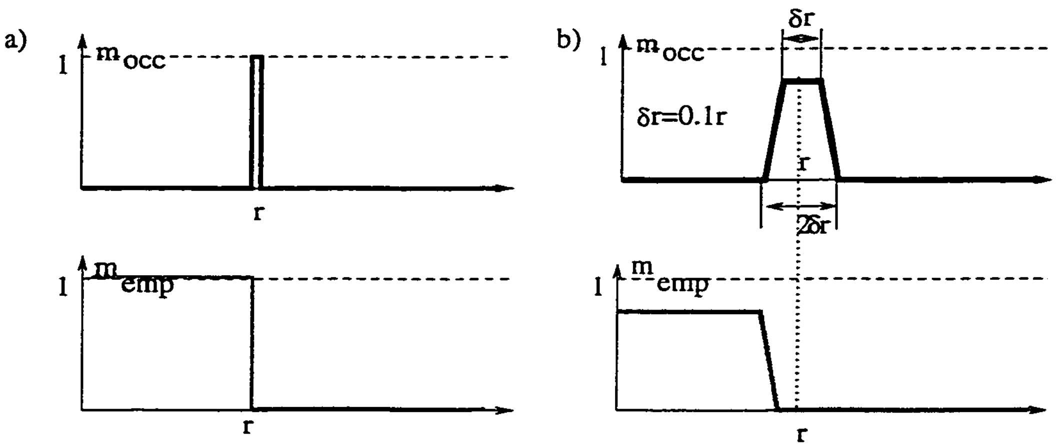


Figure 3.8: *Visual sensor model for ideal (a) and real (b) sensor.*

3.8.a. Figure 3.8.b shows the visual sensor model for a real visual sensor which is built according to the ideas described above.

The maximum value of evidence is determined by Eq. 3.8. The width of the range error δr is approximated using the Eq. 3.7 as $\delta r = 0.1r$. We also make the evidence grow gradually from zero to its maximum value, using the range error δr as a guide in determining the steepness of the slope, so as not to have infinite derivatives of the occupancy function. The evidence behind the observed point is zero for both occupancy and emptiness evidence values.

The piece-wise linear representation of the sensor model is chosen for two reasons. First, it facilitates the approximation of the occupancy function with linear surfaces. Second, it significantly reduces the amount of sample data used in fusion. In particular, the sensor model can be represented with only a few sample points on the ray of view, providing that there are certain constraints imposed on the function, which is described in more detail in the next chapter.

3.8 Experiments

The single camera stereo vision system described in the paper was tested using a mobile autonomous robot, Boticelli. The robot is placed in an approximately 5 by 6 by 2 meter room surrounded by walls. The robot has to explore the room in order to find a target which is hidden behind one of the walls. Figure 3.1.a shows the room and the target. Starting from an arbitrary location, the robot explores the environment until it finds a target. The exploration policy of the robot is determined by the knowledge of obstacle and navigation points, which are extracted from multiple 3D local occupancy models built on the basis of the range data registered by the single camera visual sensor, and also by the knowledge of the target location, which is acquired by the same sensor.

Figures 3.9.a and 3.9.b show the range data which are acquired by the robot with an aid of the single camera visual sensor by looking around from two different locations. Each figure consists of three windows. The window in left top corner shows an image observed by the single-camera sensor after it has been already preprocessed with an averaging filter. In this window we can see the environment being explored as observed by the robot. The window in left bottom corner shows pairs of 2D features used in depth calculation: shown in white are the features which were selected in the previous image (grabbed before the camera moved), while shown in black are the features which were successfully tracked in the current image (grabbed after the camera moved). Finally, the window at right shows registered 3D features projected on the floor (Oxy plane), with the robot located in the center. The evidence of the features is shown using colour: the brighter the feature, the closer its evidence to unity.

In order to ensure that there are enough visual features in the environment, we put camouflage cloths on the walls. These can be seen in Figures 3.9.a and 3.9.b. Other visible objects inside the exploration area include a small artificial tree (seen at left from the camouflage cloth in Figure 3.9.b), a couple of boxes (seen in Figure 3.9.a) and extension cords lying on the floor.

As can be seen from the figures, the depth maps obtained do not follow exactly the contour of the room, which is natural given the quality of the image and the complexity of the environment. However, these 3D range data are sufficient for world exploration purposes, and in particular for the problem of making intelligent navigation decisions on the way of finding an object in an unknown environment. In our experiments, the robot successfully locates the target while avoiding obstacles and the areas already explored, based on the 3D range data provided by the single camera range sensor. More on how visual range data are combined and used for world modeling is described in further chapters.

3.9 Conclusion

We introduced a single camera 3D range sensor, which allows one to register in real-time range data in all directions with the help of an inexpensive off-the-shelf video camera. We conclude that the proposed visual range sensor, which is able to register visual features around the robot in real time, is very suitable for mobile robot exploration. The advantages of the sensor are summarized below.

– It is inexpensive. Almost any camera, even if its images are not of high quality, can be used. This is because even unreliable features contribute to building an occupancy model though with less evidence value.

– It allows acquisition of range data in all directions around the sensor. This eliminates unnecessary motion of the robot which may result in odometric errors;

– It is fast. The time needed to build the depth map of the entire room is just equal to the time needed to make a full rotation of the camera;

– It has high angular resolution (outliers lie on the ray of view only), which, first, allows one to register data at any range and, second, is convenient for building a sensor model of the sensor;

– It can be concurrently used for surveillance purposes and range data acquisition;

– It has control over the number of registered range data. That is, instead of measuring depth to each point, which is done in sonar and laser-based range scanners, a selection technique can be used to select only a number of features which provide significant information about the occupancy of the environment. This not only speeds up range data acquisition, but is also beneficial for world modeling;

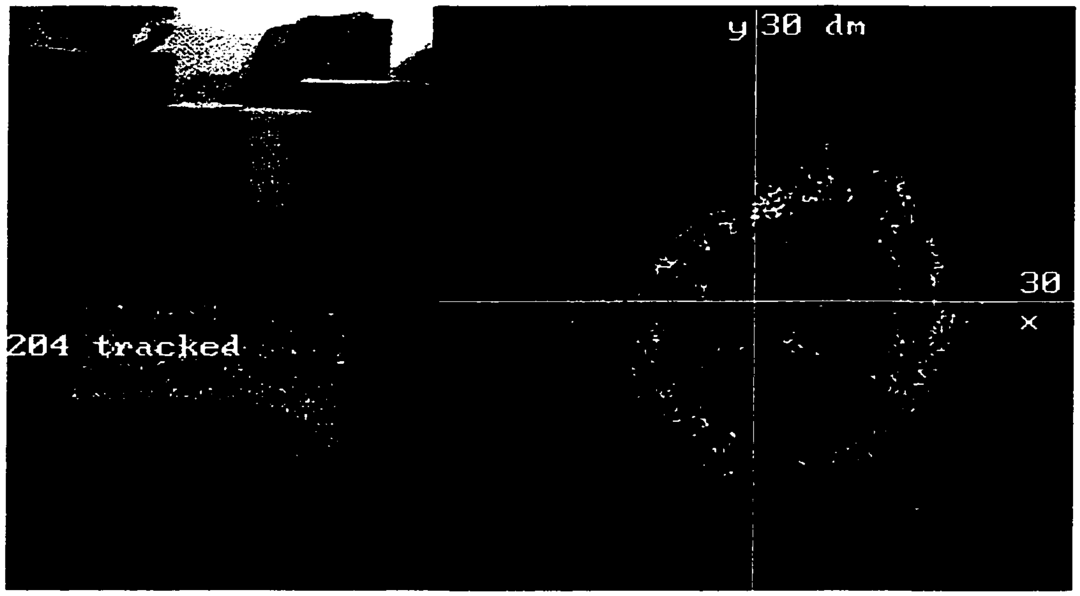
– It is convenient for designing a sensor model. Levels of uncertainty are associated with the registered data using the stereo error analysis in accordance with evidence theory.

We showed how sensor models of visual sensors can be built using the error analysis of the stereo system. We presented the stereo error analysis of the single camera sensor and built the sensor model of this visual sensor. The application of the visual sensor to mobile robot world exploration was shown. Our first argument in favour of using the evidence approach in vision-based world modeling has been used.

3.9.1 Further Improvements

The technique we use for feature selection and tracking, while simple and not time consuming, suffices for applications like the one described above. However, if there is a need for a more precise depth data registration, then the following steps can be undertaken to improve the performance of a single camera stereo: using better quality cameras; calibrating the camera and using all intrinsic parameters of the camera in depth calculation [BSG99]; rectifying images [Mor96]; using an interest operator to select features [Mor96]; using the epipolar constraint in filtering the outliers; using robust tracking approaches, e.g. like those described in [ML97, MMRK91].

As for the visual sensor model, a better approximation of the range error should be used for large scale environments. In addition, other approaches in assigning the evidence values to registered range data can also be tried. However, since the final map of an area available for navigation is determined by a threshold on an occupancy function, this assignment seems not to affect much the navigation planning process.

a)


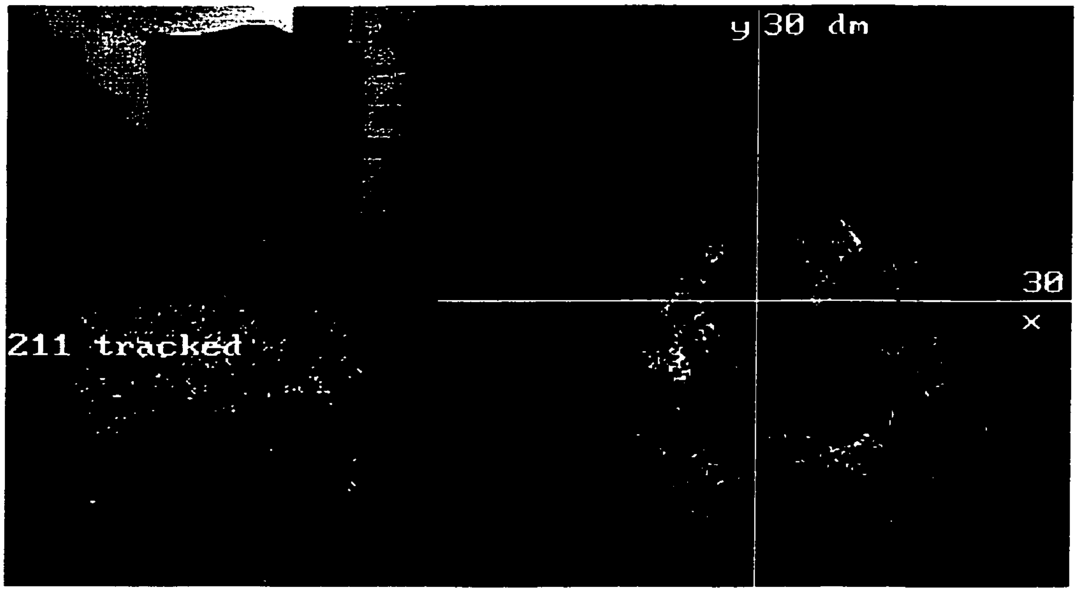
b)


Figure 3.9: *Depth data obtained by a single camera stereo (a,b) obtained at two different locations of the robot. The image grabbed, the features extracted and the depth map calculated are shown.*

Chapter 4

Combining Uncertain Range Data Using Regression

4.1 Introduction

In the previous chapter we addressed the problem of assigning the value of uncertainties to registered range data. Using the techniques proposed in that chapter, one can obtain 3D range data as a set of pairs: a 3D vector and a number describing the level of confidence associated with the vector. In this chapter, we expose another problem of the occupancy approach — the problem which comes next after range data have been collected. This problem deals with the combination of uncertain range data.

The rules used to combine uncertain data depend on the way the uncertainty is represented. There exist two paradigms for representing uncertainty. The first one is based on probability theory and the second one is based on evidence theory. Both paradigms are intensively used in robotics, and the question which one is better is widely discussed in the AI community [New99]. This chapter contributes to this discussion, as we argue in the favour of evidence theory when applied to fusion of range data.

After overviewing the previous work and describing the rules commonly used to combine range data, we show that, in certain cases, these rules can not be applied as they are, because of the assumptions made. These are the cases when the registered range data are not independent, one example of which is when the data are registered by a single-camera stereo.

We proceed to the proposal of a new range data fusion technique based on regression. This technique uses evidence theory in assigning occupancy values and builds the occupancy function by fitting the sample data provided by a sensor with a piecewise linear function.

Having developed a general framework for our approach, we apply it to building 3D occupancy models from visual range data. Both simulated and real visual data are used, where real data are obtained by using a single camera 3D sensor described in the previous chapter. The advantages of thus constructed piecewise linear models for robot navigation will be shown later in Chapter 5.

4.1.1 Previous Work

Each 3D point, which is detected either correctly or incorrectly by a sensor, provides a continuum of data for world modeling. In case of the visual sensor, it provides information about an occupied point – the detected 3D point itself, and unoccupied (empty) points – between the camera and the occupied point.

As was shown in Chapter 3, a visual sensor provides uncertain and sometimes contradictory data. The occupancy model of the world can be built by fusing these uncertain data in such a way that uncertainty is decreased by using data which over-determined the result.

As related to range data fusion, the problem of fusing uncertain data can be formulated as follows.

Fusion Problem: A sensor reading at time $t = 0$ shows that "point A is very likely non-occupied". On the other hand, according to the sensor reading at time $t = 1$, "point A seems to be occupied". And the questions are:
Is point A occupied or not? and *How sure are we in asserting that?*

This problem can be referred to as the *one-point-two-measurements* problem. It is the simplest version of a general fusion problem which can be referred to as the *multiple-points-multiple-measurements* problem and which is the following :

> Given a set of N sensor readings describing M different points A_α, tell which of points A_α are occupied and which are not.

The fusion problem is characterized as a multiple-agent two-state problem. That is, there are only two hypotheses in the domain of hypothesis: a point is either occupied or not, and there are many agents: as many as there are sensor readings in the domain of agents expressing their opinions about a two-state event. The solution to this problem is a *combination rule.*

There are several approaches dealing with combining uncertainties. These include Bayesian causal networks [Pea88, BS97], Markov fields [ACFM97], high-order probability theory and other probabilistic approaches [BCS97], fuzzy logic [Bal81], Kalman filtering [HM99], modal logic [HC96], epistemic logic [Voo95], partial probability [Voo97], connectionist networks [SD97], non-axiomatic reasoning [Wan94], non-monotonic reasoning [Voo95], Dempster-Shafer calculus [HM99] and other evidence-based and logic-based approaches [Yen89, BGHK92, FH95, GH97].

We focus our attention below on two types of approaches which are most commonly used for a multiple-agent two-state problem: probability-based (probabilistic) approaches and evidence-based (possibilistic) approaches.

4.2 Probabilistic Approaches

The most straightforward way of assigning the values of the occupancy function consists in assigning the value of 1 to an occupied point and the value of 0 to an empty point. Then, the closer the occupancy value of a point is to one, the more likely it is that the point is occupied. This approach is inspired by probability theory and this is the one originally considered by the founders of the approach [Elf86].

4.2.1 Bayesian Rule

Using the probabilistic framework, the one-point-two-measurements fusion problem can be reformulated as follows. A sensor reading gives the uncertainty of a registered point in terms of conditional probability that a point is occupied, given a range measurement r:

$$m \equiv P(o|r) \tag{4.1}$$

The values of uncertainty therefore satisfy the main probability axioms, such as:

$$0 \leq m \leq 1 \quad \text{and} \tag{4.2}$$

$$P(o|r) + P(\neg o|r) = 1, \tag{4.3}$$

where $P(\neg o|r) \equiv P(e|r)$ is the probability that a point is empty, given a range measurement r, and 'o' and 'e' designate the events that a point is occupied and empty respectively.

The fusion problem then becomes a problem of finding the probability of the event that a point is occupied conditioned on two range measurements r_1 and r_2:

$$m_{1+2} \equiv P(o|r_1, r_2) = f(m_1, m_2). \tag{4.4}$$

Since the probability axioms hold for the occupancy values, this probability can be found using the Bayesian rule as

$$m_{1+2} = P(o|r_1, r_2) = \frac{P(r_2|o, r_1)P(o|r_1)}{P(r_2|o, r_1)P(o|r_1) + P(r_2|e, r_1)P(e|r_1)} \tag{4.5}$$

In this formula, $P(o|r_1) = m_1$ and $P(e|r_1) = 1 - P(o|r_1) = 1 - m_1$ are known — they are simply the measurements of the first sensor reading. However, the values $P(r_2|o, r_1)$ and $P(r_2|e, r_1)$ are not easy to calculate and this is where the problem comes.

4.2.2 Simplifications: Making Assumptions

Applying the combination rule 4.5 requires the knowledge of conditional probability tables: $P(o|r)$ and $P(r_2|o, r_1)$, which have to be computed for all possible range values r_1 and r_1. In general case, this is practically impossible. Therefore, simplifying assumptions have to be made.

Independence of noise assumption

The most commonly made assumption is *the independence of noise in different sensor readings.* In terms of conditional probabilities this assumption can be written as

$$P(r_2|r_1, o) = P(r_2|o). \tag{4.6}$$

or as

$$P(r_1, r_2|o) = P(r_1|o)P(r_2|o). \tag{4.7}$$

With this assumption made, one can rewrite formula 4.5 as

$$m_{1+2} = P(o|r_1, r_2) = \frac{P(r_2|o)P(o|r_1)}{P(r_2|o)P(o|r_1) + P(r_2|e)P(e|r_1)} \tag{4.8}$$

This formula became well-established in data fusion [ME88]. However, we can see that it is still not ready to be used until conditional probabilities $P(r|e)$ and $P(e|o)$ have been calculated.

In a probabilistic framework, conditional probabilities $P(r|e)$ are $P(e|o)$ have to be calculated from the sensor model. This is difficult and time-consuming. Special techniques using neural networks, which are applied in advance in the predefined environments, have been proposed to do that [vDKG95, TB96]. This is why further simplifications of the combination rule have been proposed.

In the original work [Elf86, ME88], Elfes assumes $P(r|e) = 1 - P(r|o)$, and uses the following combination rule:

$$m_{1+2} = P(o|r_1, r_2) = \frac{P(r_2|o, r_1)P(o|r_1)}{P(r_2|o)P(o|r_1) + (1 - P(r_2|o))(1 - P(o|r_1))} \tag{4.9}$$

Another popular combination rule is the *independent opinion rule*, which is written as follows:

$$m_{1+2} = \frac{m_1 m_2}{m_1 m_2 + (1 - m_1)(1 - m_2)}. \tag{4.10}$$

As opposed to the linear opinion rule, which can be written, as

$$m_{1+2} = w_1 m_1 + w_2 m_2, \quad \text{where} w_1 + w_2 = 1 \tag{4.11}$$

the independent opinion rule reinforces the confidence and is closer to the original rules of Elfes and Moravec [Elf86, ME88, ME85]. At the present moment, this rule is the most commonly used in range data fusion. In particular, Payeur and D. Laurendeau [PHLG97] use this rule for building their 3D occupancy models.

Variations

Formulas 4.9 and 4.10 are computationally expensive. Therefore, a number of variations have been proposed.

A modification to the Bayesian approach described above was suggested by Konolige in 1995 [IK98], who suggested to use *odds-likelihood posterior* as a measure of uncertainty:

$$m \equiv \frac{P(o|r)}{P(e|r)} \tag{4.12}$$

rather then posterior probability $P(o|r)$ only. In this case, the values of occupancy range from 0 (absolutely impossible) to $+\infty$ (absolutely true) and the combination rule is the following:

$$m_{1+2} = \frac{P(r_2|o, r_1)P(o|r_1)}{P(r_2|e, r_1)P(e|r_1)}, \tag{4.13}$$

which after making the independence from noise assumption becomes

$$m_{1+2} = \frac{P(r_2|o,)}{P(r_2|e)} m_1 \tag{4.14}$$

A similar approach was used later by Moravec [MM96, Mor96], who considered the occupancy defined as

$$m \doteq log \frac{P(o|r)}{P(e|r)}$$

and who derived the combination rule from the independent opinion rule as

$$m_{1+2} = log\frac{P(o|r_1)}{P(e|r_1)}\frac{P(o|r_2)}{P(e|r_2)} = m_1 + m_2 \tag{4.15}$$

It is this combination rule which is used by Moravec in his latest research in building 3D evidence grids [Mor96].

4.2.3 Extended Kalman Filter

The above described approaches use only the first moments of an unknown variable, which are occupancy values of the points. Estimating second moments (i.e. deviation and the covariance matrix) of the variables would lead us to the Extended Kalman Filter [May79].

In Kalman filtering the uncertainty about the environment is propagated by means of estimating the first two moments of probability distribution of the spatial variables of the environment.

Whereas the Extended Kalman Filter has been intensively used for beacon-based world modeling approaches [Mit94, Fau93, Aya91, WHA92], it is not used in building occupancy models of the world. This is explained by a huge dimension that would be required to update the state vector of the world which is made by concatenating vectors associated with *each* voxel of the world.

4.2.4 "Unknown *vs* Contradictory" Problem

In the case of the Bayesian approach presented above, if the probability of point A to be occupied $P(o) = 1/2$, then it is not clear whether this is because there is no enough information about the point (and hence more exploration in this area is required) or because the data obtained about the point are contradictory (meaning that the environment is to complex there and extra exploration in this area most likely will not help). This is called the "unknown *vs* contradictory" problem [Voo95].

Calculating second moments of the occupancy values of the points would elucidate an ambiguity and resolve the problem. However, computational expenses incurred by such calculations would make the probabilistic approach unsuitable for robotics, and this is why the second moments are not used so far in building the occupancy models.

The "unknown *vs* contradictory" problem is a reason to consider evidence-based approaches as an alternative to probability-based approaches.

4.3 Evidential Approach

As described in Chapter 3, a way of simplifying the assignment of evidence levels to range data is to use the evidential approach, which operates with *evidence masses*, also called *basic probability assignments* in assigning occupancy values [Voo95, PNDW95]. Instead of computing probabilities $P(r|o)$ and $P(r|e)$, required by probabilistic approach, this approach assigns evidence masses to a point using sensor parameters that describe the robustness of the sensor. This assignment is more efficient than summing conditional probabilities required by the probabilistic approach.

According to the evidential theory [Voo95], the likelihood of a point A to be occupied is determined by two functions, called *belief* and *plausibility* functions. These functions can be considered as a pessimistic and an optimistic estimate of a point to be occupied and are related to evidence masses by

$$Bel(o) \doteq m(o), \quad \text{and} \tag{4.16}$$

$$Pl(o) \doteq 1 - m(\neg o), \tag{4.17}$$

The evidence masses satisfy the condition

$$m(o) + m(e) + m(o,e) = 1, \tag{4.18}$$

and are determined by a sensor model. The value $m(o,e) = 1 - m(e) - m(o) = Pl(o) - Bel(o)$ defines the interval of uncertainty and shows our level of confidence in asserting that an event is true. It is an advantage of the evidential approach that it provides this interval of uncertainty within which the occupancy value lies.

4.3.1 Dempster-Shafer Rule

Assume that the reading of a sensor tells us: "The point A is occupied". Does this sensor reading decrease the probability of point A to be empty? According to the Bayesian approach — yes. According to the Dempster-Shafer theory of evidence — no. This is because the sensor reading provides no evidence for point being empty (it has only the evidence for point being occupied).

In the evidence theory, the evidence masses of two different measurements m_1 and m_2 are combined using the Dempster-Shafer rule of combination, which can be considered as an extension of the Bayesian rule [HM99, Voo95]:

$$m_{1+2}(o) = \frac{m_1(o)m_2(o,e) + m_2(o)m_1(o,e) + m_2(o)m_1(o)}{1 - m_1(o)m_2(e) - m_2(o)m_1(e)} \tag{4.19}$$

In this formula, values $m_i(o) = m(o|r_i)$ and $m_i(e) = m(e|r_i)$ are the values of evidence that a point is occupied and that a point is empty respectively and both are provided by a visual sensor.

To apply the Dempster-Shafer combination rule one has to assume the *DS-independence* of the evidence data [Voo95, Wan94], which, roughly speaking, can be defined as probabilistic independence of sources of these data. This is not realistic in many cases; yet because of the convenience of using evidence theory in designing sensor models and because of the "unknown *vs* contradictory" problem, this rule is extensively used in mobile robotics [MSG91, WSSB93, PNDW95].

It should be mentioned that in the above references, the Dempster-Shafer combination rule is applied to fuse data obtained from sonar sensors. This is due to the fact that, in the case of sonar sensors, basic probability assignments $m(x)$ are easy to calculate, which is not true for other types of sensors. This explains also why the Dempster-Shafer rule has been applied for building two-dimensional occupancy grids only.

4.3.2 The Problem

As will be shown later, because of the DS-independence assumption, which does not hold in many cases, the evidential approach may produce nonsensical results when the Dempster-Shafer rule or any of its approximations is used to fuse range data. Therefore, it is desirable to suggest another rule for combining evidential range data which, while using the advantages of evidential approach, would handle dependent data.

4.4 The Assumption of Independence Problem

As shown in the previous section, the assumption that range data are independent needs to be made in all of the described above approaches in order to simplify the combination rules and to make the approaches computationally tractable.

Let us show now why this assumption can sometimes lead to very undesirable results. For this we are going to use a vision-based world modeling example and a single-camera visual range sensor described in Chapter 3.

4.4.1 Vision-based World Modeling Example

In Chapter 3 we introduced a single-camera sensor, which is an affordable tool for registering 3D range data. This visual sensor allows one to register 3D range data of the surrounding environment in real time. This makes a single-camera visual sensor very useful for autonomous robot world exploration. The way this sensor is used for this task is the following.

The single camera visual sensor is mounted on the top of a robot as shown in Figure 4.1 allowing the robot to "see" what is around it. Along with a 3D vector $\vec{r_i}$ of a feature measured in the robot-centered system of coordinates (see Figure 4.1), the sensor also provides the evidence value m_i of the feature. This evidence value is determined by the match error obtained during stereo acquisition and is a value between zero and one.

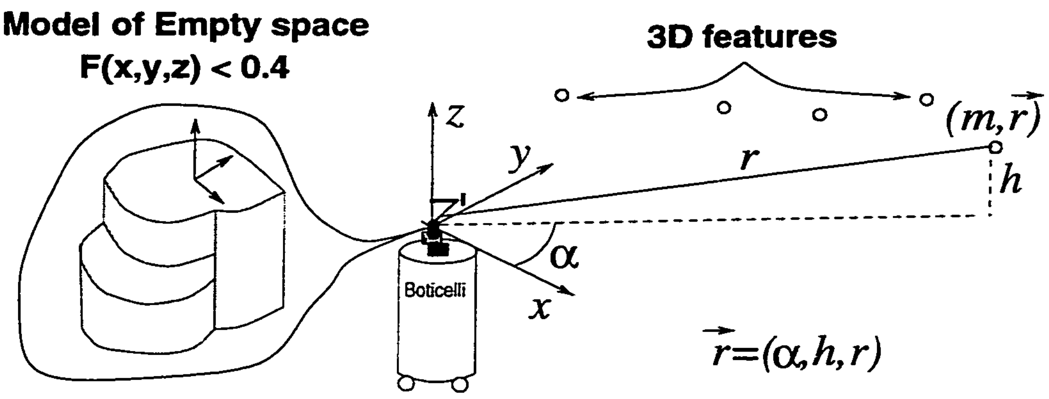


Figure 4.1: *Vision-based world modeling.*

4.4.2 When Features Are Not Good

In vision based world modeling, the observed features are not perfect due to the absence of any constraints on the environment and the low quality of the image. Because of that, most of evidence values of the features lie in a range between 0.5 and 0.8 ($0.5 < m < 0.8$). At the same time, the same scenery is observed repeatedly by a camera. What it means is the following.

If a feature is not good, which is likely a result of an indistinct visual pattern or poor lighting condition in the observed area, and its confidence level is low when observed the first time, then it is obvious that the feature will also have a low confidence level when observed the second and the third times. Let us see now what it means for world modeling based on these visual data.

Consider the combination rule 4.10 which is the one most commonly used in the situation. If the same point in space is observed twice by the same sensor (or by two similar sensors) from the same position (or from very close positions), it means that we will have two identical (or almost identical) evidence values:

$$m_1 = m_2. \tag{4.20}$$

Substituting Eq. 4.20 into combination rule 4.10, results in increasing of the total evidence value of that point:

$$m_{1+2} > m_1. \tag{4.21}$$

For example, if $m_1 = m_2 = 0.6$, which is a typical situation in our experiments, then $m_{1+2} \simeq 0.7$.

This result is nonsensical, as no extra evidence has been acquired, yet our confidence in data, which is in fact bad data, increased. This happens because of the assumption that range data are independent, which is not true, and this may have very bad consequences on navigation decision making.

We could have considered any other combination rule of those described above (Eqs. 4.8, 4.19, 4.13 or 4.15), but it does not matter which rule we use to illustrate the point, since all of them assume the independence of range data from noise.

Thus, there is a need for another combination rule which would handle dependent data.

4.5 Fusion As Regression

Though applied in many other areas [RL87], regression techniques have not been used yet for range data fusion. This is attributed to some unresolved questions which we address below. Let us first formulate the problem of building the occupancy function according the regression paradigm.

Problem:

> *Given a set of sample points $\{m_k, \vec{r}_k\}$ provided by a sensor, find a smooth approximation of the function $m = F(\vec{r})$ on the whole input domain.*

The fact that regression is used for fusion means that the approximating function $\hat{F}(\vec{r})$ is calculated by minimizing the error

$$E = \sum_k w\Big(\hat{F}(\vec{r}_k) - m_k\Big), \tag{4.22}$$

where the weighting function $w(\cdot)$ and the form of the function $\hat{F}(\vec{r})$ depend on the regression technique used. Before deciding which regression technique to use, some questions should be answered, the main one of which is the following:

What are the sample points $\{m_k, \vec{r}_k\}$ *to be used in regression?*

4.5.1 Generating Samples

Besides an observed feature i, the sensor provides occupancy information about points in the vicinity of the feature $\{m_i^j, \vec{r}_i^j\}$ according to *the sensor model*. The sensor model of the single camera stereo was described in Chapter 3.

The conventional range data fusion approaches described in the previous sections do not make use of the fact that many sample points are dependent on each other. In particular, all points $\{m_i^j, \vec{r}_i^j\}$ induced by a feature i are functions of the feature values $\{m_i, \vec{r}_i\}$. In those approaches, the sensor model is incorporated into the fusion process by using sample points generated according to the grid resolution (Figure 4.2.a). This results not only in redundancy of processed data, but also in an inability to deal with range data distributed over a large range.

We show now how this can be efficiently done in the regression-based fusion approaches using the dependencies between sample points. In order to make fusion fast, we want to use as few sample points as possible. This can be achieved by applying the following two rules.

Rule 1: Impose constraints on the occupancy function using

1. the knowledge of a sensor model;
2. the knowledge of the task the occupancy model will be applied to.

For this to be efficient, choose the system of coordinates which best suits the sensor model and the navigation task.

Rule 2: Consider sample points as additional constraints on the occupancy function

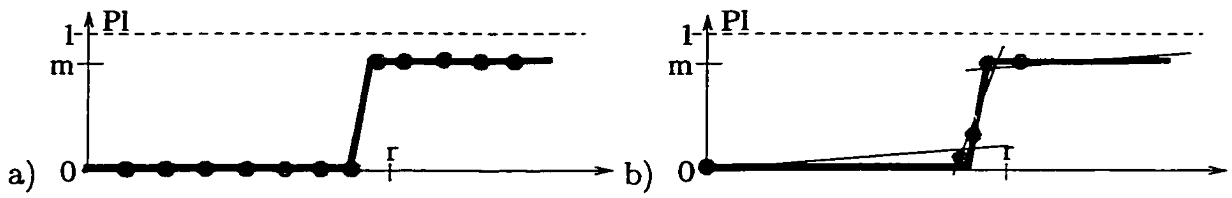


Figure 4.2: *The visual sensor model sampled according to (a) grid-based and (b) regression-based techniques.*

and generate just as many of them as needed to preserve the shape of the sensor model.

4.5.2 Choosing Coordinate System and Constraints

Which coordinate system to use for building an occupancy function depends, first, on the nature of range data, and second, on the way the occupancy function is going to be used. Let us illustrate this.

If range data is radially-based and the occupancy monotonically increases away from the origin, then spherical coordinate system (α, ψ, r) is the most appropriate one. This is the case when time-of-flight, laser-based or infrared sensors located at the origin are used. This can also be considered the case, when a single-camera visual sensor is used, if the scale of the environment is much bigger than the length of the camera arm.

The spherical coordinate system provides an easy way of constraining the function along the r-axis:

$$\frac{\partial F(\alpha, \psi, r)}{\partial r} > 0. \tag{4.23}$$

This constraint allows one to reduce significantly the number of sample points used in regression, while not adversely affecting the quality of the modeling. This is illustrated in Figure 4.2, which shows the sample points used in building the plausibility occupancy function according to grid-based and regression-based approaches. Drastic reduction in the number of sample points needed in regression can be seen. In particular, as illustrated in the figure, using the monotonicity constraint (Eq. 4.23), the sensor model can be expressed with as few as five sample points per feature (see Figure 4.2.b): two with evidence m_i to account for error in depth calculation, two with evidence less than m_i to correspond to decreasing occupancy values along the ray towards the camera, and one in the center of the robot with the evidence value equal to zero. The specific locations of these points are defined by the parameters of the sensor model. For comparison, by ignoring the fact that many sample points used in regression are functions of other points, conventional approaches generate as many sample points as there are grid cells between an observed point and the camera as shown in Figure 4.2.a.

The spherical coordinate system is best suited for building precise 3D occupancy models. However, it is inconvenient for planar map extraction and straight line navigation, for which the cylindrical coordinate system (α, h, ρ) is more appropriate. The monotonicity constraint $\frac{\partial F(\alpha,h,\rho)}{\partial \rho} > 0$ can also be imposed in the cylindrical coordinate system, where ρ is the distance from the vertical axis of the robot. In this coordinate system however it implies that it is not possible to go past the obstacle.

The cylindrical coordinate system is suited the most for such navigation commands as "go in direction α, ρ meters". It also allows one to easily examine navigation maps at different heights.

The coordinate system we use in this dissertation is yet another one, which can be considered as a hybrid between spherical and cylindrical coordinate systems. This coordinate system, which we refer to as the quasi-cylindrical coordinate system, is illustrated in Figure 4.1 and uses the pan angle α (radians), the height h (decimeters), and the distance r from the robot (decimeters).

The choice of this system is determined by the desire to suit the single camera sensor models while making it easy to extract horizontal surfaces needed for navigation.

In addition to the monotonicity constraint, which can be written now as

$$\frac{\partial F(\alpha, h, r)}{\partial r} > 0, \tag{4.24}$$

and which makes the construction and the inversion of the occupancy function easier, a set of other constraints can also be imposed. For example, we also impose constraints

$$-1 < \frac{\partial F}{\partial \alpha} < 1 \quad \text{and} \quad -1 < \frac{\partial F}{\partial h} < 1, \tag{4.25}$$

in order to prevent abrupt changes of occupancy values along the h- and α- axes.

4.6 Choosing a Regression Technique

The choice of the regression technique to be used for fusing range data depends on two factors: 1) the use of the constructed model, and 2) the quality of range data. The considerations to be taken into account are the following

Time Efficiency Consideration

From the efficiency point of view, piecewise linear regression techniques are preferable to others: they are fast and it is easy to impose the constraints of Eq. 4.24, 4.25 on piecewise linear functions. Piecewise linear functions are also easy to invert, providing they are monotonic in one of the variables.

The Issue of Outliers

Robust regression techniques, like Least Median of Squares, Least Trimmed Squares, Weighted Least Squares or M-estimators [RL87, CT94, BW88] can handle data which have many outliers. However, they are usually slow, as they do not allow an on-line calculation of error E in Eq.4.22. On the other hand, Least Squares techniques are fast [RT95]. They however are very susceptible to outliers, and because of that extra care has to be taken to get rid of outliers in the preprocessing stages if Least Squares regression techniques are used to fuse range data.

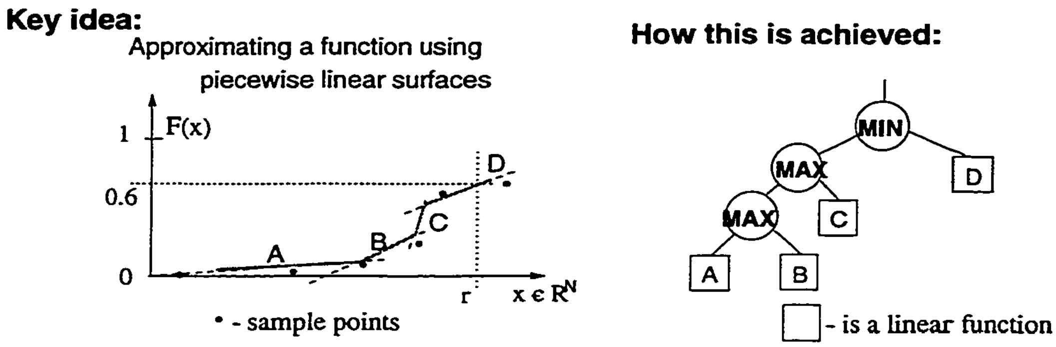


Figure 4.3: *The idea of ALN: Approximating a function with linear surfaces.*

4.7 Adaptive Learning Network Regression

4.7.1 Semantics of Adaptive Logic Networks

The *Adaptive Logic Network* (ALN) originally proposed in 1974 and subsequently improved [AT96a] is a tool for approximation of any continuous, real-valued function $y = f(\vec{x})$ on N-dimensional space given a set of its sample points $\vec{x}^m \in \Re^N$, $y^m \in \Re$, $m = 1..M$ (see Figure 4.3.a).

The main difference and advantage of the ALN over other approximation techniques is that it utilizes piecewise linear surfaces. Because only a few pieces are involved in computing a particular output, as may be determined using a decision tree on the components of the input, a considerable speed-up in processing is obtained. Control over the weights of the pieces can also lead to good generalization [AT96a]. The way the pieces are put together is determined by a logic tree which classifies (x, y) pairs according to whether they are below a function graph or not, or by an equivalent tree of maximum and minimum operators (Figure 4.3.b) acting on linear functions, which computes a y given x. To smooth out the corners at the junctions of linear pieces, quadratic fillets are used.

The training of an ALN consists of many multiple linear regressions working in concert with the goal of fitting the training data with low error, where error is computed by summing the squares of the distances between the approximating surface and output values. Formally it can be described as follows, where L is the number of hyperplanes.

Given *M sample points* $(\vec{x}^m, y^m)$, $\vec{x}^m \in \Re^N$, $y^m \in \Re, m = 1..M$, and *acceptable error level* ϵ find *coefficients* $W_p, W_{p1}, .., W_{pN}$ such that *hyperplanes*

$$y = W_p + W_{p1} * x_1 + ... + W_{pN} * x_N, \quad p = 1..L$$

pieced together according to the tree of maxima and minima, approximate the training samples with the root mean square (RMS) error less than ϵ.

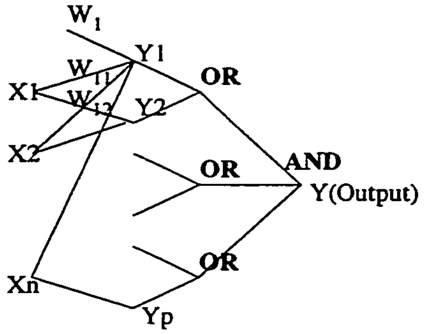
The examination of the ALN then is:

Given a new pattern vector $\vec{x}^{new}$, find $y = F(\vec{x}^{new})$, where $F(\vec{x})$ is the result of evaluating the tree of maximum and minimum operations on the linear functions given by the hyperplanes.

The most distinctive feature of the ALN is the way it combines hyperplanes. The ALN combines hyperplanes by introducing maximum and minimum operators (MAX and MIN) as shown in Figure 4.3.b. Direct application of MIN (MAX) to half-spaces produces convex-up (down) surfaces. There can be any number of these operators and they can be nested.

The MIN operators and MAX operators that take inputs from linear functions or other operators make up an *ALN logic tree.* There are two ways of building a logic tree. The first way consists in manual setting of its depth. In the second way however the logic tree is built automatically by the ALN, the depth and complexity of it being determined by the *acceptable error level* ϵ, which defines how precisely we want a function to be approximated. One starts with one linear piece, and only breaks pieces if, after adjustment of weights, the RMS error is greater than ϵ. This is a greedy approach to minimizing the number of linear pieces in the ALN.

The structure of the ALN can be represented as shown in Figure 4.4. In such a representation, an ALN looks very similar to a multilayered neural network, such as a feed-forward multilayer perceptron, for example. The main difference is only that



$$\vec{x} \in \Re^N$$
$$\vec{y} = \mathbf{W}\vec{x} \in \Re^P$$
$$Y \in \Re$$

Figure 4.4: *ALN from the Neural Network paradigm point of view.*

logical comparisons are used instead of threshold functions and weighted connections. This is the reason why ALNs are often considered as a type of neural network.

4.7.2 Advantages of ALN

Many of the currently used neural networks are treated as "black boxes", which are trained and then used. Because of this, the solutions these neural networks yield are often far from optimal and they cannot be guaranteed to obey common sense constraints. However, this is not the case for ALNs, because they provide a clear understanding of what is being computed and how. This may allow ALNs to perform better than other neural networks on certain types of problems where direct geometrical constraints on linear pieces are useful. For example, placing bounds on the slopes of pieces helps the ALN to solve the "two spirals" problem with good generalization in a few seconds on a Pentium PC, while back-propagation has great difficulty in obtaining any solution [AT96a] in real-time.

Comparing to more advanced neural networks like attractor-based networks [GR97] or GMDH [MI94] networks, ALNs are considerably faster than the former in examination and considerably faster than the latter in training, which makes them very suitable for real-time applications.

A summary of the main features of the ALN follows.

- It possible to integrate into the training procedure qualitative knowledge about the desired function, e.g. slope bounds and convexity constraints; this may aid generalization and prevent a learned function from having nonsensical properties. It also tends to reduce the amount of training data required;

- It is easier to understand the function that ALN is computing compared to other neural networks since it is made up of linear pieces put together in a simple way.

- Generalization can be aided by using jitter, whereby additional training data is generated by adding random amounts to the components of the x-vector.

- ALNs are very fast in evaluation. In fact, the decision tree approach enables the system to narrow the computation to about $N+1$ linear pieces, where N is the dimension of the input space. ALNs are rapidly trained too by using $\alpha - \beta$–pruning as in game playing.

- in cases where the inverse of a function exists, it is possible to extract it directly from the ALN using the same linear pieces combined differently as another ALN decision tree.

As can be seen ALNs can be applied to any problem where the data have some regularities, or in other words, are determined by some function. So far the ALN has been successfully applied in predicting financial and economic events, machine failure, and real-time control including rehabilitation of patients with spinal cord injury [ACT95, AKST95] and face recognition [GAL97].

Let us describe now how these advantages of the ALN can be used to compute the occupancy function from range data.

4.7.3 Applying ALN to Fusion

Function representation

As described above, the ALN uses a Least Squares linear regression technique and produces a binary tree of minimums and maximums of linear functions as a result of the fitting process. That is, in the chosen system of coordinates (see Section 3.3), the form of the function approximating the occupancy function $m = \hat{F}(\alpha, h, r)$ can be written as

$$m = tree_{l=1...L}^{(MIN,MAX)}\{a_l\alpha + b_l h + c_l r + d_l\}, \tag{4.26}$$

where L is the number of linear pieces used in regression. It determines how close we want the occupancy function to be to sample points and can be either set in advance or obtained during the regression process. In the latter case, the ALN grows and linear pieces with greater than the specified error are split in two, until either the allowable error E in Eq. 4.22 is achieved or a predefined number of iterations is completed.

By setting the number of iterations or the number linear pieces used in regression, we determine how close the approximating function $\hat{F}(\vec{r})$ lies to sample points.

The min-max tree structure of the ALN allows also very efficient inversion of the occupancy function, which is beneficial for map extraction and navigation planning, as will be shown in the next chapter.

Constraints

As follows from Eq. 4.26, the constraints on function $\hat{F}(\vec{r})$ described in Section 3.1 (Eqs. 4.24,4.25) become simply the constraints on parameters a_l, b_l and c_l:

$$c_l > 0, \quad \text{and} \tag{4.27}$$

$$-1 < a_l < 1, \quad -1 < b_l < 1. \tag{4.28}$$

This is one of the advantages of the ALN that it provides a very efficient way of constraining the function being learnt.

Getting rid of outliers

Since ALN uses Least Squares error in fitting the function, extra measures have to be undertaken to eliminate outliers.

As described in Chapter 3, visual data contain a lot of outliers. Therefore, we first try to eliminate as many of them as we can during the image processing stage, by increasing the threshold used in tracking the features (see Section 3.). Second, we use regression with a fixed number of iterations so that outliers do not cause spurious over-fitting of the function.

Now, as all steps for combining visual range data with the adaptive logic network have been described, we can proceed to the implementation of the proposed technique. One way of reducing the effect of outliers is to use bagging [Bre96], but we have not yet tried this.

4.8 Experimental Results

A regression-based fusion technique is proposed to fuse range data in situations when other fusion techniques cannot be applied for various reasons. This includes a situation when range data are obtained by the same sensor and are therefore highly correlated, and also a situation when range data are used to build large-scale environments, maintaining 3D occupancy models of which is problematic because of the amount of memory and calculation required by conventional range data fusion techniques.

The experiments of this chapter are aimed to show, first, the validity and the promise of applying the proposed regression-based fusion technique in the situations described above, and second, to expose the problems occurring when applying this new fusion technique.

Let us answer the main questions concerning the experiments.

4.8.1 Questions About Experiments

What we are going to do?

These experiments treat a specific problem of vision-based world exploration which is tackled using a linear regression tool. Therefore, in the experiments we deal with the following: first, a specific type of regression — the least-squares regression which is the basis of the ALN; second, a specific type of range data — visual data sensed from the same location; and third, a specific type of environment — large, full of visual features.

What can we change?

Within the setup of the experiment described above, we can vary the following. We can change data: a) their amount, b) their distribution, e.g. the complexity of the environment, and c) the number of outliers. We can use different numbers of sample points generated per feature in training. We can tune the parameters of ALN learning, such as "jitter" control, fan-in and fan-out factors etc [AT96b]. Whereas it is the influence of various range data on regression-based fusion that is of prime importance for our research, it is also desirable to know how other issues affect the performance of regression.

How are we going to judge the quality of fusion?

Because of the plausibilistic nature of 3D occupancy models, their visualization is difficult, which was first noted by Moravec [Mor96]. Because of this, we examine the quality of fusion, from the point of view of a) its efficiency in computation and space consumption, and b) the applicability of the constructed models.

The former is measured in terms of linear pieces involved in constructing the occupancy function and can be examined for both simulated data and actual visual data. The latter however depends on how the models are used. This way of examining the quality of fusion is applied to actual visual data which are used for a specific task.

4.8.2 Simulations

The purpose of the simulations is to analyze how a regression-based fusion technique reconstructs different commonly used environments. There are two main questions needing to be investigated. First, how can a radially-based coordinate system, which is good for a visual sensor and which we use in regression, represent planar objects, which are the most common objects in man-made environments? Second, how can regression reconstruct parts of the world that are not covered by features? In order to answer these questions, we build models of geometrically simple environments which are easy to visualize.

Empty environments

The first environments to simulate are empty rooms with vertical walls and no objects inside. These environments are shown Figures 4.5 – 4.6. They are a cylindrical room observed from its center (Figures 4.5.a and 4.5.b) and from an arbitrary point (Figures 4.5.e and 4.5.d); a rectangular room, where a sensor is surrounded by four walls (Figure 4.6.a); and a corridor, in which two walls are close to each other and one wall is far from the sensor or may not be seen (Figure 4.6.b).

Environments are approximately five by five in size. They are simulated as if they were observed by a single-camera range sensor described in the previous section located at the origin. The walls are two meters high and are assumed to be full of features. Features are picked randomly with evidence $m = 0.9$. The number of features varies from 500 to 50000. The floor is considered featureless.

Cylindrical rooms are first simulated as if they were observed without outliers and then as if they were observed with 5% of outliers, in which case features appear twice as close to the camera than they actually are (see Figures 4.5.b and 4.5.d). The sensor model is incorporated using 5 sample points per feature according to the visual sensor model as described in Section 4.5.2, which results in 2500–10000 sample points used in regression. The allowable error of fitting is set to 0.15.

The even rows of Figures 4.5 – 4.6 show the results of combining the occupancy data shown in the odd rows of the figures. 3D occupancy models of the environments constructed by ALNs are shown as projections on the floor. The values of occupancy are shown using the intensity of the pixel: the darker the point, the closer its occupancy value to unity. The circular appearance of the data is due to uniform sampling in the coordinate system we use.

It takes only one iteration and one linear surface to model a cylindrical room centered at the origin for the given allowable fitting error (see Figure 4.5.c). However, as the number of outliers increases, it takes more iterations to model the room and more linear surfaces used in modeling. For example, the model shown in Figure 4.5.d was built with 7 linear surfaces within three iterations. The model shown in Figure 4.5.h was built with 57 linear surfaces, whereas the same environment without outliers (shown in Figure 4.5.g) took 16 linear surfaces to model. A rectangular room and a corridor shown in Figures 4.6.d and 4.6.e took 45 and 6 pieces to approximate, respectively.

At this point it should be emphasized that the constructed world models are occupancy models and they should not be confused with geometry-based models, which are very common in 3D reconstruction and which, in many cases, are also built by fitting piece-wise linear surfaces. As seen in the figure, the constructed occupancy model does not follow exactly the contour of the range data in the depth map. However, while using only few parameters, it is able to show clearly the areas of high evidence of occupancy as well as the areas of insufficient occupancy information.

Environments with an object inside

The more complex an environment, the more linear pieces required to approximate it with a given allowable error. Figure 4.7 shows the results of building occupancy models of environments which contain objects. In Figures 4.7.a and 4.7.b a small box, half a meter high, is observed from the center of a cylindrical room. In Figures 4.7.e and 4.7.d a small box is put in an rectangular room. The allowable precision of fitting is set to 0.15 for all environments.

The results of combining these occupancy data are shown in the even rows of the figure. For the given precision of fitting, it took 59 and 65 linear pieces to build the occupancy models of the cylindrical rooms shown in Figures 4.7.a and 4.7.b, respectively, and 151 and 252 linear pieces to build the occupancy models of the rectangular rooms shown in Figures 4.7.e and 4.7.f, respectively.

The problem in fitting the planar objects with radial-based function can be seen.

4.8.3 Real Data

When dealing with range data obtained from real environments, which are very diverse and complex, the tradeoff between precision of the model and computational efficiency becomes an issue of concern. The ALN-based approach however allows us to control this tradeoff by setting the number of linear pieces L used to model the world. If we are interested in a fast but crude approximation of the world, then we choose L small by setting the allowable error of fitting high. The need for approximate and fast-to-calculate models of the world arises in mobile robotics, where a robot has to navigate in an unknown environment in real time.

In our experiments we used a single camera visual sensor described in Chapter 3. The acquired visual range data are used by the robot Boticelli. In our application, a robot explores a room by building models of the observed environment. The constructed occupancy models are used as an input to the thinker module of the robot which decides where to go. The decision where to navigate is determined by the knowledge of obstacle and exploration points. This knowledge is extracted from the occupancy models the robot builds.

The way this knowledge is extracted from the regression-built occupancy models is described in the next chapter. Here we present the results of regression-based occupancy modeling from real data in order to analyze them from the efficiency point of view and also from the point of view of potential problems which may arise.

The environment used for modeling is the same as the one described in the previous chapter (see Section 3.8 and Figure 3.1.a). It consists of an approximately 4 by 5 by 2 meter room, the walls of which are full of visual features. An example of the range data acquired by the robot which observes the room is given in Figure 4.8.a. The presence of features in all directions can be noted.

Figures 4.8.b and 4.8.c show the occupancy models which were built from these range data using the regression based technique with eight ($L = 8$) and sixteen ($L = 16$) linear pieces. These 3D models are used in navigation planning. For this application, the quality of the constructed models was found to be sufficient. In general, we found that for the considered environment and task, using up to 32 linear pieces was enough. This means that a 4 by 5 by 2 meter room was represented with 32 linear equations only, i.e. with 128 real values and 31 boolean conditions only. In contrast, a grid model would require several thousands of stored values for that, even if grid resolution is as low as one decimeter.

The time needed for the robot to build an occupancy model of this quality was approximately one minute on a Pentium Pro 200 MHz computer, which enables the robot to navigate in an unknown environment in almost real time while constantly maintaining a 3D occupancy model of this environment.

4.9 Analysis

In our research, we strive to avoid using ad-hoc parameters and techniques in promoting our new approaches, which is extremely difficult, due to the nature of the tasks involved. Instead, we try to obtain results, which, while based on a particular set of parameters, will serve as a "basis" in the "space of possible solutions", and will provide insights into the nature of the problems considered, thereby setting the ground for further investigation of these problems.

Thinking this way, one realizes that it does not matter what shapes of environments are used in the experiment, how many and what type of simulated outliers are generated or what the particular number of generated sample points is. What is important is that all those experimental examples should serve not as a proof of an advantage of the approach but rather be an illustration of the problems and the advantages of the approach. They should be like bricks from which anybody later can build a house. The analysis of the results below is done with this in mind.

4.9.1 Nature of The Problems

Choice of simulated environments

There is no need to simulate many environments in our experiments, for three reasons. First, in a radially-based coordinate system modeling is isotropic, i.e. orientation of the simulated rooms does not matter. Second, the scale and translation parameters of the environment do not matter either, as their changes result in numerical changes in the solution but not in semantic changes. Finally, the complexity of the simulated environment is also believed not to matter much in studying the approaches proposed, for it would change the size and the complexity of the solution but not its nature.

LS and outliers

There is no need to have many and different types of outliers to show the bottleneck of using the LS regression technique. We consider only 5% of outliers. The results obtained for higher percentage of outliers can be extrapolated from the results obtained. In particular, having 10% of outliers in the experiments of Figures 4.5.b and 4.5.f would require to increase the allowable fitting error from 0.15 to 0.2 and to 0.25 when dealing more complex environments.

Discontinuity of the models

Parametric modeling encompasses a problem which is clearly seen in the simulation results presented in Figure 4.6. Since the model is represented using radially-based coordinates, there is a discontinuity of the model at the pan angle of 2π. In all of the presented figures, pan angle is measured from the bottom side of the Oy axis, where discontinuity can be seen. This problem can be partially resolved however by generating extra "copies" of all sample points beyond the $[0, 2\pi]$ interval. This is what we do in our experiments. This does not eliminate discontinuity completely, it makes it only less significant. This problem however does not prevent one from using the parametrically built models.

Lack of features

The lack of features in a part of the environment may result in undesirably high occupancy values of that part. Moreover, this problem is aggravated by the discontinuity problem mentioned above. Figure 4.9 illustrates this problem. Figure 4.9.b shows the results of fusing of range data corresponding to a corner of the room (shown Figure 4.9.a), where only two walls are observed by the video-camera, and one quarter of the environment is featureless.

However, this problem can be handled by generating a few samples points all over the space with occupancy values equal to zero. Figure 4.9.c shows the result of fusion with 1000 extra sample points generated randomly all over the space to pull the occupancy function down to zero.

Visualization problem

As mentioned by Moravec [Mor96], visualization of 3D occupancy data is a problem in itself, requiring extra investigation from researchers in the Computer Graphics community. This makes judging the quality of the occupancy models by visual resemblance to the real world very difficult. In our case however, the visualization problem is made even worse by not using grids in representing the models. In particular, understanding the results plotted in Figures 4.5 – 4.9 is obscured by the quasi-cylindrical system of coordinates we use.

Therefore, the best way to analyze the results is from the point of view of their utility to a particular application. This can be done with real data only.

Other

At the present moment we do not see any other semantic problems of the regression based range data fusion approach.

4.10 Conclusion

We developed a regression-based technique for fusing uncertain range data. This technique was shown to have several semantical advantages over conventional rule-based range data fusion techniques. First, it does not assume range data to be independent and is therefore more suitable for combining highly correlated range data like those obtained by the same range sensor.

Second, as opposed to other techniques, the regression-based technique automatically finds the number of parameters needed to model an environment. These parameters are the parameters of linear equations which suit best a given environment. This is beneficial for modeling large-scale environments, and is also useful for rapidly building crude models of an environment.

Finally, the regression-based technique provides the basis for building parametrically represented occupancy models, which as will be shown in the next chapter are seen as a solution to the "curse of dimensionality problem" which occurs when building 3D occupancy models of large scale environments.

Thus, we can conclude that the proposed regression-based range data fusion technique is very promising for constructing 3D occupancy models of unknown large-scale environments. Further development of the technique is required however.

This dissertation provides the basis for the new framework in range data fusion, and now this framework can be used and further improved. We see main directions for the development of the framework in, first, studying other neural network and regression techniques for fitting the occupancy data, and second, investigating the possibility of combining several occupancy models. The concluding chapter gives more details on our vision for further research in the area.

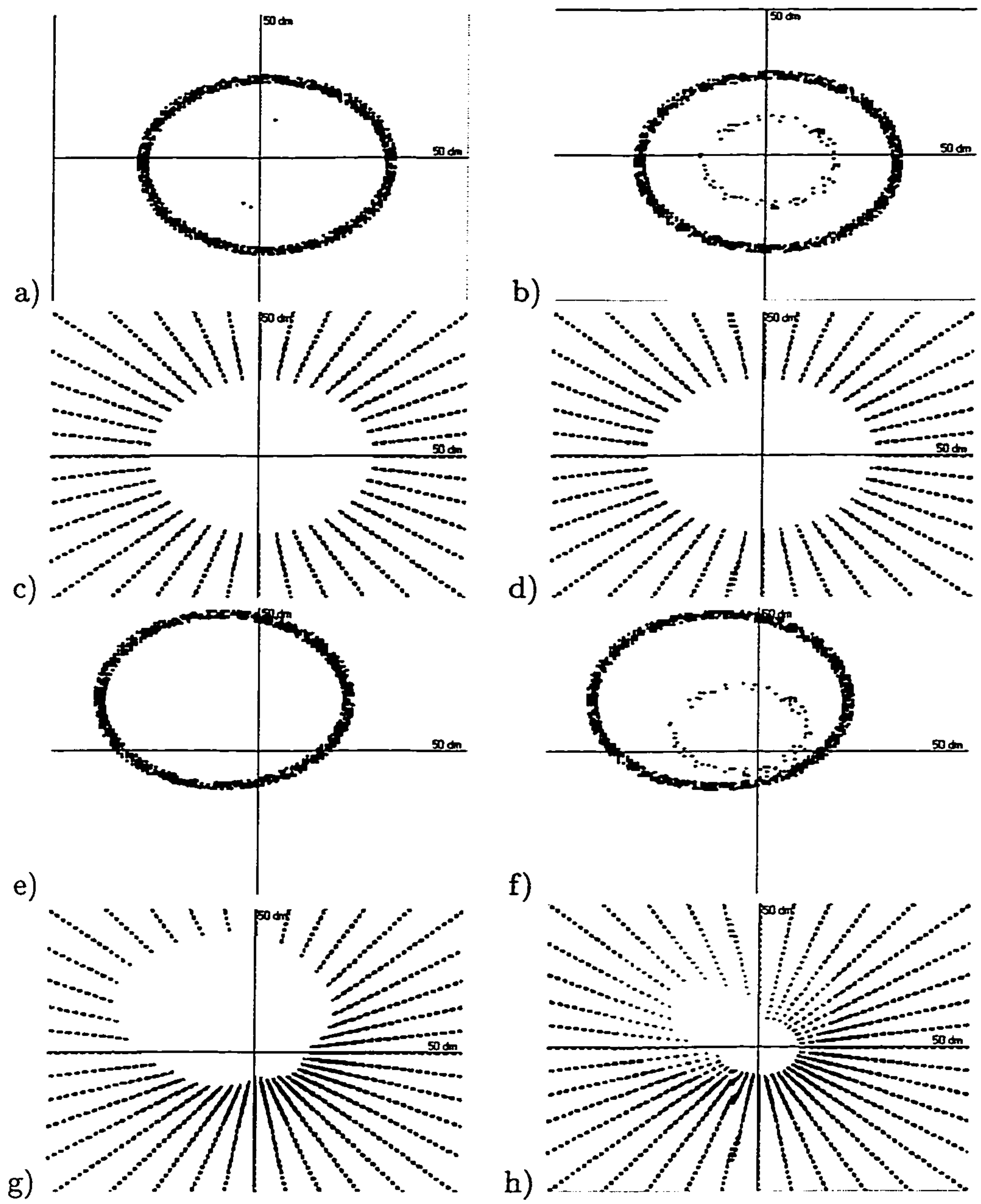


Figure 4.5: *A simulated cylindrical room observed from its center (first two rows) and from an arbitrary position (last two rows), with 5% of outliers (right column) and with no outliers (left column): simulated range data (odd rows) and the constructed occupancy models built (even rows) are shown.*

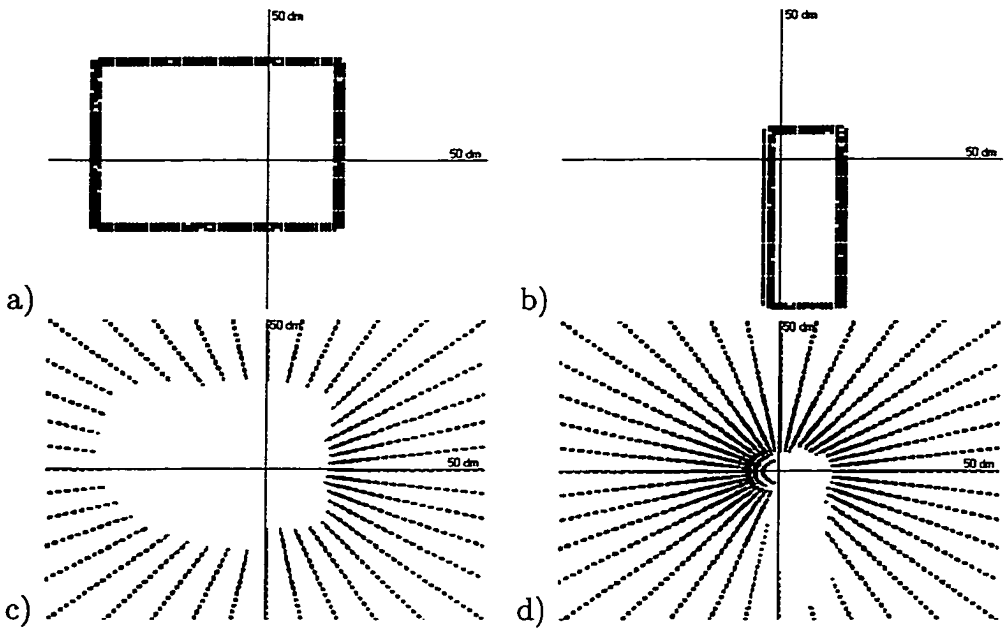


Figure 4.6: *Rectangular room and corridor: simulated range data (first row) and the constructed occupancy models (second row).*

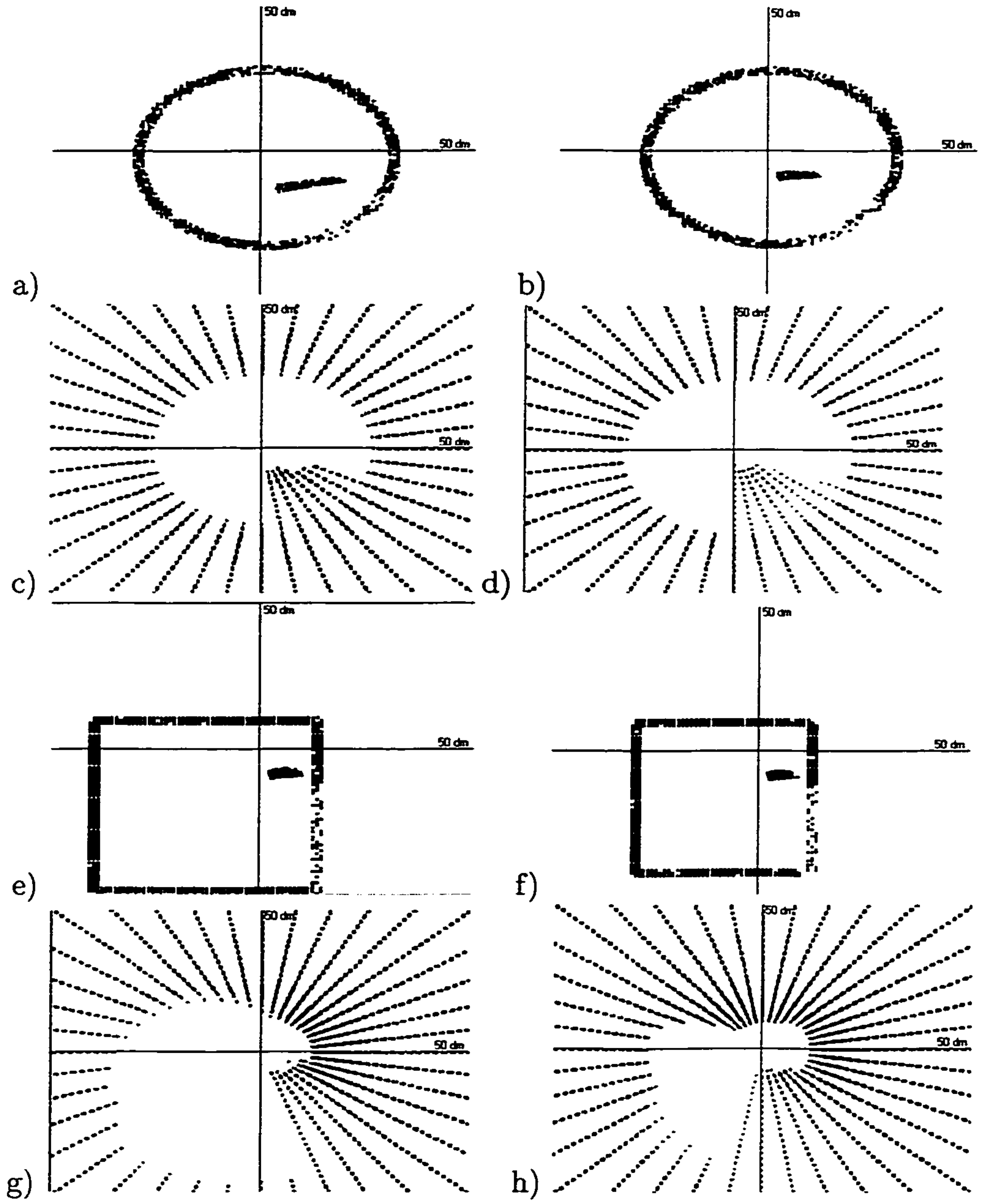


Figure 4.7: *Environments with an object inside: simulated range data (odd rows) and the constructed occupancy models (even rows).*

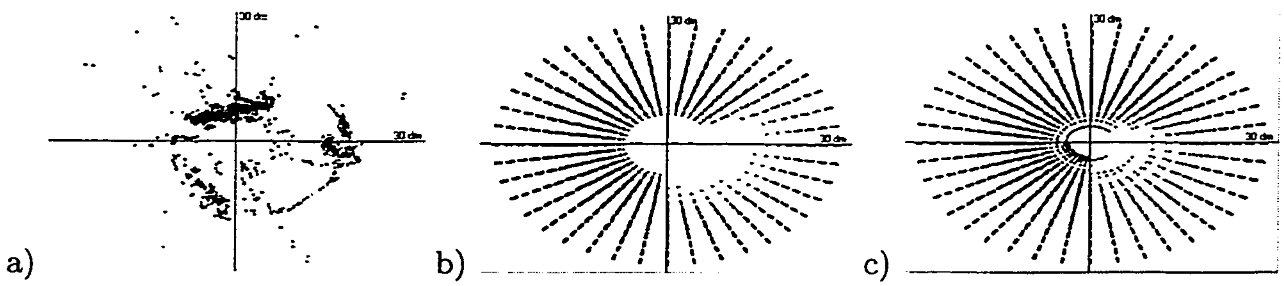


Figure 4.8: *The occupancy models constructed for real range data.*

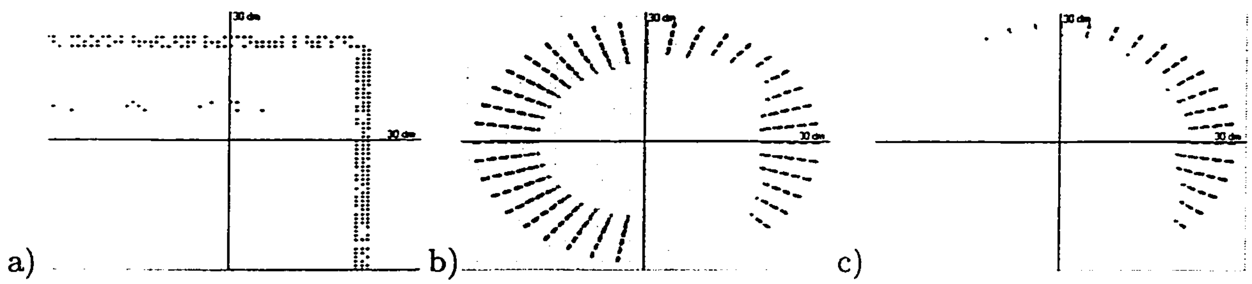


Figure 4.9: *Dealing with the lack of features.*

Chapter 5

Using Piecewise Linear Representation for World Exploration

5.1 Introduction

In the previous two sections we introduced a technique for building piece-wise linear models of visual sensors and a regression-based technique for fusing range data. While resolving the visual sensor design problem and the problem of combining dependent range data, these techniques also provide the basis for resolving another major problem of the occupancy approach — the *representation* problem.

The representation problem deals with the redundancy and inconvenience of the grid representation of the occupancy models for world exploration. It is the one that impedes most the development of the occupancy approach. In particular, it is because of this problem that the 3D occupancy approach was applied so far for modeling of small objects only. This is also why up till now in mobile robotics, where environments are big, only 2D occupancy models are used to model the environment.

In this chapter a new approach for representing the occupancy function using piecewise linear surfaces is proposed. First, we show the disadvantages of the grid representation of the occupancy function and demonstrate the idea of the new approach on a simple example. Then, we show how the occupancy model of the world can be efficiently represented using a min-max tree of combining linear functions.

Then we show how parametrically represented occupancy models can be used for the world exploration problem, which is the problem the occupancy approach is commonly used for. In particular, we describe an approach which uses the information extracted from the constructed occupancy models in order to make navigation decisions. This approach uses a reinforcement learning technique where the reinforcements are obtained from the knowledge of the goal location as well as from the knowledge of unexplored area and the likelihood of obstacles in the exploration area, which is extracted from the occupancy models. We demonstrate the validity and the promise of our approach by implementing it on the mobile robot Boticelli, which searches for objects in an unknown environment using a single camera stereo range sensor.

5.2 Problems With The Grid Representation

Let us illustrate the problem with the grid representation on a concrete example. We use an example from [Mor96] slightly modified to the robot and environment we use in our experiments.

Consider a 10 meter by 10 meter by 3 meter hall where a robot equipped with a camera has to navigate (see Figure 5.1). In order to decide where to navigate the robot has to build a model of this hall.

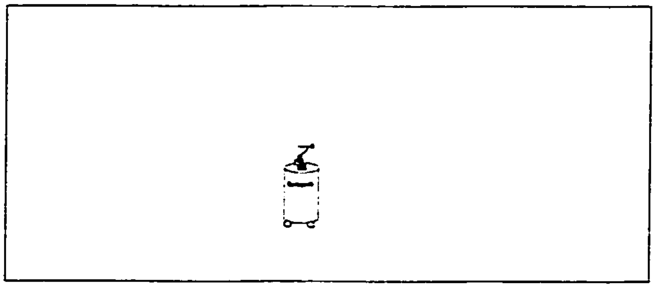


Figure 5.1: *Robot in a hall.*

5.2.1 Memory Requirements

First, since the environment is considered to be unknown, we do not know what the resolution of the grid should be.

A common approach in this case would be to choose the resolution of the grid according to the resolution of the sensor [Mor96]. For vision based sensors, the resolution of one decimeter can be used [MS87, GA99b]. This results in storing a 3D array with 300,000 real values. That is, the grid approach requires megabytes of memory to the given task.

5.2.2 Amount of Calculations

For each of registered features, ray tracing should be done in order to update the occupancy values of all voxels lying in the ray of view. This results in update of several millions voxels for each of about 2000 evidence rays obtained from one image frame. For this to be executed in real-time, one needs an extra powerful computer with at least 100 MIPS speed [Mor96] .

5.2.3 Occupancy Models for Navigation

In the occupancy approach [MM96, BEFW97, JD97, YSA98, BBC+95, PNDW95, vDKG95], the exploration policy is usually determined by the following information which is extracted from the occupancy model: the observed obstacles, the navigation area, which is the area free of obstacles, and the unexplored area, which is area where insufficient range data has been acquired.

When this information is obtained, it is processed in order to produce the command for the robot. Such methods as potential fields [JD97], value iteration [Thr95] and other reinforcement learning techniques [SB98, ACG99] are most common at this stage.

As understood, no matter what a technique is used at later stages, the success of the occupancy-based world exploration depends on the quality of the occupancy model and also its suitability for extraction of the information required for navigation.

As demonstrated above, because of the grid representation, which results in storing a huge amount of data and performing very time-consuming calculations, modeling of 3D, and large-scale environments in real time is practically impossible. Because of the this, in mobile robotics, where the issue of time is critical, only 2D occupancy models have been used. That is, instead of treating a world the way it is, a robot has to consider only a 2D shadow of it in making a navigation decision. And this is not the only problem encountered using the grids.

Grid models are not suitable for radial range data and they are very inefficient for map extraction. This can illustrated by the fact that in order to get the boundary of the area available for navigation, the robot has to ray-trace a 3D array of data and then to use line extraction algorithms to obtain a 2D boundary. This consumes a lot of computational power.

5.2.4 Concluding the Example

Let us return to the example of a robot navigating in a hall which we alluded to in the beginning of the section. We know now how much memory and computational power it requires for the robot to build the model of a hall. Now assume that the hall is completely unfurnished and empty.

It is quite striking to realize that all computational effort was spent to built a model of an empty space! This is when one realizes that there should be another way of building and representing the occupancy function. Unfortunately, however there seem to be no other approaches known in occupancy world modeling except grid-based ones, and this work seems to be the first to suggest a way of building occupancy models without using grids.

5.3 Parametric Representation

5.3.1 Empty Room Example

Let us recall that the occupancy model of a room is defined by an *occupancy function* which maps 3D points of the world into a real interval so that higher values of the function indicate points that are more likely to be occupied:

$$m = F(\vec{r}), \quad m \in [0, 1], \quad \vec{r} \in \Re^3. \tag{5.1}$$

It is obvious that in certain cases, for certain room shapes and coordinate systems, the occupancy function of the room can be represented very simply using equations. For example, the occupancy function of a cylindrical room with radius $R = 5$ meters, which is observed by a robot from the center of the room (see Figure 5.2.a), can be expressed as simply as

$$m = MIN\left(\frac{r^n}{R^n}, 1\right), \qquad \text{where } r = \sqrt{x^2 + y^2} \tag{5.2}$$

Here the sensor model along the ray of view is approximated as $m = (\frac{r}{R})^n$ and n is determined by the resolution of the sensor.

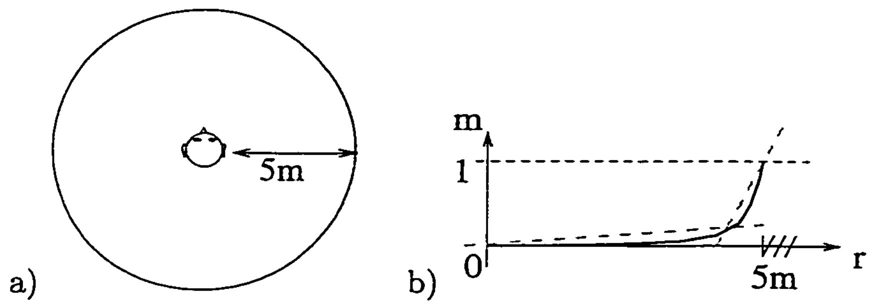


Figure 5.2: *A robot in a cylindrical room (a) and an approximation of the occupancy function using linear equations (b).*

5.3.2 Linear Representation

It is clear that any continuous function can be approximated using piece-wise linear surfaces. For example, using the cylindrical system of coordinates and piece-wise linear representation, the occupancy function of Eq. 5.2 can be represented as a system of linear equations (see Figure 5.2.b):

$$m = \begin{cases} a_1 r, & \text{if } r < r_1 \\ a_2 r - b_2, & \text{if } r \geq r_1 \end{cases} \tag{5.3}$$

For a more complicated environment, like the one illustrated at Figure 5.3, for instance, where there is an object between a wall and the robot, the occupancy function would look like the following

$$m = \begin{cases} a_1 r + b_1 \alpha + c_1 h + d_1 & \text{if } h < 1,\ 0.00 < \alpha < 0.10,\ r < 4.5 \\ a_2 r + b_2 \alpha + c_2 h + d_2 & \text{if } h < 1,\ 0.00 < \alpha < 0.10,\ r \geq 4.5 \\ a_3 r + b_3 \alpha + c_3 h + d_3 & \text{if } h < 1,\ 0.10 < \alpha < 3.14,\ r < 6 \\ \ldots & \end{cases} \tag{5.4}$$

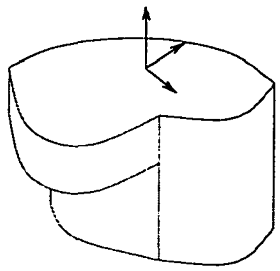


Figure 5.3: *Illustration of piece-wise linear representation of the occupancy model.*

Thus, it is clear that the occupancy function of an arbitrary environment can be represented as a system of linear equations, where the number of equations is determined by the complexity of the environment. The simpler an environment is, the less memory and calculation are required to build the model of the environment. This is exactly what we strive to achieve.

The two main questions which arise regarding building parametric models are the following.

1. How should surface equations be calculated using a set of registered range data? and

2. How can these equations be converted to 2D navigation maps?

The solution to the first problem has been already presented in the previous chapter. It is the regression-based range data fusion technique, which computes the occupancy function by fitting range data with piece-wise linear surfaces. The occupancy functions which are constructed using this technique are represented using the Adaptive Logic Network (ALN) as a binary logic tree over linear functions.

The solution to the second problem is presented in this chapter. It is the technique which, given a parametrically represented occupancy function, extracts the information needed for making the navigation decision and then uses this information to decide where to navigate.

Before presenting the technique for map extraction from the regression-built occupancy functions, we first explaine the ALN representation of a function.

5.3.3 ALN Representation

Figure 5.3.3 shows the contents of a file created by the ALN learning program. It contains linear equations and MAX,MIN relationships of linear pieces used in fitting sample data. This set of equations and minimum (MIN) and maximum (MAX) operators over the equations comprise the ALN representation of the function.

The function presented in the figure is a 2D occupancy function built by the ALN for the 2D environment which consists of two walls. The same data as those shown in Chapter 2 in Figure 2.1 are used. The system of coordinate is Cartesian. Linear equations are written in the form

$$a_0(x_0 + c_0) + a_1(x_1 + c_1) + a_2(x_2 + c_2) = 0, \tag{5.5}$$

where x_2 is the output value and x_0 and x_1 are the input values, where $(_0, c_1, c_3)$ is roughly the centroid of the piece.

In the case of 3D occupancy modeling of more complex environments, there are many linear pieces involved in building the model, and, in general, understanding the relationships among those linear pieces is very diffucult, which is due to both the extra (4th) dimension and a different coordinate system used in modeling. However, the idea is still the same.

```
------------------------------------------------------------------------------------------
VARIABLES = 3;
x0 : [0, 100];
x1 : [0, 100];
x2 : [-0.10000000000000001, 1.1000000000000001];
OUTPUT = x2;

LINEARFORMS = 8;
0 : -0.0025775417471730249 (x0 - 68.406933942514058) +
  0.00097339244428427496 (x1 - 5.2084780444034102) - 1 (x2 - 0.0091314141451077861);
1 : -3.473863743550066e-013 (x0 - 47.251255792582434) +
  4.1563576567829662e-013 (x1 - 13.160881035134665) - 1 (x2 - 4.9063554368901343e-012);
2 : 0.00016062073088807331 (x0 - 39.786671388398872) +
  0.27508702928044448 (x1 - 28.655593772023508) - 1 (x2 - 0.62594933920327589);
3 : -5.4931402340175244e-019 (x0 - 44.432716144393254) -
  1.3666804381996871e-017 (x1 - 38.450723277018646) - 1 (x2 - 0.99999999999999978);
4 : -1.6221746004891314e-031 (x0 - 14.634643492882514) -
  3.359150903932276e-032 (x1 - 40.880839341599732) - 1 (x2 - 7.9796233328002509e-032);
5 : 7.2701086463233145e-064 (x0 - 6.7833882297211661) +
  1.9110809581926726e-063 (x1 - 24.723279572639708) - 1 (x2 - 3.1360567225957893e-062);
6 : 0.37157500203172533 (x0 - 17.657355731337013) +
  0.0011219898249259102 (x1 - 34.622157550187197) - 1 (x2 - 0.62070213711450717);
7 : 0.3752589541124618 (x0 - 17.546701092500417) +
  0.00012452206178189317 (x1 - 41.886512024589692) - 1 (x2 - 0.57993755330086338);

BLOCKS = 1;
0 : MIN(MAX(MIN(0, 1), MIN(2, 3)), MAX(MIN(4, 5), MIN(6, 7)));
------------------------------------------------------------------------------------------
```

Figure 5.4: *The ALN representation of a 2D occupancy function.*

5.4 Extracting Maps

Once a 3D occupancy model of the world is constructed as a tree of minima and maxima of linear functions of three variables as described in Chapter 4, that is function

$$m = F(\alpha, h, r) = tree^{(MIN,MAX)}_{l=1...L}\{a_l\alpha + b_l h + c_l r + d_l\}$$

is calculated, it is possible to determine a 2D polygon within which it is safe for the robot to navigate.

5.4.1 Occupancy Function Inversion

The volume with occupancy less than a certain threshold (e.g. 0.6), is considered unoccupied and therefore available for navigation. In order to find this volume, the first step is to invert the occupancy function. The inverse function returns a distance within which it is safe to move as a function of pan angle, height and occupancy. This can be done theoretically due to the strong monotonicity condition imposed on the occupancy function (Eq. 4.24) during ALN training.

The ALN max-min tree representation of the occupancy function allows the inversion to be done very efficiently. The inverted ALN is constructed as follows: in the original ALN tree, each maximum node is replaced by a minimum node, and each minimum node by a maximum. Then the weights on variables are normalized in such a way that the weight on the new output variable (occupancy) is –1. The simplicity and speed of this inversion is another advantage of using ALNs for robot navigation.

5.4.2 Computing Navigation Polygons

The inverse occupancy function, which can be now written as

$$r = F^{-1}(\alpha, h, m), \tag{5.6}$$

where the occupancy is fixed at some level, say $m = 0.6$, is now applied at several values of height h. The r values are converted to a horizontal distance by taking $\sqrt{r^2 - h^2}$. The result of this computation is a set of polygons obtained at different heights centered at the robot's current position. In addition to the monotonicity constraint imposed in the regression (Eq. 4.24), we also impose upper and lower bounds on the weights for pan angle and height (Eq. 4.25). This allows us to use a finite set of height values and yet be sure that no point on the robot at any height will collide with any point of the environment exceeding a certain value of occupancy.

The final step in calculation of the polygon of the 2D local map of the area available for navigation consists in shrinking all polygons by the radius of the cylindrical robot and taking the intersection of their areas. This ensures that the whole body of the robot can go in a straight line to any point inside the intersection polygon.

5.4.3 Experimental Results

Figures 5.5 show examples of polygons extracted from the occupancy models using the described techniques. The polygons shown in the figure are extracted using a threshold of $m = 0.6$ on the maximum occupancy considered safe for navigation.

5.4.4 Obstacle and Exploration Points

The polygon boundary far from the robot is most susceptible to error. This could result from absence of depth data or errors in depth estimation. This led us to upper-bound the distance to the periphery of the polygon from the robot position. The points of the navigation polygon which lie inside the bounding circle represent obstacles. Obstacle points are defined for evenly spaced angles in angular sectors where obstacles occur. When making a decision where to navigate, these points will be given negative reinforcement values to keep the robot from hitting the obstacles.

Points of the navigation polygon on the periphery represent points where the knowledge of the environment becomes undependable, so further data must be collected near them. A collection of exploration points is defined at evenly spaced angles in sectors where the navigation polygon lies on the circle. The density of exploration points is chosen to be adequate to find channels through which the robot could pass, but which may not be observable from the current robot position. Exploration points have positive reinforcement values, thus encouraging the robot to move near to exploration points.

5.5 Planning Navigation

5.5.1 Reinforcement Learning in Planning

Reinforcement learning occurs when a system learns, from environmental feedback, to what degree its past actions have been satisfactory. It uses this information to determine future actions. A book [SB98] provides an introduction to reinforcement

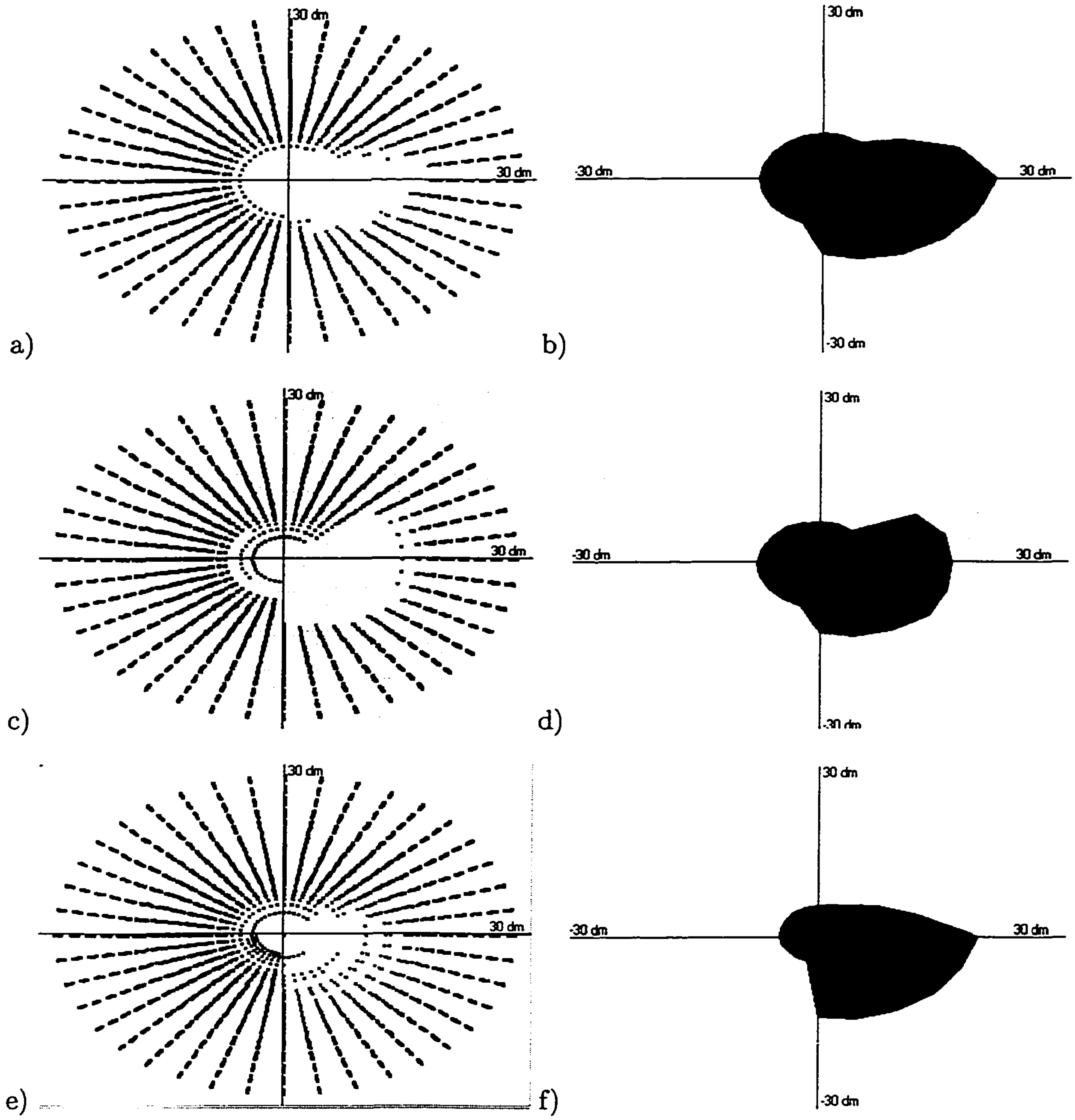


Figure 5.5: *Occupancy models (b) and the navigation polygons (c) extracted from these models.*

learning. The model is as follows. If the system is in state s_t at time t, and a certain action a_t is taken, then it enters state s_{t+1} and receives reinforcement r_{t+1}. We shall model the next state and the reinforcement as deterministic functions of the current state and action.

A policy π specifies what action to take in any given state. A policy must be developed taking into account how much reinforcement is received at $t+1$ as a result of an action at time t, and how much will be received for all subsequent actions under the policy. In this way, a policy can determine an action which is likely to increase reinforcement over the sequence of all future actions. The value of the present and all future reinforcements is the *discounted* reinforcement defined as

$$V^{\pi}(s) = \sum_{n=0}^{\infty} \gamma^n r_{t+1+n} \tag{5.7}$$

where actions are taken according to π and $0 < \gamma < 1$. This is referred to as the state-value function or simply the value function for policy π. For the optimal policy V^*, which produces the greatest discounted reinforcement that the system can achieve over all possible future actions starting from state s, Bellman's equation [Bel57] is satisfied:

$$V^*(s) = \max_a (r(s,a) + \gamma V^*(s')) \tag{5.8}$$

where s' is the state resulting from applying action a to state s.

5.5.2 Application to Navigation

The way we have applied reinforcement learning to navigation is similar to the work of others [JD97, Thr95, TB96], where the states of the system are positions of the robot. The reinforcement resulting from an action is determined by whether the new state is close to the goal or close to a point that requires further exploration, or whether it is close to an obstacle. The former situations give rise to positive reinforcements, while being close to an obstacle leads to negative reinforcement. Solving Bellman's equation gives rise to a state-value function that represents a potential field, where the robot is attracted to exploration points and the goal, and repelled from obstacles.

The function $r(s, a)$ in Eqs. 5.7 and 5.8 which gives the local reward is defined using the sum of rewards at the arrival state s' derived from all obstacle points, exploration points and the goal, if already known. Each obstacle point was given a reward function zero everywhere except in the neighborhood of the obstacle point, where the graph was in the shape of a negative-pointing cone centered at the obstacle point.

Various cone radii were tried, but the idea was to give a gentle repulsion to the robot if it came close to an obstacle point $\vec{p}$. Thus the contribution of the obstacle to the reward function is

$$r_{obs,p}(s, a) = min\{0, (V_{obs} + S_{obs}||s' - \vec{p}||)\}.. \tag{5.9}$$

Here and in the following, V_{type} is a fixed value at the apex of the cone, and S_{type} is the "slope", i. e. the rate of change from that value as distance from $\vec{p}$ increases. These two parameters determine the height and the shape of the cone. We recall that s' is a function of r and a.

Each exploration point $\vec{p}$ is given a reward

$$r_{exp,p}(s, a) = max\{0, V_{exp} - S_{exp}||s' - \vec{p}||\}. \tag{5.10}$$

which is an upward-pointing cone, where it is not constant 0.

The goal point was given a similar reward

$$r_{goal,g}(s, a) = max\{0, (V_{goal} - S_{goal}||s' - \vec{g}||)\}. \tag{5.11}$$

Thus $r(s, a)$ is the sum of all the contributions from obstacle points $\vec{p_i}$, exploration points $\vec{q_j}$ and the goal $\vec{g}$ that are currently known to the planner.

With this reward function, the next step was to solve Bellman's equation.

5.5.3 Calculating Value Function Using ALN

An Adaptive Logic Network is used to calculate the value function in Eq. 5.8. The properties of the ALN that make it suitable for approximating the value function are discussed in [KR96]. An ALN learns to approximate V^* by growing a tree of maximum and minimum operators acting upon linear functions. The training algorithm adapts the weights of the linear functions and splits linear pieces into two if the error of a piece is too great, so that the tree grows. At a given input s, generally only one linear piece is responsible for forming the function value. It is found by searching the ALN tree, whereby the branch of the lesser input is chosen if the node is a minimum node and the branch of the greater input is chosen if the node is a maximum node. Training of the ALN is done by adapting the weights of that one responsible linear piece. A refinement of the procedure joins linear pieces by quadratic fillets and splits the responsibility among a few linear pieces. The occupancy function is a mapping from a box in three dimensional space to real values, and the function is linear on pieces of the space which are abutting polyhedra. The state-value function for reinforcement learning has two inputs, namely the coordinates of the robot.

To perform reinforcement learning, the value function being learned by the ALN is used to compute training values for itself: values of the expression on the right of Bellman's equation are used to define the target value for the function at the state on the left. The control policy after training is complete is

$$\pi^*(s) = arg\max_a(r(s,a) + \gamma V^*(s')) \tag{5.12}$$

where s' results from action a.

A significant speedup of computation is achieved using a piecewise linear value function approximant because many of the linear pieces that do not influence the ALN computation for a given input point do not have to be evaluated. (Finding out the exact set of pieces that have to be evaluated is a computationally intractable problem; but a good heuristic solution is achievable.) This makes the ALN approach to reinforcement learning a promising candidate for real-time applications.

To demonstrate that ALNs can learn complicated functions, the Q-function for basketball balancing was learned [Arm98, AL98]. In this task, a basketball must be balanced on a "finger" that moves in a horizontal plane only.

5.5.4 Experimental results

In the experiments, states s are randomly chosen in the area opened to navigation. The robot is given only a fixed number of actions to apply in each state s. In the simulation, there were four possible directions of motion. For the real robot, eight possible directions were sampled at a constant distance, although it would have been better to do more sampling to get a better value of the maximum. The value function $V(s)$ currently represented by an ALN was trained using the estimates $V(s')$ at the states s' accessible through this finite number of applicable actions at s. This gave rise to a function approximating the value of each state under an optimal policy.

Figure 5.6 shows the value function $V(s)$ constructed by the ALN at two different locations of the robot. After reinforcement learning, the robot can be sent an appropriate direction to move so as to obtain maximal gradient of the Value function.

5.6 Conclusion

As can be seen from the way the occupancy models are used, high precision of these models is not required. What is required is obtaining fast a rough idea of what is around the robot. If there is a need for more precise information about a particular part of the environment, then other exploration techniques can be employed, including, for example, conventional grid-based occupancy techniques to model small parts of an environment or telepresence techniques [BBZ98], which use the help of a teleoperator in order to guide the robot using the information from the video-camera.

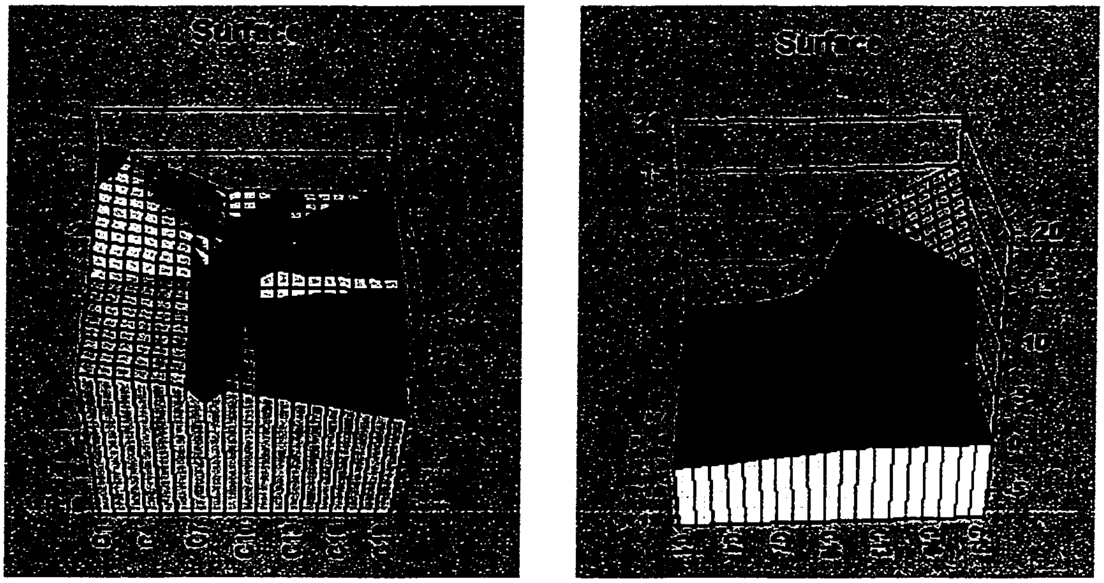


Figure 5.6: *The value functions built by the reinforcement learning technique at two different locations of the robot.*

Chapter 6

The Framework and Robot Boticelli: Implementation of the Ideas

6.1 Introduction

This chapter puts together all techniques presented in the previous chapters into a one framework — a framework for building occupancy models using unreliable visual range data and parametrically represented occupancy functions. The first half of the chapter presents all stages which lead from the first step of grabbing a video-frame to the final step of making a navigation decision based on the constructed models. The stages are described in a sequence to be used in implementation of the framework.

The implementation of the framework is presented in the second half of the chapter. We describe the software and hardware architectures of robot Boticelli which was designed for the purpose of demonstrating the approaches proposed in the thesis. The data obtained from running the robot are presented.

6.2 Framework for Vision-based World Modeling

Let us recall what task we want a robot to accomplish. This task is to

$$\boxed{\textit{``Explore an environment with a videocamera''}}$$

How much is hidden in this simply-stated six-word problem description! How many stages are there in between the very first stage of grabbing a 640 by 480 image and the very last stage of making a move in a certain direction!

This dissertation attempts to cover all of these stages: from Image Processing to Navigation Planning, by studying the limitations of the current approaches and proposing new techniques. We can now summarize all these stages and present the entire framework for vision-based occupancy modeling which uses techniques proposed in the dissertation for a) registering, b) combining, c) representing and d) applying the occupancy values.

Below we summarize all steps on the way from grabbing an image to making a navigation decision. First, we describe each step stressing the importance of doing it and then we illustrate each step using an example from an experiment.

6.2.1 From Image Processing to Navigation Planning

The stages of the vision-based world exploration using occupancy models as proposed in the dissertation are the following.

Image Processing and Computer Vision

1. Decrease the resolution of the grabbed image. There is no need for high resolution, as we deal with uncertain data. The smaller the image, the faster range data acquisition.

2. Select visual 2D features. The number of selected features should be large enough (enough for building models of a desired precision), but not very large (to make registration fast). The features should be selected in such a way that it will be easy to track them. The higher the derivative of the feature intensity in the direction of camera motion, the more robust the feature in tracking.

3. Move camera so that to make baseline as large as possible while keeping features in the field of view and track features along the epipolar line which should be

made wide enough to account for image warping. Calculate feature match error.

4. Calculate the distance to the successfully tracked features. A feature is considered successfully tracked, if its match error is lower than a certain threshold. The lower the threshold, the more robust the feature.

5. Calculate levels of evidence m_i of registered 3D features $\vec{r_i}$ using the feature match error as a measure of uncertainty.

6. Move camera around and register 3D features in all directions, generating a set of range data $\{m_i, \vec{r_i}\}$ to be used in fusion.

Range Data Fusion and Modeling

1. Choose the coordinate system which suits best the sensor model and the environmental constraints, such as the shape of the robot or obstacle geometry if known. Appropriate choices include spherical (α, ψ, r), cylindrical (α, h, r_{xy}) and quasi-cylindrical (α, h, r) coordinate systems.

2. In the chosen coordinate system, impose constraints on the occupancy function according to the sensor model and the task for which the occupancy model will be used. The main one is the monotonicity constraint

$$\frac{\partial F(\alpha, h, r)}{\partial r} > 0.$$

   Other constraints may prevent abrupt changes of occupancy values (e.g. $-C_1 < \frac{\partial F}{\partial \alpha} < C_1$ and $-C_2 < \frac{\partial F}{\partial h} < C_2$ for some constants C_1 and C_2).

3. For each registered 3D point, generate a few additional range data on the ray of view with the plausibility values calculated according to the sensor model. The number of generated points should be as small as possible to make modeling fast, yet it should be large enough to represent the shape of the sensor model.

4. Given range data about the surrounding environment, which is encoded in terms of registered and generated sample points $\{m_k, \vec{r}_k\}$, find an occupancy function $m = F(\vec{r})$ which fits these range data with the given precision subject to the imposed constraints. For this purpose use multiple linear regression tools, such as ALNs, which are fast and which yield a piece-wise linear representation for constructed functions. Other regression or neural network techniques such as GMDH [MI94], ALB [Hoo99], MARS [Fri91] or PPR [FS81] can also be applied at this stage.

5. The level of the fitting precision should be set according to a) the technique used in fitting, b) percentage of outliers in the range data and c) time constraints. If, as a result of using non-robust range sensors, the percentage of outliers is high, then the level of the allowable error in fitting should be not very small to avoid spurious overfitting of the function. In the case of ALNs, overfitting can be avoided by either setting in advance the number of linear pieces used in regression or by keeping the allowable error above the level of noise.

Using Occupancy Models

1. As a result of fitting, ALN builds an occupancy function represented as a binary tree of minima and maxima of linear functions:

$$m = tree_{l=1...L}^{(MIN,MAX)}\{a_l\alpha + b_l h + c_l r + d_l\},$$

2. From the constructed 3D occupancy model, extract a 2D navigation polygon within which it is safe to navigate. This is done by inverting the occupancy function and setting the occupancy level considered to be safe for navigation

$$r = F^{-1}(\alpha, h, m).$$

Due to the monotonicity condition imposed on the occupancy function, this inversion is possible, and due the linear representation of the function, it is fast.

3. Obtain the lists of obstacle and exploration points from the extracted 2D navigation polygon. Exploration points are those which lie far from the robot. Another way of extracting exploration points would be a) to build the belief occupancy function in addition to the plausibility occupancy function, and b) to consider the points which have high uncertainty interval defined as a difference between the belief and plausibility values.

4. Assign negative reinforcement values to the obstacle points and positive reinforcement values to the exploration points and to the goal location, if the goal is observed.

5. Use reinforcement learning to calculate the value function. After the function is calculated, move into the direction of the maximal gradient of the calculated value function.

6.2.2 Illustrations of The Ideas

Figure 6.5 illustrates the described framework at work, as applied to a concrete world exploration problem. In this problem, a robot has to find a predefined target in an unknown environment, using a single-camera range sensor (Figure 6.5.II). The target is a green triangle glued to a white paper background as seen in Figure 6.5.I. An approximate plan of the room used in the experiments is shown in Figure 6.5.III. Figure 6.5.III also shows approximately the moves of the robot on its way to the goal. The moves of the robot are guided by the framework described in the previous section.

Figure 6.5.1 shows a 640x480 image of monochrome green rectangles as observed by a camera. The first reason for the uncertainty in data – imperfection of the camera – is clearly seen.

Figure 6.5.2 shows a preprocessed 160x120 image to be used for depth calculation. Selected 2D features are shown in white. There are about 500 of them per image.

The second reason for the uncertainty — the complexity of the environment and the undistinctiveness of visual features, – is also clearly seen.

Figure 6.5.3 illustrates the depth calculation technique based on projection geometry, which is applied to the selected 2D features which have been successfully tracked.

Figure 6.5.4 shows how uncertainty of feature registration is incorporated into the sensor model. Tuckey byweight [RL87] is applied to the feature match error to obtain the value of evidence. Depth error due to the limited resolution is approximated as 10% of the distance. The range data collected by observing the environment are shown in Figure 6.5.5. The search for the goal is done concurrently with registering 3D features.

Figure 6.5.6 shows the occupancy model built from these range data. The quasi-cylindrical system of coordinates used in regression is shown in Figure 6.5.7. The number of generated sample points per feature is five, as shown in Figure 6.5.8. The number of iterations used in the ALN training is ten. This results in fast calculations (less than two minutes) and less than fifty linear pieces needed to represent the environment. The idea behind ALN fitting is illustrated in Figure 6.5.9.

Finally, Figure 6.5.10 shows the 2D navigation map extracted from the 3D occupancy model with the level of safe occupancy set to 0.6. The figure also shows obstacle points (in black) and exploration points (in white) to be used in making the navigation decision. The value function which is calculated for these points is shown in Figure 6.5.11. This value function determines the next move of the robot, until finally the robot reaches the goal.

6.3 Robot Boticelli

Boticelli is a prototype mobile robot, built for the purpose of demonstrating the ideas presented in the thesis: new techniques for vision-based range data registration and new paradigms in range data fusion and world modeling.

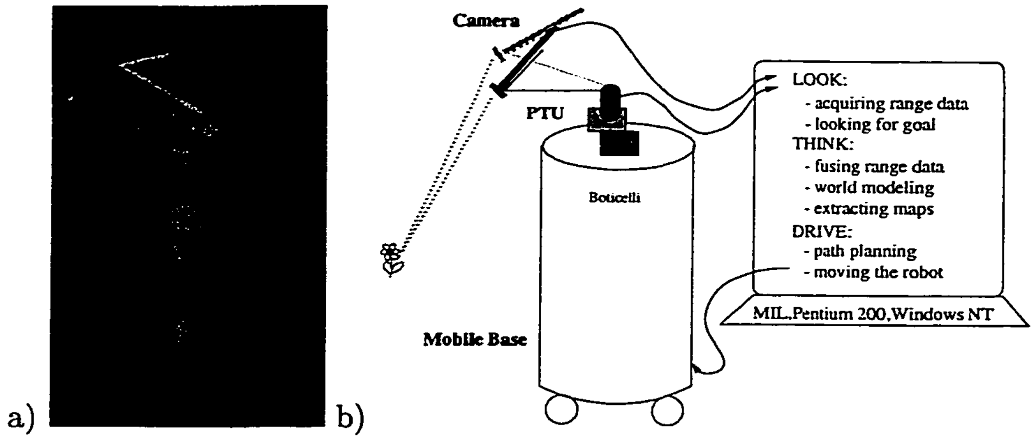


Figure 6.1: *Boticelli (a) and its architecture (b).*

6.3.1 Robot Architecture

Boticelli, shown in Figure 6.1, consists of an RWI B12 mobile base with a video camera mounted on a Direct Perception pan-tilt unit. Boticelli communicates with two external computers by a tether link. One Pentium Pro 200MHz computer is used for image processing and data fusion and a Dual Pentium II 333MHz computer is used for reinforcement learning and navigation planning.

It is designed for navigation in an unknown environment. Navigation decisions are determined by the following information:

1) observed "obstacle points" which are likely to block motion,

2) "exploration points" at which further observation will be required, and

3) the goal location (if known).

Until the goal is reached, the robot is required to continually look for the goal while watching for obstacles, determining where to move and moving. This suggested a software architecture of the robot, which we call the "Look-Think-Drive" architecture (see Figure 6.1). The robot repeats a cycle of looking (vision), thinking (planning), and driving (motion). The architecture consists of three main modules.

The first module is the Vision module, which executes all image processing and vision tasks. Using this module, the robot scans its environment and calculates the depth to features around it. The location of the goal, if observed, is obtained as well.

This module captures images with the video-camera via a Matrox Meteor video card. Use of the Matrox Imaging Library (MIL) augmented custom-developed imaging code.

Due to camera limitations and the complexity of the environment, the range data obtained by the robot are uncertain. The level of evidence of being occupied m_i of each registered 3D point $\vec{r_i}$ is calculated according to the visual sensor model presented in Chapter 3. The acquired range data are then fused in the second module.

The second module is the Thinker module. It consists of two sub-modules: vision understanding and map extraction. For vision understanding, range data $\{\vec{r_i}, m_i\}$ are fused and a 3D occupancy model of the world represented by the occupancy function $m = F(\vec{r})$ is is built. This function is built using the regression-based fusing technique introduced in Chapter 4 and is represented as a binary tree of minima and maxima of linear functions. As described in Section 5, this representation is not only optimal space-wise, but it also allows efficient extraction of a 2D polygon of the area available for navigation.

In the map extraction submodule, a 2D polygon of the area available for navigation is extracted from the occupancy model. Lists of obstacle points and exploration points are derived using the polygon. These two lists are then added to previous lists and are used, along with the location of the goal, if known, to perform path planning.

The third module, the Driver, also consists of two sub-modules. The first sub-module is the path planner. It uses a reinforcement learning technique to build the value function to be used in making the navigation decision, using the reinforcement rewards obtained in the previous module. It computes a suggested direction for the robot to move and is implemented as a separate piece of software to permit it to function in a simulation mode to test reinforcement learning technique described in Chapter 5 on simulated data. The second submodule sends motion commands to the robot and performs some consistency checking on the output of the planner.

All three modules are active in each cycle of a robot's navigation. Cycles are repeated until the robot manoeuvres close to the goal.

Figure 6.2 shows the GUI which runs on the Vision and Thinker modules of the robot at the end of the work of the Thinker module in one of the cycles of robot's navigation. It consists of four windows. The first window shows the range data acquired by the robot projected onto the floor. It also shows a part of the environment observed by a camera at any moment. The second window shows the 3D occupancy model of the world built by the robot. The circular appearance of the data is due to the uniform sampling in the coordinate system we use. The third window shows the navigation polygon extracted from the 3D occupancy function and the fourth window is a log window, showing a number of iterations and linear pieces required to build a model.

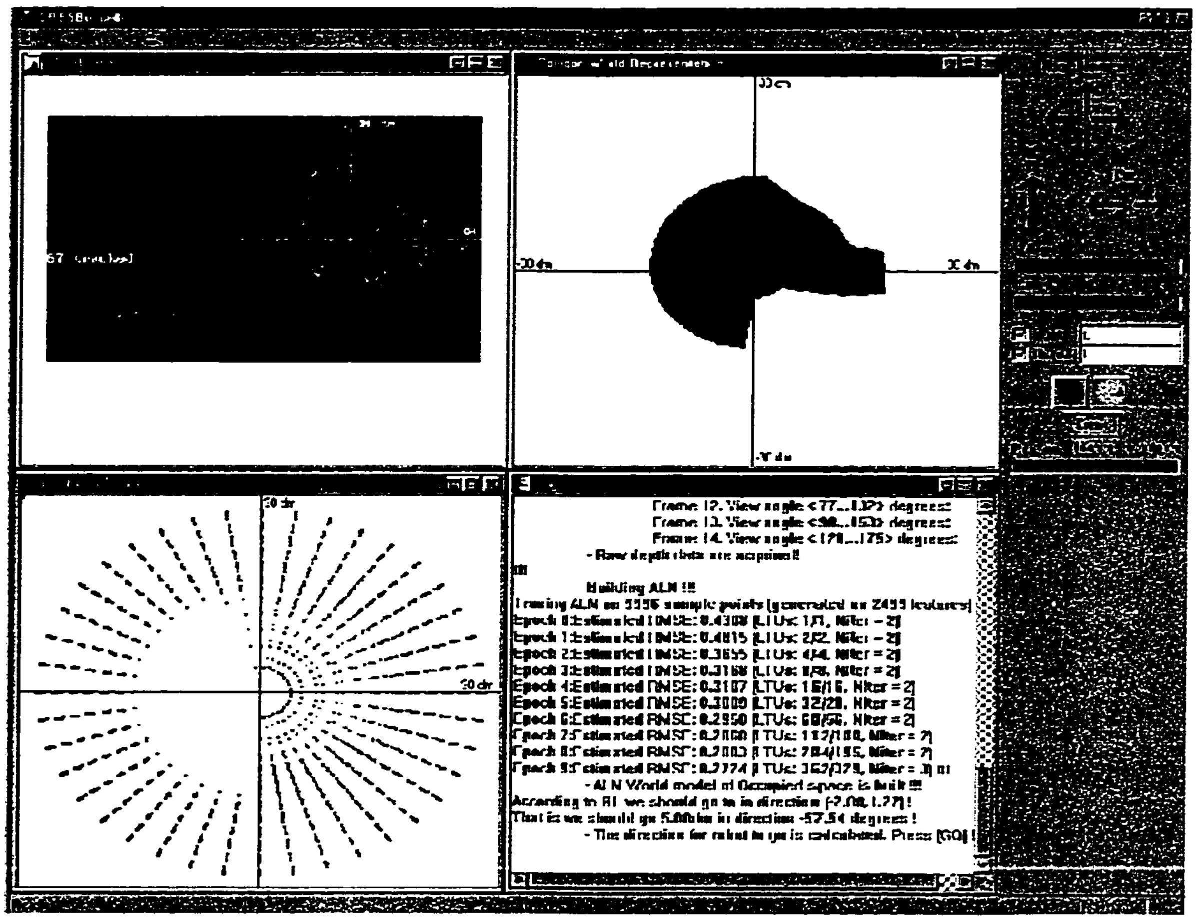


Figure 6.2: *GUI of the robot running on Vision and Thinker modules.*

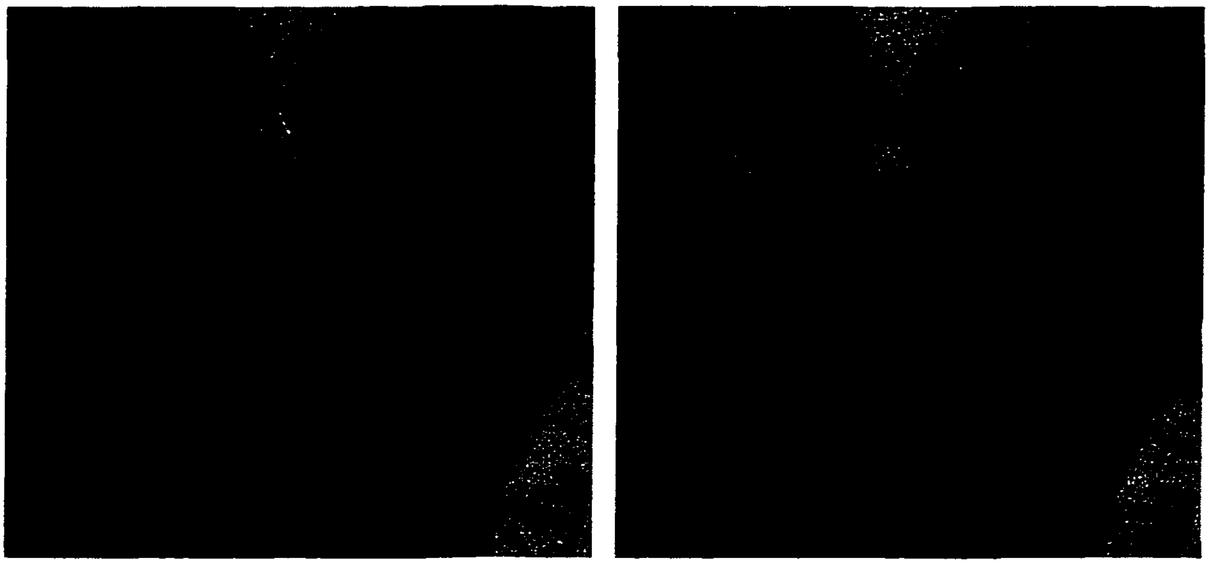


Figure 6.3: *Robot Boticelli exploring the room.*

6.4 Experimental Results

The results of executing each individual module of Boticelli were presented in the previous chapters. Figures 3.9, 4.8, 5.5, 5.6 and 6.2 show data obtained from the runs of the robot. In this section we summarize the results.

In our application, the robot explores a room shown in Figure 6.3 in order to find a goal hidden behind a wall, which is a green triangle glued on white paper seen on the back wall in the figure. An approximate plan of the room used in the experiments is shown in Figure 6.4. The figure also shows approximately the moves of the robot on its way to the goal.

During the course of exploration, in each of its locations, the robot acquires range data, which are then converted to a 3D occupancy world model. The constructed model is then used to provide the robot with a navigation map consisting of the list of obstacle and exploration points, which is used by the robot to decide where to go. A reinforcement learning method is applied at this stage.

With the technique described in the paper the robot was able to find the goal, while maintaining the world model. We used less than 32 linear pieces in the occu-

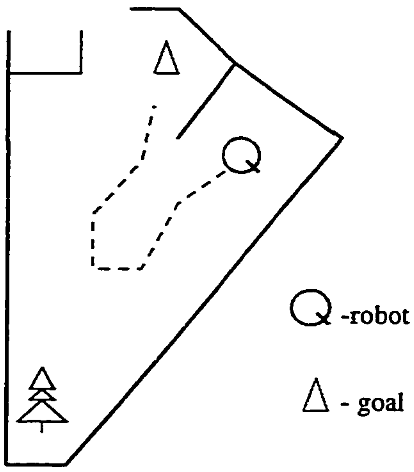


Figure 6.4: *Approximate plan of the room.*

pancy function representation. This appeared to be sufficient for the problem and did not take much time to evaluate. More specifically, it takes approximately the same amount of time to build a model as it takes to collect the range data in the vision stage, that is, less than two minutes on a Pentium Pro 200 MHz computer.

It has been observed that in most cases a single camera range sensor provides adequate range data for 3D modeling. In the cases, when visual features did not have a prominent change of the intensity derivative in the vertical direction (e.g. leaves of the tree which was put in the corner of the room, vertical edges of chairs etc.), the camera ignored them. This resulted in some parts of the environment not being covered with features, which, in turn, resulted in deterioration of the quality of modeling. This can be considered as a drawback of the visual sensor.

When there were enough 3D features present all around the robot, fusion based regression of those features yielded adequate occupancy models of the world, where by "adequate" we mean that prominent occupied and empty areas were clearly observed in all directions around the robot. The drawback of the fusion technique was noticed in the fact that if there were a lot of outliers present, they would be fused as actual range data, narrowing the empty area around the robot. However, this drawback can be controlled by changing the allowable precision of fitting. For example, by setting the allowable precision of fitting high (between 0.2 and 0.25) we insure that the occupancy function will not be overfitted. This, of course, results in using fewer linear pieces in fitting, and this is why in Figures 4.8, 5.5, 6.2 we can clearly see the circular shapes of empty areas.

As for parametric representation of the occupancy function, as soon as the function is built, no problems have been observed in extracting 2D navigation polygons. It was quite interesting to observe how by slicing the occupancy model at different heights, we could clearly see different sizes and shapes of navigation polygons. This is when the advantage of building 3D and not 2D models became the most apparent. By making use of the knowledge of the height of the robot, we could easily determine where the robot can go and where it cannot. This knowledge was used by the reinforcement-learning-based navigation policy.

As for choosing the navigation policy, it was noticed that the robot never went very close to the goal. This can be explained by the ad-hoc choice of positive and negative reinforcements given to the goal and obstacle points. — The goal, being also an observed obstacle, repelled the robot at close range.

6.4.1 Assumptions Made

There was an assumption made about the environments the robot was exploring that there were visual features present all around the robot. In our experiments this was achieved by putting camouflage cloths on otherwise featureless walls, which can be seen in Figure 6.3. If there are no features available in a part of the environment, then, because of the linear regression fitting, this could result in undesirably high occupancy values in that part. This situation however can be handled by generating a few samples points all over the space with occupancy values equal zero.

Another thing to be mentioned is that the visual range data we used in the experiments contain a lot of imprecise data. Approximately 5% of the data are estimated to be outliers. This is another reason why we build a coarse model of the world, i.e. with only few linear pieces. However, if range data are obtained by more robust range sensors like laser range finders [EHBR98, PHLG97], for instance, then there is reason to believe that the proposed piece-wise linear representation of the occupancy function would yield a better approximation of the world, if more linear pieces are used in regression.

Since the number of sample points used in building the occupancy model of an environment does not depend on the scale of the environment, it can be assumed that the proposed technique of representing occupancy models can be used equally well for representing environments at different scales.

6.5 Conclusion and Future Work

We proposed a framework for building 3D occupancy world models from uncertain visual range data. The framework yields the reduction in space consumption and the amount of calculations at the expense of model precision, allowing one to build crude 3D occupancy worlds of large scale environments in real-time. The new framework opens new areas for research concerning occupancy world modeling. Directions for further research are presented in the next chapter.

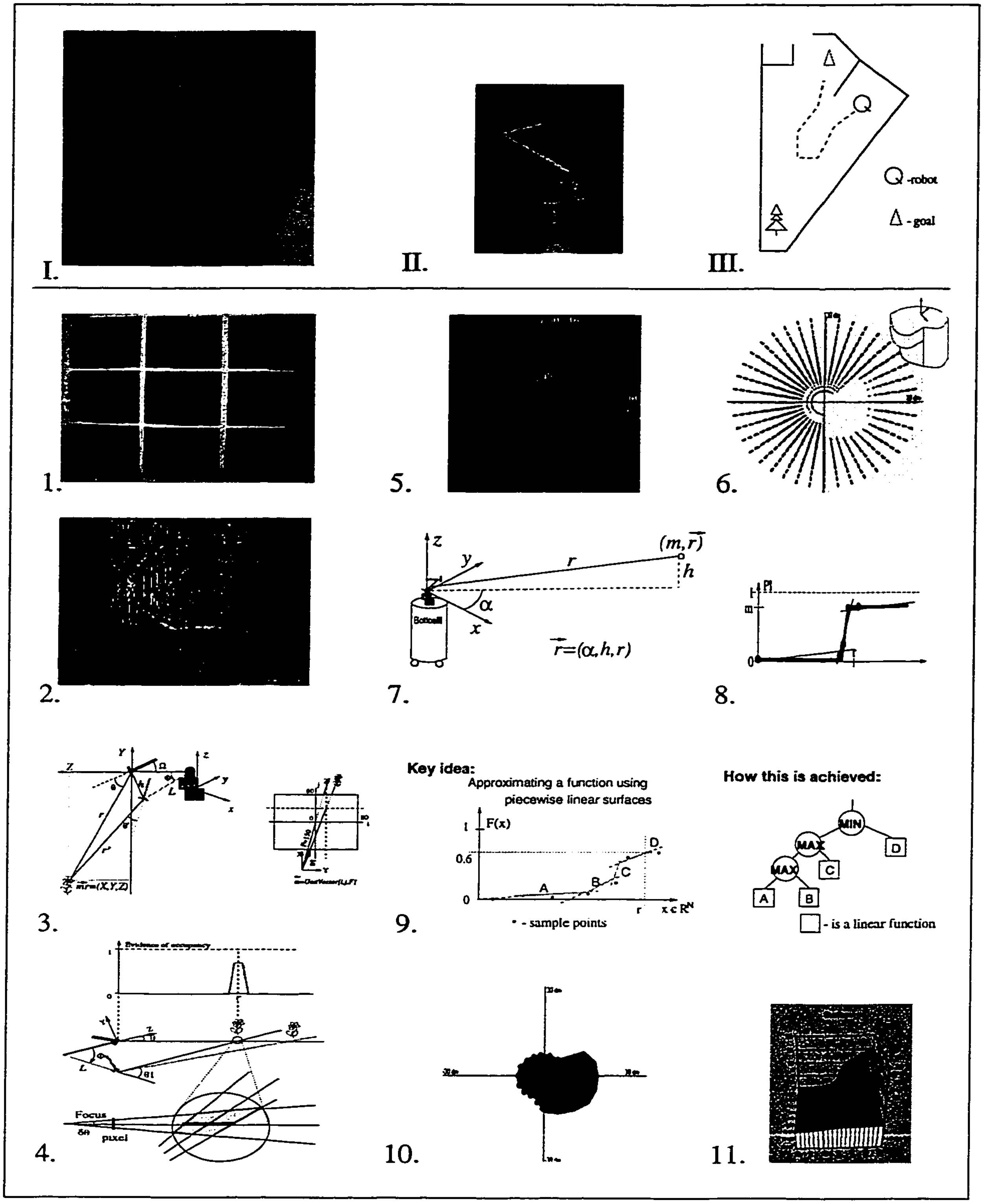


Figure 6.5: *From Image Processing to Navigation Planning: The Framework for Vision-based World Exploration Using Occupancy models.*

Chapter 7

Conclusion

7.1 Contributions

We are standing on the edge of the new millennium. A millennium is a long time in the history of humankind. If we consider the advance of science during in the last millennium, we might as well be able to foresee the magnitude of the changes still to come in the next millennium. For those changes to happen we have to create a basis to build on. We have to look for new open-minded ideas. We have to challenge the conventional approaches.

This dissertation challenges one of these approaches — the occupancy approach used for world modeling from uncertain range data. We expose the problems of this approach. This can be considered as the first main contribution of the thesis. The problems we have identified are the following.

1. Vision-based range sensors are very attractive for world modeling. However, building sensor models of unreliable video system is computationally intractable, if conventional probability-based approaches are used. Because of that up till now only highly calibrated visual range sensors are used for the problem, which are usually expensive and/or slow.

2. Rules used to combine uncertain range data assume independence of noise in data, contrary to the usual situation. This results in non-sensical results when

applied to repetitive range data.

3. The grid representation, which is the foundation of the occupancy approach, requires storing and processing tremendous amounts of data, which makes modeling of large scale 3D environments practically impossible. It is also unsuitable for efficient map extraction and navigation planning.

The dissertation showed the ways to resolve these problems, trying not to leave any of them without attention. This set of solutions is the second major contribution of the thesis. Each of these solutions can be considered independently from each other or they can be combined in a one framework. Let us review these solutions.

7.1.1 Evidence-based Visual Range Sensor

We developed an evidence-based sensor model design technique which allows one to build a 3D range sensor using an inexpensive off-the-shelf video camera. We showed how to design such a single camera range sensor which registers 3D range data around a robot in real time, where range data consists of both a 3D vector of a visual feature and the value of evidence which shows the amount of uncertainty associated with the depth measurement.

We resolved the hardware problems associated with the design of the single camera range sensor, which includes the problem of the stereo rig design, as well as the software problems such as the problem of the stereo error analysis of the sensor. We also demonstrated the advantages of the proposed single camera range sensor for world exploration problems.

7.1.2 Regression-based Range Data Fusion Technique

We developed a new regression-based technique for fusing uncertain range data. The technique has the semantical advantage over other range data fusion techniques in that it does not assume range data to be independent and can therefore be applied on repetitive range data, including those obtained by a single-camera visual sensor.

This technique has also two other major advantages over conventional rule-based data fusion techniques. First, it automatically finds a number of parameters needed for modeling an environment. These parameters are the parameters of linear equations, which fit best the occupancy function of the environment. Second, it can be used for building parametrically represented occupancy models which have many advantages over grid-based occupancy models.

7.1.3 Parametrically Represented Occupancy Models

We developed methods for building and using parametrically represented occupancy models. In particular, we showed that occupancy models of the world can be parametrically represented using a min-max tree combining linear functions and showed how to build parametrically represented occupancy functions from uncertain range data using the regression-based fusion technique. We developed a technique for extracting 2D navigation maps from parametrically represented occupancy models. This technique uses the inverse of the occupancy function and is much more efficient than conventional ray tracing techniques.

The major drawback of parametrically represented occupancy models is seen in the fact that they are rather crude; their precision may not suffice for such application as building virtual environments. However, they consume little memory and are convenient for world exploration problems.

7.1.4 Putting The Methods Together

In the dissertation we put all the described above solutions together, establishing thereby a new framework for building 3D occupancy world models from uncertain visual range data. The framework yields the reduction in space consumption and the amount of calculations at the expense of model precision, allowing one to build crude 3D occupancy worlds of large scale environments in real-time. The new framework also opens new areas for research in occupancy world modeling.

7.2 Future Work

What is the future of the ideas and results presented? Will they be used by other generations of scientists or will they lie on a library shelf covered with dust?

I will not be as optimistic (or should I say "pessimistic"?) as Hans Moravec about the future of robots. He asserts in his recent book [Mor98] that by the middle of this century robots will "run companies and do the research". Robots still have a long way to go to approach a human's ability of world comprehension. There is still a lot of work and research to be done in many areas of Computing Science to make robots more sophisticated than they presently are. This also concerns the area of vision-based occupancy world modeling which we considered in the dissertation. Much remains to be done in the area. However, we believe that the strategies proposed in the dissertation for building sensor models, combining uncertain range data, and representing and using occupancy functions provide the basis for new applications of the occupancy approach and will promote the development of the approach in both world modeling and robot navigation.

Below we present an overview of further improvements of the solutions and techniques proposed in the dissertation. Our vision of further research in all areas of Computing Science as related to the vision-based occupancy modeling problem is outlined.

7.2.1 In Computer Vision and Range Sensoring

The technique we use for feature selection and tracking, while simple and not time consuming, suffices for applications like the one described above. However, if there is a need for a more precise depth data registration, then the following steps can be undertaken to improve the performance of a single camera stereo:

- Using better quality cameras;
- Calibrating the camera and using all intrinsic parameters of the camera in depth

calculation [BSG99]

- Rectifying the images [Mor96];
- Using an interest operator to select features [Mor96];
- Using the epipolar constraint in filtering the outliers;
- Using robust tracking approaches, e.g. like those described in [ML97, MMRK91].

Other promising directions for further improvement of the vision based range sensors include:

- Designing of a "smart" visual sensor, which selects features according to the knowledge of explored and unexplored areas thus avoiding the redundancy of computation.
- Using active light with a single camera stereo creating thereby more visual features.

As for the visual sensor model design, a better approximation of the range error should be used for large scale environments. In addition, other approaches in assigning the evidence values to registered range data can also be tried. However, since the final map of an area available for navigation is determined by a threshold on an occupancy function, this assignment seems not to affect much the navigation planning process.

It would be also interesting to apply evidence-based assignment of confidence values in designing sensor models of other types of range sensors.

7.2.2 In Range Data Fusion for World Modeling

It would be very interesting to apply the proposed regression-based fusion technique to different types of range data. In particular, range data obtained by laser-based scanners should be tried, as they do not have many outliers and are therefore more suitable for least squares regression techniques.

Different sensor models should also be tried. In the cases when the monotonicity constraint cannot be imposed on the sensor model, regression should be applied without constraining the occupancy function. In this case the quality of modeling should be examined as a function of the number of generated sample points.

As for using regression for occupancy modeling in general, we see three main directions for further research. We describe them below.

Using Other Neural Network and Regression Techniques

Since imprecision in modeling is mainly attributed to the inability of the regression technique to deal with outliers, other than ALN based regression techniques should be tried for the problem. This includes such robust regression techniques like Least Median of Squares, Least Trimmed Squares, Weighted Least Squares or M-estimators [RL87, CT94, BW88].

Using the L_1 norm instead the L_2 norm in error estimation (Eq. 4.22) is another way to improve the performance when dealing with outliers. It should be considered too.

Neural network techniques can be considered for fusing range data as well. For example, GMDH networks [MI94], which approximate functions using Volterra functional series, would be a very promising choice to try. Another network of the similar type, which uses different set of basis functions, is ALB [Hoo99]. These techniques can also be tried.

Outliers can be eliminated using the Hat matrix, which is the projection matrix on the subspace spanned on the vectors-features. The Hat matrix can be calculated on line using the techniques proposed in [Gor97] for example. This however is quite computationally expensive.

Using Belief Occupancy

In this dissertation we considered plausibility evidence assignments in building the occupancy models. A way to improve the modeling is to build the belief occupancy model of the world in addition to its plausibility occupancy model. This is the same as building both occupancy and emptiness models done by other researchers [PNDW95]. The combination of two models is more descriptive of the environment.

For the belief occupancy function to be built, the belief sensor model has to be incorporated into sensor fusion. The belief model of most sensors however is not monotonic. There are two ways then to incorporate it into fusion. The first, the most straightforward, way of achieving that is not to impose any constraints on the occupancy function and to generate sample point both in front of and behind an observed point. Another way would be to use extra postprocessing of the occupancy function after it has been built. This postprocessing will consist of finding all points in space which have occupancy higher than one and then clipping their occupancy values to zero.

Combining Several Models

Occupancy models are often used for dead-reckoning problems, that is for the purpose of finding the location of an agent with respect to the known environment. Then the question of combining local occupancy models into one global model will arise.

In this work we build local occupancy models, that is, models that are centered at the location of the robot. These models can be used for localization problems. However a technique for combining two occupancy model built at two different locations should be investigated. Parametric representation of the occupancy function makes it easy to translate and rotate the model. Yet, how to fuse two already built occupancy functions is still an open question.

7.2.3 In Applications of Occupancy Models

For world exploration, high precision of occupancy models is not required. Yet, what is required is obtaining as fast as possible a rough idea of what is around the robot. If there is a need for more precise information about a particular part of the environment, then other exploration techniques can be employed, including, for example, conventional grid-based occupancy techniques to model small parts of an environment or telepresence techniques [BBZ98], which use the help of a teleoperator in order to guide the robot using the information coming from the video-camera.

We see two major directions for future work in using parametrically represented occupancy models. First, another way of extracting exploration points should be developed. A better way to do it would be to use the belief occupancy function in addition to the plausibility occupancy function, and then to consider the points which have high uncertainty interval defined as a difference between the belief and plausibility values. Second, other ways of assigning reinforcements should be investigated. Perhaps, a better way would be to assign a reinforcement to the entire line segment included in the boundary, as the current way of extracting exploration and obstacle points deviates from our primary goal of using the advantages of parametric representation of the occupancy function.

Concluding, we would like to add that, being a multi-disciplinary problem, vision-based occupancy modeling would benefit the most from the collaboration of scientists of different scientific backgrounds. We believe that this collaboration, which is the key to the success of the projects like those considered in the dissertation, will inevitably happen.

Bibliography

[ACFM97] O. Aycard, F. Charpillet, D. Fohr, and J. Mari. Place recognition using hidden markov models. In *International Conference of Intelligent Robots and Systems proceedings, Grenoble, France*, pages 1741–1747, 1997.

[ACG99] W. W. Armstrong, B. Coghlan, and D.O. Gorodnichy. Reinforcement learning for robot navigation. In *International Joint Conference on Neural Networks (IJCNN'99) CD-ROM proceedings, Washington DC, July 21–23*, 1999.

[ACT95] W. W. Armstrong, C. Chu, and M. M. Thomas. Using adaptive logic networks to predict machine failure. In *Proc. of the 1995 Workshop on Environmental and Energy Applications of Neural Networks", World Scientific, Richland, USA, pp. 97-107*, 1995.

[AKST95] W. W. Armstrong, A. Kostov, R. B. Stein, and M. M. Thomas. Adaptive logic networks in rehabilitation of persons with incomplete spinal cord injury. In *Proc. of the 1995 Workshop on Environmental and Energy Applications of Neural Networks, World Scientific, Richland, USA, pp. 154-171*, 1995.

[AL98] William W. Armstrong and Darwin Li. A new technique for reinforcement learning for control. In *IEEE Conf. on Systems, Man and Cybernetics, LaJolla CA, IEEE cat. no. 0-7803-4778-1/98 (CD-ROM)*, 1998.

[Arm98] William W. Armstrong. Reinforcment learning applied to simulated basketball balancing. In *Proc. Int. ICSC-IFAC Symp. on Neural Computation, Vienna, (CD-ROM), Sept 21 - 25*, 1998.

[AT96a] W.W. Armstrong and M.M. Thomas. Adaptive logic networks. *Handbook of Neural Computation, Section C1.8, IOP Publishing and Oxford U. Press, ISBN 0 7503 0312 3*, 1996.

[AT96b] W.W. Armstrong and M.M. Thomas. The Atree 3.0 educational kit with user guide. In *available via anonymous ftp from ftp.cs.ualberta.ca [129.128.4.241] in pub/atree3.tar.Z (for Unix systems)*, 1996.

[Aya91] Nicolas Ayache. *Artificial Vision for Mobile Robots (translation from French, 1989, INRIA).* The MIT Press, London, 1991.

[Bal81] J.F. Baldwin. Fuzzy logic and fuzzy reasoning. *Fuzzy Reasoning and Its Applications, E.H. Mamdani and B.R. Gaines (eds.)*, 1981.

[BBC+95] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *AI Magazine*, 16:1, 1995.

[BBS+97] Francois Belair, Eric Bourque, Robert Sim, Phil Ciaravola, Yiannis Rekleitis, and Gregory Dudek. Autonomous exploration: An integrated systems approach. In *AAAI-97 Proceedings, AAAI/MIT Press, pp.779-780*, 1997.

[BBZ98] J. Baldwin, A. Basu, and H. Zhang. Hybrid video for telerobotics. In *Intelligent and Autonomous Systems IAS-5, Sapporo, Japan, June 1-4, pp. 78-85*, 1998.

[BCS97] O. Basir, A. Chung, and H. Shen. A decentralized approach to sensory data integration. In *International Conference of Intelligent Robots and Systems proceedings, Grenoble, France*, pages 1409–1415, 1997.

[BEFW97] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, 14(4):231–249, 1997.

[Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[BGHK92] F. Bacchus, A. J. Grove, J.Y. Halpern, and D. Kolle. From statistics to beliefs. In *Proceedings of AAAI-92 (Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 602-608*, 1992.

[BH87] S. Blostein and T. Huang. Error analysis in stereo determination of 3-D point postition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):752–772, 1987.

[Bre96] Leo Breiman. Bagging predictor. In *Machine Learning, 24, pp. 123-140*, 1996.

[BS97] A. Berler and S.E. Shimoni. Bayes network for sensor fusion in occupancy grids. In *Uncertainty in Artificial Intelligence Conference Proceedings (http://www.sis.pitt.edu/~dsl/uai.html)*, 1997.

[BSG99] Bajcsy R. Beauchemin S.S. and Givaty G. A unified procedure for calibrating intrinsic parameters of fish-eye lenses. In *Vision Interface (VI 99), pp. 272-279, Trois-Rivieres, Canada, May 18-21*, pages 272–279, 1999.

[BW88] D.M. Bates and D. Watts. *Nonlinear Regression analysis and its Applications*. New York:John Wiley, 1988.

[CT94] B. Cheng and D.M. Titterington. Neural networks: A review from a statistical perspective. *Statistical Science, 9,2-30*, 1994.

[EHBB+97] S.F. El-Hakim, P. Boulanger, F. Blais, J.-A. Beraldin, and G. Roth. A mobile system for indoors 3-D mapping and positioning. In *Proceedings of the Optical 3-D Measurement Techniques IV, pp. 275-282*, 1997.

[EHBR98] S.F. El-Hakim, C. Brenner, and G. Roth. A multi-sensor approach to creating accurate virtual environments. *ISPRS Journal of Photogrammetry and Remote Sensing, 53, 379–391*, 1998.

[Elf86] A. E Elfes. A sonar-based mapping and navigation system. In *IEEE International Conference on Robotics and Automation, San Francisco*, 1986.

[Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. In *IEEE Computer Magazine, special issue on Autonomous Intelligent Machines, pp. 46-58*, 1989.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision – A Geometric Viewpoint.* The MIT Press, Cambridge, 1993.

[FBE94] L. Feng, J. Borenstein, and H.R. Everett. Where am I? Sensors and methods for autonomous mobile robot. Technical Report UM-MEAM-94-21, The University of Michigan, 1994.

[FH95] N. Friedman and J.Y. Halpern. Plausibility measures: a user's guide. In *Proceedings of the Eleventh Conference on Uncertainty in AI, pp. 175-184*, 1995.

[For98] Johan Forsberg. Mobile robot navigation using non-contact sensors. *Doctoral thesis Lulex University of Technology, November*, 1998.

[Fri91] J.H. Friedman. *Multivariate Adaptive Regression Splines.* Annals of Statistics 19, 1-141, 1991.

[FS81] J.H. Friedman and W. Stuetzle. *Projection Pursuit Regression.* Journal of the American Statistical Association, 1981.

[GA99a] D.O. Gorodnichy and W.W. Armstrong. A parametric alternative to grids for occupancy-based world modeling. In *Quality Control by Artificial Vision (QCAV'99) conference proceedings, May 18-21*, 1999.

[GA99b] D.O. Gorodnichy and W.W. Armstrong. Single camera stereo for mobile robots. In *Vision Interface (VI'99) conference proceedings, pp. 73-80, May 18-21*, 1999.

[GA00a] D.O. Gorodnichy and W.W. Armstrong. Building and using parametrically represented occupancy models in mobile robot world exploration. *Robotics Today*, 13(2):1–6, 2000.

[GA00b] D.O. Gorodnichy and W.W. Armstrong. Neurocomputational approach for modeling large scale environments from range data. In *Neural Computation Simposium (NC2000) Proceedings (to appear), Berlin, May 23-26*, 2000.

[GA00c] D.O. Gorodnichy and W.W. Armstrong. Vision-based occupancy modeling. In *International Simposium on Robotics (AI/GI/VI/ISR'2000) Proceedings (to appear), Montreal, 14-17 May*, 2000.

[GAL97] D.O. Gorodnichy, W.W. Armstrong, and X. Li. Adaptive logic networks for facial feature detection. In *Lecture Notes in Computer Science, Vol. 1311 (Proceedings of 9th Intern. Conf. on Image Analysis and Processing (ICIAP'97), Vol. II), pp. 332-339, Springer*, 1997.

[GH97] A.J. Grove and J.Y. Halpern. Probability update: conditioning vs. cross-entropy. In *Proceedings of the Thirteenth Conference on Uncertainty in AI, pp. 208-214*, 1997.

[GMB98] M. Gruber, S. Meissl, and R. Bohm. Das dreidimensionale digitale stadtmodell wien, erfahrungen aus einer vorstudie. *In Vermessung und Geoinformation 83, volume 1+2, pages 29-36*, 1998.

[Gor97] D.O. Gorodnichy. Investigation and design of high performance fully connected neural networks. *PhD dissertation, National Academy of Sciences of Ukraine, Kiev, Ukraine (written in Russian). (The summary of the dissertation also appeared as a "Best Presentation" award paper under the title "The Optimal Value of Self-connection" in CD-ROM Proceedings of IJCNN'99, Washington, July 12-17, 1999)*, 1997.

[Gor99] Dmitry O. Gorodnichy. On using regression in range data fusion. In *Canadian Conference on Electrical and Computer Engineering (CCECE'99) proceedings, pp. 1345-1350, May 9-12, Edmonton*, 1999.

[GR97] D.O. Gorodnichy and A.M. Reznik. Increasing attraction of pseudo-inverse autoassociative networks. *Neural Processing Letters*, 5(2):123–127, 1997.

[Gri99] Eric Grimson. *Personal communication.* Medical Engineering, MIT, 1999.

[HC96] G.E. Hughes and M. J. Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[HM99] R. Hammel and L.M. Manevitz. Combination calculii for uncertainty reasoning: representing uncertainty using distributions. In *to appear in Annals of artificial intelligence and mathematical science*, 1999.

[Hoo99] Peter Hooper. Flexible regression modeling with adaptive logistic basis functions. *Submitted to Journal of the American Statistical Association*, 1999.

[IK98] L. Iocchi and K. Konolige. A multiresolution stereo vision system for mobile robots. In *AIIA (Italian AI Association) Workshop Proceedings, Padova, Italy*, 1998.

[JD97] C. Jennings and D.Murray. Stereo vision based mapping and navigation for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation, pp. 1694-1699*, 1997.

[Kan93] K. Kanatani. *Geometric Computation for Machine Vision.* Oxford University Press, 1993.

[KR96] S S Keerthi and B Ravindran. Use of neural and other function approximators in reinforcement learning. In *Handbook of Neural Computation, Section C 3.6*, 1996.

[Lau99] Denis Laurendeau. Personal communication. *Computer Engineering Dept., Laval University, Quebec*, 1999.

[Mal93] Xavier Maldague. Nondestructive evaluation of material by infrared thermography. In *Springer-verlag*, 1993.

[Mar97] Randy Marsden. *Personal communication.* Madenta Communications, Edmonton, 1997.

[May79] P. Maybeck. *Stochastic Models, Estimation and Control, Vol. 1 and 2.* Academic Press, London, 1979.

[ME85] H. P. Moravec and A. E. Elfes. High resolution maps from wide angle sonar. In *proceeding of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, March, pp 116-121*, 1985.

[ME88] L. H. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA*, 1988.

[MI94] H.R. Madala and A.G. Ivakhnenko. *Inductive Learning Algorithms for Complex Systems Modeling.* CRC Press Inc., Boca Raton, 1994.

[Mit94] A. Mitiche. *Computational Analysis of Visual Motion.* Plenum Press, New York and London, 1994.

[ML97] C. Menard and A. Leonardis. Stereo matching using M-estimators. In LNCS-*1296, (*Proc. of CAIP'97*)*, pages 305–312, 1997.

[MM96] M.C. Martin and H.P. Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, CMU RI, 1996.

[MMRK91] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[Mor81] H. P. Moravec. *Robot Rover Visual Navigation.* UMI Research Press, Ann Arbor, Michigan, 1981.

[Mor96] Hans P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, CMU RI, 1996.

[Mor98] Hans Moravec. Robot: mere machine to transcendent mind. In *Oxford University Press*, 1998.

[MS87] L. Mathies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics Automation*, 3/3:239–247, 1987.

[MS97] J. Miura and Y. Shirai. Vision-motion planning for a mobile robot under uncertainty. *Int. J. of Robotics Research*, 16:806–825, 1997.

[MSG91] L. Moreno, M. Salich, and D. Gachet. Fusion of proximity data in certainty grids. In *Parallel and Distributed Computing in Engineering Systems, Proceedings of IMACS/IFAC Intern. Symposium, Corfu, Greece, pp. 269-274*, 1991.

[Mul98] Jan Mulder. *Personal communication.* ARC, Calgary, Alberta, 1998.

[New99] Newsgroup. http://www.cs.orst.edu/ dambrosi/uai-archive/. *Uncertainty in AI*, 1999.

[Pea88] J. Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. In *Morgan Kaufman*, 1988.

[PHLG97] P. Payeur, P. Hebert, D. Laurendeau, and C.M. Gosselin. Probabilistic octree modeling of a 3D dynamic environment. In *Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1289-96*, 1997.

[PNDW95] D. Pagas, E. Nebot, and H. Durrant-Whyte. An evidential approach to probabilistic map-building. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 165–169, 1995.

[Pol99] Sergey A. Polivtsev. *Personal communication.* Institute for Artificial Intelligence, Donetsk, Ukraine, 1999.

[RA90] J. Rodriguez and J. Aggarwal. Stochastic analysis of stereo quantazation error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12/5:467–470, 1990.

[RGEZ99] P. Renton, M. Greenspan, H.A. ElMaraghy, and H Zghal. Plan-N-Scan: a robotic system for collision-free autonomous exploration and workspace mapping. *J. Intell. Robot. Syst. 24(3): 20*, 1999.

[RL87] P.J. Rousseeuw and A.M. Leroy. *Robust regression and outlier detection.* New York : Wiley, 1987.

[RT95] C. R. Rao and H. Toutenburg. *Linear models : least squares and alternatives.* New York : Springer, 1995.

[SB98] R.S. Sutton and A.G. Barto. *Reinforcement Learning - An Introduction.* MIT Press, 1998.

[SD97] A. Silva and P. Menezes. J. Dias. Avoiding obstacles using a connectionist networks. In *International Conference of Intelligent Robots and Systems proceedings, Grenoble, France*, pages 1236–1242, 1997.

[SRG99] Beauchemin S.S., Bajcsy R., and Givaty G. Modelling and removing radial and tangential distortions in spherical lenses. *submitted to Pattern Recognition*, 1999.

[TB96] S. Thrun and A. Buecken. Integrating grid-based and topological maps for mobile robot navigation. In *AAAI'99 conference proceedings, pp. 944-950*, 1996.

[TFB98] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 5:29–53, 1998.

[Thr95] S. Thrun. An approach to learning mobile robot navigation. *Robotics and Autonomous Systems, 15: 301-319*, 1995.

[Thr98] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *AI Journal 99(1), 21-71*, 1998.

[UD94] A. Ude and R. Dillmann. Vision-based robot path planning. In *Advances in Robot Kinematics and Computational Geometry, Kluwer, Dordrecht, pp 505–512,*, 1994.

[UD95] A. Ude and R. Dillmann. Robot motion specification: A vision-based approach. Surveys on mathematics for industry. In *Surveys on Mathematics for Industry, 5, pp. 109-131*, 1995.

[vDKG95] J. van Dam, B. Krose, and F. Groen. Neural network application in sensor fusion for an autonomous mobile robot. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 263–277, 1995.

[Vid97] Video. *MPEG*. http://www.cs.ualberta.ca/~dmitri/WolrdModeling/Demos, 1997.

[Voo95] Frans Voorbraak. Reasoning with uncertainty in AI. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 52–90, 1995.

[Voo97] Frans Voorbraak. Combining evidence under partial ignorance. Qualitative and quantitative practical reasoning. In *LNAI 1244, D. Gabbay et al. (eds.). Berlin:Springer (1997) pp. 574-588*, 1997.

[Wan94] Pei Wang. A defect in Dempster-Shafer theory. In *Uncertainty in Artificial Intelligence Conference Proceedings (http://www.sis.pitt.edu/~dsl/uai.html)*, 1994.

[WHA92] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer Series in Information Sciences, vol. 29, Springer-Verlag, New York, 1992.

[WSSB93] M. Weigl, B. Siemiatkowska, K. Sikorski, and A. Borkowski. Grid-based mapping for autonooumous mobile robot. In *Robotics and Autonomous Systems, vol. 11, no. 1, pp 13-21*, 1993.

[WW94] P. Weckesser and F. Wallner. Calibrating the active vision system Kastor for real-time robot navigation. In *Close Range Techniques and Machine Vision, ISPRS Commision V, pp 430-436*, 1994.

[Yen89] J. Yen. Gertis: A Demster-Shafer approach to diagnosing hierarchical hypothesies. In *Commincations of the ACM, pp. 573-585*, 1989.

[YSA98] B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map-building with continuous localization. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium*, 1998.

Appendix A

Epipolar Lines of Single-Camera Stereo

The calculation of the equation of the epipolar line of the single-camera stereo presented in Chapter 3 follows below.

A point observed at pixel (i, j) in the first image of the stereo pair can be observed in the second image at any pixel (i', j') which satisfies the following epipolar constraint:

$$(\vec{m}, \vec{h} * R\vec{m}'r') = 0, \tag{A.1}$$

where (see Figure 3.3)

$$\vec{m} = \begin{bmatrix} i \\ j \\ F \end{bmatrix}, \qquad \vec{m}' = \begin{bmatrix} i' \\ j' \\ F \end{bmatrix} \tag{A.2}$$

and

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & sin\Phi \\ 0 & -sin\Phi & cos\Phi \end{bmatrix}, \qquad \vec{h} = \begin{bmatrix} 0 \\ h_z \\ h_y \end{bmatrix} = \begin{bmatrix} 0 \\ -Lsin\Phi \\ L - Lcos\Phi \end{bmatrix} \tag{A.3}$$

This yields the equation of the epipolar line: $i' = K * j' + B$, with slope K and displacement B given by

$$K = i/F * (h_z * \cos\Phi + h_y * \sin\Phi)/(j/F * h_z - h_y) \tag{A.4}$$

$$B = i/F * (h_z * \sin\Phi - h_y * \cos\Phi)/(j/F * h_z - h_y) \tag{A.5}$$

In our setup the lever length $L = 21$ cm and the angle of the camera tilt rotation $\Phi = 7.7^o$. This results in the slope K varying from -0.0242 to 0.0242 for $i \in [-53, 53]$

and $j \in [-40, 40]$. Figure A shows epipolar lines for some marginal points (i, j) shown in circles.

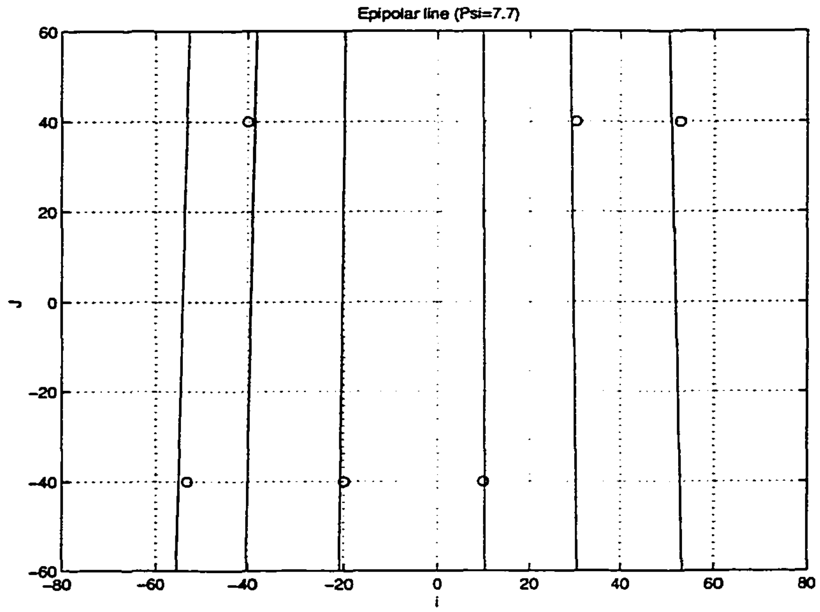


Figure A.1: *Epipolar lines of single-camera stereo.*