University of Tennessee, Knoxville

TRACE: Tennessee Research and Creative Exchange

Masters Theses

Graduate School

12-1997

Volumetric modeling through fusion of multiple range images with confidence estimate

David L. Elsner

Follow this and additional works at: https://trace.tennessee.edu/utk_gradthes

Recommended Citation

Elsner, David L., "Volumetric modeling through fusion of multiple range images with confidence estimate. " Master's Thesis, University of Tennessee, 1997.
https://trace.tennessee.edu/utk_gradthes/10513

This Thesis is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Masters Theses by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a thesis written by David L. Elsner entitled "Volumetric modeling through fusion of multiple range images with confidence estimate." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

R. T. Whitaker, Major Professor

We have read this thesis and recommend its acceptance:

M. A. Abidi, Jens Gregor

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by David L. Elsner entitled "Volumetric Modeling Through Fusion of Multiple Range Images with Confidence Estimate." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

R. T. Whitaker, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

VOLUMETRIC MODELING THROUGH FUSION OF MULTIPLE RANGE IMAGES WITH CONFIDENCE ESTIMATE

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

David L. Elsner

December 1997

ACKNOWLEDGEMENTS

I would first like to thank my parents, Lawrence and Janice Elsner, for their support over the last couple of years. Thanks to Dr. M. A. Abidi for selecting me for the Graduate Research Assistant position in the Imaging, Robotics and Intelligent Systems laboratory that made obtaining my master's degree financially possible. I also wish to thank my advisors, Dr. R. T. Whitaker and Dr. M. A. Abidi, for their guidance throughout my program. Thanks also to the members of my committee, Dr. R. T. Whitaker, Dr. M. A. Abidi and Dr. J. Gregor for their help and constructive criticism.

The work in this thesis was supported by the DOE's University Research Program in Robotics (Universities of Florida, Michigan, New Mexico, Tennessee, and Texas) under grant DOE–DE–FG02–86NE37968. Additional support was provided by Mechanical Technology Incorporated and the U.S. Department of Energy Federal Energy Technology Center under grant DE–AR21–95MC32093.

Thanks to R. E. Barry and the Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, Managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract DE-AC05-96OR22464. They provided the Coleman laser range data.

ABSTRACT

This thesis describes our investigation of creating 3-D models from multiple range images. There are many aspects to this research. We begin with an evaluation of the occupancy grid approach to fusing range data. After our evaluation, we develop a confidence metric that provides additional information about the workspace. Our research in the occupancy grid method shows that it is not ideal for our application using range images. A review of the volumetric modeling literature leads us to believe that we can develop a new technique that has the advantageous properties of the occupancy grid but is designed to be as efficient with range images as other volumetric approaches.

The second half of this thesis describes the new volumetric technique that we have developed. It presents the theory behind the approach, develops the method for combining information from several views, and tells how we extract a surface to create the final model. Finally we give several examples of our technique on real and synthetic data sets.

TABLE OF CONTENTS

CHAPTER PAGE

1. Introduction **1**

2. Previous Work **5**

2.1 Review of Occupancy Grids 5

2.1.1 Occupancy Grid Values 6

2.1.2 Data Fusion Methods 7

2.1.2.1 Bayesian Estimation 7

2.1.2.2 Dempster-Shafer Theory 8

2.1.2.3 Fuzzy Logic 9

2.1.2.4 Neural Networks 10

2.2 Volumetric Modeling 10

3. Occupancy Grid Method **13**

3.1 Review of the Occupancy Grid Method 13

3.2 The Confidence Metric 16

3.2.1 Confidence on Probability 17

3.2.2 Derivation of Confidence Metric 18

3.2.2.1 Variance Combination Formula 20

3.2.2.2 Variance to Confidence 25

3.2.3 Confidence Map Properties 26

3.3 Occupancy Grid and Confidence Map Results 35

3.4 Summary 42

CHAPTER PAGE

4. Volumetric Modeling Technique **44**

4.1 Motivation 44

4.2 Theory 46

4.2.1 Modeling the Range Image 46

4.2.2 Fusing Multiple Range Images 62

4.2.3 Extracting a Surface 64

4.3 Confidence Metric 65

4.3.1 Number of Sensor Readings 65

4.3.2 Slope-Based Confidence 65

4.3.3 Slope-Normal Confidence 67

4.4 Summary 69

5. Evaluation and Discussion **70**

5.1 Reconstruction of Synthetic Models 70

5.1.1 Confidence Measure on Synthetic Data 78

5.2 Reconstruction using Real Data 79

5.2.1 Object Modeling 82

5.2.1.1 Results 86

5.2.2 Scene Modeling 99

5.2.3 Pixel-by-Pixel Noise Estimate 105

6. Conclusions **108**

BIBLIOGRAPHY **111**

APPENDICES **117**

A. Derivation of Variance Equations **118**

CHAPTER PAGE

B. Confidence Metric and Higher-Order Uncertainty **128**

B.1 Violation of the Subjective Theory 131

B.2 Infinite Hierarchy of Probabilities 132

B.3 Reduction to First Order Uncertainties 132

B.4 Cognitive Imprecision 133

B.5 Uncertainty of Underlying Conditions 135

VITA **137**

LIST OF FIGURES

FIGURE PAGE

3.1 An occupancy grid divides 3-D space into rectangular volume elements known as voxels. 13

3.2 Occupancy grids are used to model a workspace. 14

3.3 The probability of occupancy function resulting from a single sensor reading of 70. 15

3.4 The probability of occupancy is increased with multiple consistent sensor readings. 16

3.5 Derivative of combination formula with respect to P1. 22

3.6 Derivative of combination formula with respect to P2. 22

3.7 Four plots of equation (3.5) with various values of P_1 and P_2. 24

3.8 A 2-D occupancy grid example. 27

3.9 Variances caused by uncertainties in the occupancy grid formulation. 28

3.10 The overall variance map. 29

3.11 Result of using the variance to confidence equation on the combined variance waveform. 30

3.12 The result of multiple consistent sensor readings in a 2-D grid. 32

3.13 The confidence map created by three nonoverlapping sensor readings. Shows how the sensor readings in different locations affect each other. . . . 33

3.14 The confidence map created by three nonoverlapping sensor readings when only updating the region where new sensor information is obtained. 34

3.15 Shows the interference effect when the trough of one sensor reading intersects the mound of another. 35

FIGURE PAGE

3.16 A waveform rotated every 30 degrees about a complete circle. 36

3.17 The simulated scene used in the following examples. 36

3.18 Occupancy grid and confidence map with range and angular standard deviation set to 1. 38

3.19 Occupancy grid with the confidences color coded onto it. 39

3.20 The occupancy grid created for the box in room scene when all sensor readings taken less than 3 degrees from the wall are eliminated. 39

3.21 Occupancy grid and confidence map with range standard deviation set to 2 and the angular standard deviation set to 0.25 41

3.22 A confidence colored occupancy grid. 42

4.1 The 1-D occupancy grid resulting from a single sensor reading of 70. 45

4.2 A simple range image of a mug. 47

4.3 A surface mesh of the range image shown in figure 4.2 47

4.4 A surface mesh of the range image shown in figure 4.2 with a small amount of noise added. 48

4.5 A sensor is located at a distance r from an object. 49

4.6 The density function arising from uniform noise with a width of 2ϵ. 49

4.7 The distribution function arising from uniform noise with a width of 2ϵ. . . 49

4.8 Outliers will alter the shape of the probability density function. 51

4.9 This is the distribution function that results from letting the region in which outliers may occur to $\pm\infty$. 52

4.10 A diagram that shows that a region may be occluded from one sensor but not another. 52

4.11 An altered distribution function that takes into account the possibility of other surfaces that may be seen from other directions. 53

4.12 This figure shows how we project voxel vertices onto the range image. . . . 55

FIGURE PAGE

4.13 How we find the certainty value for a voxel from the value 'd.' 61

4.14 This is a plot of the super-Bayesian operator. 62

4.15 A voxel vertex maps to point 'A.' 68

5.1 The ideal mug and gripper models. 72

5.2 Twelve synthetic range image of the ideal mug model. 73

5.3 Twelve synthetic range image of the ideal robotic gripper model. 74

5.4 Reconstruction of the mug and gripper using a 80×80×80 volumetric grid. . 75

5.5 A reconstruction of the mug on a 80×80×80 grid with uniform noise added. 76

5.6 A reconstruction of the gripper on a 80×80×80 grid with uniform noise added. 77

5.7 The ideal reconstruction of the mug model using the number of times confidence measure. 78

5.8 The effect of differing noise levels in the slope-based confidence metric. . . . 80

5.9 The effect of varying levels of noise on the slope-normal confidence metric. . 81

5.10 A reference range image that we used to find the distance from the range finder to the center of rotation. 83

5.11 This figure shows how we align misregistered range images. 85

5.12 Six range images taken of five reams of printer paper stacked on top of each other. 87

5.13 Surface plots of the first two paper range images. 87

5.14 Two views of the model reconstructed from the paper images on an 80×80×80 grid. 88

5.15 Twelve range images taken of the 'shapes' test object. 90

5.16 Surface plots of the first two 'shapes' range images. 91

5.17 The model reconstructed from the 'shapes' range data on an 80×80×80 grid. 92

FIGURE PAGE

5.18 Twelve range images taken of a styrofoam head sitting on a ream of printer paper. 93

5.19 Surface plots of the first two head range images. 94

5.20 Two views of the reconstructed head model on an 80×80×80 grid. 95

5.21 Twelve range images taken of a monitor sitting on a ream of printer paper. 96

5.22 Surface plots of the first two monitor range images. 97

5.23 Two views of the reconstructed monitor model on an 80×80×80 grid. . . . 98

5.24 Three range images taken from a Coleman scanner. 100

5.25 The reconstruction of the cone scene using two of the three range images on a 200×200×200 grid. 101

5.26 The reconstruction of the cone scene using all three of range images on a 200×200×200 grid. 102

5.27 The slope-normal confidence metric used with one 'cone' image from the Coleman scanner. 104

5.28 The reconstruction with three views color coded with confidence using the slope-normal method. 104

5.29 A comparison of two types of noise estimates. 106

5.30 For this reconstruction we median filtered the second image and set its noise estimate to five times the noise estimate of the other two images 107

CHAPTER 1

Introduction

The creation of 3-D computer models of objects from sensor information can be an important step in many applications. These include reverse engineering, automatic creation of CAD models, object recognition, inspection tasks, medical imaging, computer graphics and remote mapping.

An important consideration in automatic model building is selecting the type of sensor to use to obtain information about the object. Laser range finders that produce range images are a natural choice. Range images give what is commonly referred to as 2.5-D information about a scene. That is, they provide depth information in one direction but they do not provide a full 3-D description of an object because not all surfaces of an object can be seen by a single range image. To obtain a full 3-D description of an object, multiple range images from several viewpoints must be taken and combined into an overall model.

This task is not as simple as it may at first appear to be. There are several challenges that need to be overcome. A laser rangefinder's output is an array of 3-D points; how does one represent these points to produce a 3-D model? The range images are taken from multiple viewpoints; how does one properly align these images? Noise in the scanning process means that range images are inherently uncertain; how does one overcome this uncertainty to get a good 3-D model?

The alignment problem, commonly known as registration, has been addressed by several researchers [1, 2, 3, 4] and will not be investigated in this thesis. We will assume that the range images have been registered prior to applying our algorithm.

The remaining two questions pertain to the *fusion* of range data from multiple view-

points. Surface-based frameworks and volume-based frameworks are two strategies for fusing range data. A surface-based approach typically creates a triangle mesh from each of the range images. These techniques commonly do some kind of averaging in regions that are covered by several range images to reduce noise. Surface-based methods also require a procedure to combine meshes at the edges to create a continuous model.

A volume-based approach uses a volumetric grid that can be thought of as a 3-D array of cubical volume elements, known as *voxels*. These methods typically use an *implicit function* and the range data to create a 3-D function in the volumetric grid for each range image. Information from multiple images is combined in a voxel-by-voxel manner to get an overall implicit function. These methods typically create a final surface model by using a form of the *marching cubes* algorithm to extract a surface where the implicit function is equal to zero.

We have chosen to use a volume-based framework because they have several advantages that make them attractive for this type of application. They provide a convenient data structure for keeping track of and combining information from multiple sources. One can calculate the effect that each range image has on each voxel and fuse the values from each range image voxel-by-voxel. Volumetric grids also provide an effective way to represent uncertainty; if one is unsure of which voxel has a specific property, the certainty of that property can be blurred among several voxels.

One disadvantage of the volumetric grid approach is that it requires a large amount of computer memory and computer time to operate in many situations. Especially if one wishes to model intricately detailed objects.

This thesis makes two main contributions to the field of 3-D modeling. First, it presents a confidence metric for the occupancy grid framework. Much research has been done with occupancy grids, however, the question of how to quantify our degree of certainty in the probability values remains open. This *confidence measure* could be useful for robot

navigation tasks and deciding which regions of a scene or object would benefit the most from additional sensor scans.

Our second contribution is a unique volumetric 3-D modeling technique. It is similar to other methods, but has several enhancements. Our formulation of the certainty function allows us to explicitly model characteristics of the range finder. Additionally, our method allows us to carve away free space around objects. This can significantly reduce some types of noise. Our technique works with the certainty that a voxel in inside an object; this is more intuitive then the weighted implicit functions used in many volume-based methods.

The remainder of this document is as follows. Chapter 2 is a review of similar work in volumetric modeling including a review of occupancy grids. Chapter 3 discusses our investigation of the occupancy grid method. It begins with an overview of the theory behind the method. Then we show our derivation of the confidence metric used to complement the occupancy grid technique. Finally, chapter 3 gives some results of the confidence metric applied to a synthetic scene.

Our evaluation of the occupancy grid method showed several opportunities for improvement. We decided to create our own technique that would be more effective with the range data obtained from a *time-of-flight* laser range finder that we use. Chapter 4 begins by describing in more detail why we chose to develop our own method. Then it explains the method itself and its associated confidence metric.

Chapter 5 contains several examples of our volumetric technique and confidence metric in action. We begin by showing the results of the method applied to a couple of synthetic objects and give a discussion of the results. Then we show how the method performs on data taken from an actual rangefinder. Chapter 6 gives our conclusions about our work in volumetric modeling.

Finally, we include a couple of appendices that we hope will aid the reader. Appendix

A gives the step-by-step derivation of the confidence metric for the occupancy grid method. Readers who have a working knowledge of occupancy grids may find this section interesting. Appendix B contains a review of higher-order uncertainty. This is a controversial subject; hence we give a brief discussion of higher-order uncertainty and explain why it is useful for our application.

CHAPTER 2

Previous Work

The previous work in modeling 3-D objects can be divided into two main categories, surface-based approaches and volume-based approaches. Surface-based approaches attempt to model an object's surface directly from the range data. Soucy et. al. [5, 6], Chen and Medioni [7, 8], Ferrie et. al. [9], and Turk and Levoy [10] have done some work in this area. Volume-based approaches use an intermediate step of representing the range data on a 3-D grid before extracting a surface for visualization.

This chapter gives a brief review of work that other researchers have done in volumetric modeling. The first section is dedicated to occupancy grids. A significant amount of work has been done in this area. We have classified the occupancy grid research according to the data fusion method used to update the grid. The final section reviews other types of volumetric modeling research. These methods are closely related to the volumetric technique that we have developed and describe in chapter 4.

2.1 Review of Occupancy Grids

Occupancy grids, also known as *evidence grids* or *certainty grids*, are tessellations of 2-D or 3-D space into regions [11]. In 3-D rectilinear space, these regions are called volume elements (voxels). Each cell contains information about the space that it represents.

Occupancy grids provide an effective framework for the fusion of data from multiple sensors and sensing positions. The sensors are usually modeled by a probability density function that varies throughout space. When a reading is taken, the sensor model is

overlaid on the grid, and each cell is updated. Different types of sensors have differently shaped sensor models. However, because the sensor models share a common statistical framework (as a probability density function for example), they can be combined in the same grid.

Occupancy grids are primarily used in robot navigation tasks. In these applications, occupancy grids have been mostly 2-D representations of the floor plan of a scene. Each cell typically represent a rather large area, often more than 36 square inches. In robot navigation applications, occupancy grids are most often used to locate obstacles for path planning and obstacle avoidance.

2.1.1 Occupancy Grid Values

The cells of an occupancy grid contain information about the space that it represents. This information can be in many forms. In some of the early work [12], each cell contains two values, the probability that the cell is occupied and the probability that it is empty. Later it was decided [13, 14, 15] that within a Bayesian framework, only the probability of occupancy is needed for this formulation.

Environmental conditions that may effect sensor readings can also be included. For example, specular reflection is a significant problem in sonar sensing. Lim and Cho [16] introduce a range confidence factor and cell orientation probability at each cell to account for this effect. Auran, Malvig and Silven [17, 18, 19, 20, 21] store many values in their cells. These include probability state, radial limits, total number of echoes in the cell, time elapsed since first and last echo, and a "flag" parameter that is set to different values if the cell appeared or disappeared in the last scan.

In many instances, especially robot navigation applications [22, 23, 24], a binary occupancy grid is used. A cell is set to "1" if the cell is considered occupied and "0" if it is considered to be empty. Sometimes a value of "1/2" is added to reflect an "unknown"

state. Borenstein and Koren [25] store the number of times the cell is read as occupied by a sensor in each voxel.

2.1.2 Data Fusion Methods

Occupancy grids provide an effective framework for the fusion of data from multiple sensors and sensing positions. Several different fusion methods have been explored. This section reviews most of the common ones.

2.1.2.1 Bayesian Estimation

Bayesian estimation is the foundation of the most common fusion method [13, 14, 16, 15, 26]. In this formulation the sensor is modeled as a conditional probability density function. When a range value is read, the associated density function is overlaid on the occupancy grid. The updated version of the grid is determined by a form of Bayes rule acting on the old grid and the sensor model.

There are several advantages to this approach. It is founded on a solid statistical base. It has been used often in the past and has been proven to be effective in many applications. The fusion is commutative and associative; hence data can be combined in any order. Combining evidence of any value with the value of "1/2" (associated with the unknown condition) returns the original value as a result. Combining contradictory information causes the voxel value to move toward the unknown state [14]. Algorithms have been developed to compare an occupancy grid to another grid, an ideal grid, or the average expected score of a grid [26]. Schiele and Crowley [27] have developed a method to match occupancy grids using lines found with the Hough transform.

Several shortcomings have been reported in the literature. In the Bayesian approach, no distinction is made between uncertainty and lack of information [28, 29]. A large number of cells must be updated when new information is received. This makes it too

computationally expensive for real-time operation with large grids [21]. If readings from a sensor have a large variance, a greater reliance is placed on the data already in the map, than on new measurements. Some researchers feel that this makes the Bayesian estimate too conservative in these cases [29].

2.1.2.2 Dempster-Shafer Theory

Several researchers [28, 29, 30] have used a data fusion method based on the Dempster-Shafer theory of evidence. The Dempster-Shafer theory of evidence uses two values. The belief measure is a number in the interval [0,1] that is interpreted as the degree of belief that a proposition is true. The plausibility measure is a number in the interval of [0,1] that represents the extent that the evidence fails to refute a proposition [29].

Moreno, Salichs, Puente, and Gachet [28, 30] use a probabilistic sensor model with the Dempster-Shafer approach to assign three probabilities to each cell: the probability of occupancy, the probability that it is empty, and the probability that it is unknown. They list three methods to combine evidence with the Dempster-Shafer approach. They are conjunctive integration, disjunctive integration and trade-off integration. Conjunctive integration is the standard method and is referred to as Dempster's rule. In [28, 30] they make a comparison between these three methods and the Bayesian approach. It is concluded that disjunctive integration is inadequate for occupancy grid use. Conjunctive integration gives results similar to Bayesian estimation but it requires more memory and three times the operations. Trade-off integration results are not as good as Bayesian estimation but it requires fewer operations and more memory.

The primary motivation listed in the literature for using Dempster-Shafer over Bayesian estimation is that Dempster-Shafer allows for the explicit representation of ignorance [28, 29]. It can distinguish between the lack of information and uncertainty. This is especially useful in determining the difference between regions that are occluded by an

object and regions that are too far away to be accurately sensed. In [28] Moreno et. al. argue that the better handling of uncertainty in the Dempster-Shafer approach makes it possible to detect failed sensors in some instances. The Dempster-Shafer method behaves less conservatively in situations with multiple concurrent readings from noisy sensors. This allows a more rapid accumulation of data in these cases.

Several disadvantages to this method have been cited. It requires at least twice as much memory as the Bayesian since it uses two probabilities instead of one. Conjunctive integration also requires three times the number of computations [28]. Like the Bayesian, it is too computationally expensive for many real-time applications.

Zurada and Graham [31] have trained a neural network to perform Dempster-Shafer fusion on a 5×6 2-D occupancy grid. The trained network performs the fusion more quickly than can be done normally. However, because the number of output nodes on the neural network must be equal to the number of cells in the occupancy grid, this is feasible only for small grids. Other uses of neural networks with occupancy grids will be discussed later in this paper.

2.1.2.3 Fuzzy Logic

Fuzzy logic has also been used as a fusion method for occupancy grids [19, 32]. Poloni, Ulivi, and Vendittelli [32, 33] use fuzzy rules to create a sensor model. Two occupancy grids are created with them. One that contains the certainty that the voxels are empty and the other the certainty that the voxels are occupied. Additional readings are fused using the Dombi union. These two maps can be combined using fuzzy union and intersection operations to give maps of ambiguous, unexplored and free cells.

Auran and Malvig [19] create a 3-D spherical occupancy grid with a probability of occupancy in each cell. A rule-based fuzzy logic approach was used to update the cell values. Their primary motivation for using fuzzy logic was speed. They needed to update

a grid with approximately 6 million cells in real-time. This approach allowed them to update cell states with a minimum of calculation.

The advantage of fuzzy logic is that it allows great flexibility in the handling of information [32]. The use of natural language in the formation of rules makes it straightforward to customize the rules to the application. This could also be a disadvantage because it may be too application specific and may have to be modified for each application. Another drawback to rule-based fuzzy logic is that it may be more difficult to analyze the output when its not created by a rigorous mathematical formulation.

2.1.2.4 Neural Networks

Santos, Goncalves, and Vaz [34] use a 2-D, robot-centered, binary occupancy grid for their mobile robot. The robot contains 24 sonar sensors. The grid is composed of 60 irregularity shaped cells which are optimized to the characteristics of the sensors. A three layer neural network is used. The input has 24 nodes, one for each sensor. The output has 60 nodes, one for each cell in the grid. The number of nodes in the hidden layer is adjusted until the network converges. The network is trained by placing the robot in a known environment and providing it with the correct output. After training, the network is able to correctly fill the occupancy grid in similar environments much of the time.

The advantage of this approach is that the sensor errors are automatically accounted for in the learning process. This is feasible for only small grids, however, because there has to be one output node for each cell of the occupancy grid.

2.2 Volumetric Modeling

Much of the previous work with volumetric modeling has been done with binary data. This involves storing a 1 or a 0 in each cell of the volumetric grid. A 1 means that that

voxel is part of the object while a 0 means that it is not. Sakaguchi et. al. [35] use this type of representation for modeling parts to be used with CAD systems. Croteau et. al. [36] use a binary volumetric grid to model the workspace in which a mobile robot works. Tarbox and Gottschlich [37, 38] use a binary volumetric model in a process to inspect machined parts. They use a structured light range scanner to completely map the object's surface and create the volumetric model. Then they compare the model to a reference part stored in a CAD system. Wada et. al. [39] integrate multiple range images in a similar fashion. They use the marching cubes algorithm to extract a surface. They adjust the vertices of the triangle mesh to achieve a smoother surface. Li and Crebbin [40], and Chien et. al. [41] have done work directly encoding range images into the compressed octree type data structure.

Curless and Levoy [42] use a non-binary volumetric method to reconstruct surfaces from multiple range images. Each voxel of the volumetric grid has two values associated with it for each range image. An implicit function that represents the surface and a weight that is used when updating the implicit function. The implicit function ranges from a F_{min} which is negative to F_{max} which is positive. The surface is located where the implicit function is zero. The weight function is always positive, but tapers to zero behind the estimated surface. Data from multiple views are combined additively to create an overall implicit function and weighting function. These can be used later for fusing additional views. Methods to fill holes in the surface are also discussed.

Hoppe et. al. [43] have also done work using an implicit function for object modeling. Hoppe's algorithm works on a set of unorganized points. They define the *signed distance function* as the estimated distance from a point to the true surface. The *sign* signifies the side on which the point is located. They estimate the signed distance function in a 3-D region near data points and extract a surface where the signed distance function is equal to zero using a modified marching cubes algorithm.

Hilton et. al. [44, 45] have done similar work using multiple 2.5-D images. They calculate the signed distance function and a confidence function for each range image. The signed distance functions are combined using an average weighted by the confidence values. A set of rules are used to account for special circumstances such as surfaces on opposite sides of a thin object.

Our technique employs aspects of the occupancy grid method and methods based on the implicit function to create a technique well suited for data from *time-of-flight* laser range scanners. Like the occupancy grid method, we create a *certainty function* that is based on the characteristics of the scanner; however, our function is designed to facilitate surface extraction. We map the certainty function into the volumetric grid in a way similar to the way implicit functions are mapped into volumes. This avoids some drawbacks that we noticed with the occupancy grid method [46]. We combine views using the super-Bayesian combination formula to get an overall certainty function. The certainty function has a more intuitive meaning than an implicit function.

CHAPTER 3

Occupancy Grid Method

The first volumetric modeling technique that we will investigate is the occupancy grid method. This chapter begins with a brief overview of the technique. Then it presents the confidence metric that we developed to extend the method. Finally, we show some results and give our evaluation.

3.1 Review of the Occupancy Grid Method

Occupancy grids, originally developed by Elfes [47], divide 3-D space into rectangular volume elements known as voxels (see figure 3.1). Each voxel contains information about the workspace that it represents. Probability of occupancy is the most common type of information stored. This is the probability that the regions of the workspace corresponding to a particular voxel contains an object. Figure (3.2) shows a simple 2-D example. The

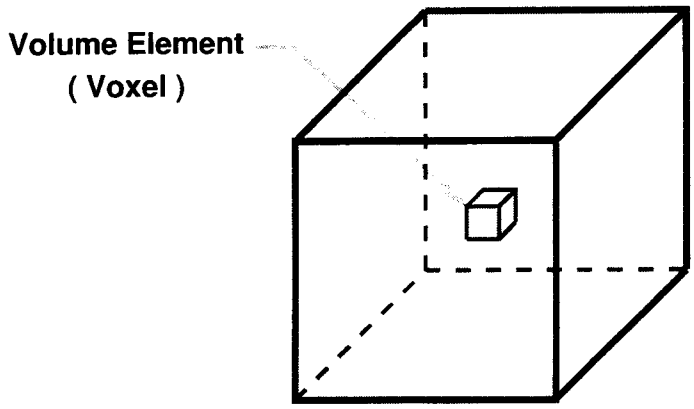


Figure 3.1: *An occupancy grid divides 3-D space into rectangular volume elements known as voxels.*

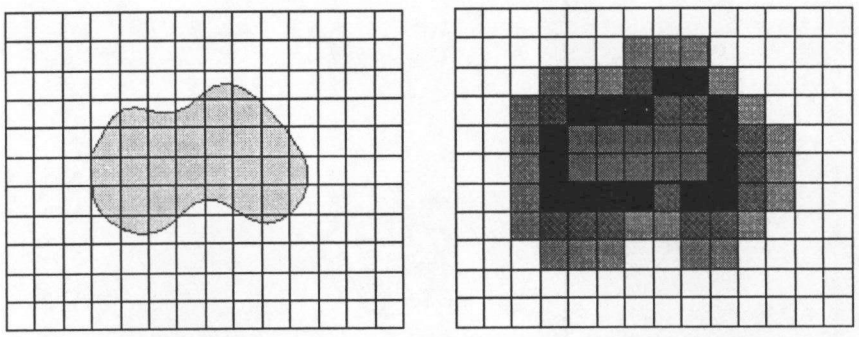


Figure 3.2: *Occupancy grids are used to model a workspace. The object on the left is modeled by the occupancy grid on the right. The darker colors in the occupancy grid represent higher probabilities of occupancy.*

figure on the left shows an object located in a workspace that has been divided into voxels. The figure on the right shows a possible occupancy grid filled with probability values. Darker colors represent higher probability of occupancy. Note that the highest probabilities correspond to regions on the object's edge.

There are a couple of obvious questions. How does one assign these probability values? How is information from multiple sensors combined into an overall representation? The occupancy grid method uses a statistical approach to answer these questions. The probability values are determined from a probability of occupancy function that is derived from the characteristics of the sensor. The method begins with the range value returned from the sensor and information about pose and location of the sensor. This information enables the method to determine the cell from which the reading was returned; this cell is the one that sensor data implies is occupied. However, due to uncertainties in the range finder, the exact location of the object causing the return is not known. The occupancy grid formulation uses an estimate of the sensor error model to determine the probability that nearby cells contain the object that caused the return. The sensor reading also contains implicit information about which voxels are empty. The beam from the range finder

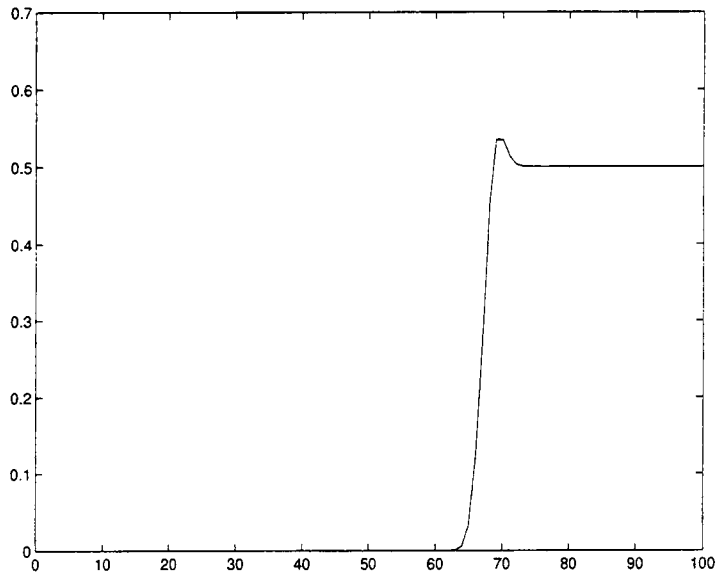


Figure 3.3: *The probability of occupancy function resulting from a single sensor reading of 70.*

must pass through the voxels that are between the sensor and the object causing the range reading; this implies that these cell are empty. Figure (3.3) shows a 1-D probability of occupancy function for a sensor reading of 70. The highest probability is located at the position returned by the sensor. The area between the range finder and the sensor reading is likely to be empty; thus this region has a low probability of occupancy. The probability decreases behind the sensor reading as chances of sensor errors this large becomes decreasingly likely. The probability drops to a value of 1/2 several cells behind the sensor reading. This region is occluded from the range finder; hence the sensor reading gives no information about this area. The value of 1/2 implies an equal chance that the voxel is occupied or empty; therefore this is referred to as the *unknown* value.

We have shown how the occupancy grid method handles a single sensor reading. However, practical applications require many sensor readings be combined into an overall model. The occupancy grid method uses a form of Bayes rule, known as the *super-Bayesian*

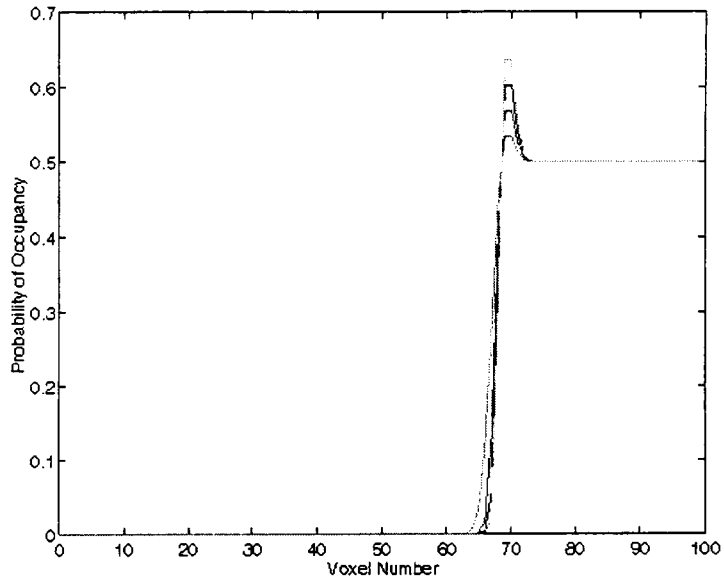


Figure 3.4: *The probability of occupancy is increased with multiple consistent sensor readings.*

combination formula, to accomplish this. We will discuss this combination formula in more detail later. The method overlays the probability of occupancy function for an individual sensor reading onto the overall grid. It then uses the super-Bayesian combination formula in a voxel-by-voxel manner to merge the individual sensor information into the world model. Figure (3.4) shows a 1-D example. We simulated receiving five sensor readings of 70 and combined them into an overall map. Note that the probability of occupancy at 70 increases as more readings are added.

3.2 The Confidence Metric

Each voxel of an occupancy grid contains the probability that that voxel is occupied. However there are many possible situations in which the same probability value may be assigned to a cell. For example, the occupancy grid is usually initialized to a value of 1/2, but a single cell may be assigned the value of 1/2 based on numerous sensor readings. The

probability value alone cannot distinguish between these different situations. It would be helpful to know the accuracy of the information upon which the probability value is based. This is the purpose of the confidence map; it is a grid that is associated with the occupancy grid that contains a number between 0 and 1 which reflects the degree of confidence in the occupancy grid probabilities.

There are several potential uses for the confidence map. One is the situation that we described above. It could eliminate the ambiguity between cells that contain the value of 1/2. Unsensed cells that are initialized to 1/2 would have a low confidence while cells containing a 1/2 based on many sensor readings would have a higher confidence. This could distinguish explored cells from unexplored cells. This is a somewhat extreme example; it is straightforward to distinguish between cells that have not been sensed and cells that have without a confidence map. One simply has to devise a method to keep track of the cells about which no information has been received. However, other similar situations are not as easily solved. For example, two cells may contain the same probability value but one may have been sensed dozens of times while the other may have only been sensed a single time. The confidence metric should reflect the fact that we are more confident about multiple consistent readings than a single sensor reading. Accurate sensor readings should create higher confidence values than less accurate readings.

Another potential use is in robot navigation. The robot could use the confidence map to plan a path though cells in which there is a high degree of belief that the occupancy grid is correct. The robot could also elect to use further sensing to increase the confidence in the areas along the most desirable path.

3.2.1 Confidence on Probability

The occupancy grid contains the probability that each voxel is occupied. Our task is to develop a confidence map that enhances the occupancy grid by giving a confidence estimate

of the probability value. Let us define $P(S = occ|r)$ to be the probability that each cell S is occupied given the sensor information r. Now we will consider the implications of defining a confidence on the probability $P(S = occ|r)$. Clearly, if we have less than 100% confidence in $P(S = occ|r)$, we have some uncertainty that the $P(S = occ|r)$ is correct. As with any situation that contains uncertainty, it is possible to define a probability which represents how sure we are of an uncertain event. In our case it would be written as $P(P(S = occ|r) = x|e)$. This is the probability that $P(S = occ|r)$ equals some value x given the evidence e. This probability of a probability is commonly known as a *higher-order uncertainty*. The confidence values that we are developing are related to this higher-order uncertainty.

The validity of higher-order uncertainty is the subject of an ongoing debate [48, 49, 50]. See appendix B for a discussion of higher-order uncertainty and an explanation of why we believe it is appropriate for our application.

3.2.2 Derivation of Confidence Metric

This section gives a description of the confidence metric that we have developed. We propose a confidence metric that is based on higher-order uncertainty. The first step is to estimate the variance in the occupancy grid probability values due to the uncertainty of the parameters used in the occupancy grid method. We do this by finding the sensitivity of the occupancy grid formulas to the considered parameters. We define the variance of the occupancy grid as the weighted sum of the individual sensitivities. Then we convert the variance estimate into a confidence value. This section explains the method to estimate the variance grid for a single sensor reading. Then it describes the procedure to combine these individual maps into the overall variance map. Finally this section discusses some of the properties of the confidence map and gives some result of the method applied to a simple scene.

One way to think of a confidence on a probability is the confidence in the parameters used to develop the probability value. Two items that have an affect on the final probability values are the assumption of the prior distribution and the variance of the sensor. In three dimensions, the sensor has variance in three directions. This makes four parameters that we will look at to determine the variance of the probability:

b - prior distribution

$\sigma(r)$ - standard deviation in the range direction

$\sigma(\theta)$ - standard deviation in the θ direction

$\sigma(\phi)$ - standard deviation in the ϕ direction

The variance for a single sensor reading is estimated similarly to a sensitivity analysis using the equation

$$V_1 = \frac{\partial P_1}{\partial b}\Delta b + \frac{\partial P_1}{\partial \sigma(r)}\Delta\sigma(r) + \frac{\partial P_1}{\partial \sigma(\theta)}\Delta\sigma(\theta) + \frac{\partial P_1}{\partial \sigma(\phi)}\Delta\sigma(\phi), \tag{3.1}$$

where V_1 is the estimated variance of the probability. P_1 is probability of occupancy obtained using the occupancy grid formulas. The values Δb, $\Delta\sigma(r)$, $\Delta\sigma(\theta)$, and $\Delta\sigma(\phi)$ are estimates of the variance of the parameters b, $\sigma(r)$, $\sigma(\theta)$ and $\sigma(\phi)$ respectively. These can be selected by the user or based on the sensor error model. We have initially selected to set them to reasonable values. The Δb is set to $1/12$. This is because b can vary from zero to one. $1/12$ is the variance of a uniform distribution in this range.

The values of the probabilities in the occupancy grid also depend on the the range reading and the probabilities can also be affected by positional errors in the sensor's placement. We chose not to include variance estimates for these parameters for several

reasons. The range measurement is discretized before it is directly used in the occupancy grid method. It is used to determine which cell caused the sensor reading and which cells had to be traversed for a sensor reading to have originated from that cell; the precise location of the reading within a single cell is not determined. This means that only range reading variations that large enough to move the reading to another cell will affect the occupancy grid equations. While deviations this large are possible, the variance estimate that we are using would involve taking a derivative of the occupancy grid equation with respect to the range measurement. Derivatives are defined assuming a small change in the affected parameter. In our case this would cause the derivative of the occupancy grid equation with respect to the range to be zero. The same is true for positional errors.

Changes in the range measurement indirectly affects the occupancy grid equations through the parameter $\sigma(r)$. However, we are already estimating the variance due to this parameter. As for the positional error, it seems best to estimate it and include it in the $\sigma(r)$, $\sigma(\theta)$, and $\sigma(\phi)$ where both the occupancy grid equations and the confidence map equations can account for it.

The values $\Delta\sigma(r)$, $\Delta\sigma(\theta)$ and $\Delta\sigma(\phi)$ reflect the accuracy at which the variance of the sensor is known. We have set them at $.2\sigma(r)$, $.2\sigma(\theta)$, and $.2\sigma(\phi)$. This was done with the assumption that the values of these parameters are well characterized. The sensor error model should provide more information on this.

The details of finding an expression for equation (3.1) from the occupancy grid equations is included in appendix A.

3.2.2.1 Variance Combination Formula

In the previous section we estimated the variances of the probabilities created by a single sensor reading. One of the strengths of the occupancy grid framework is that it allows data from many sensor readings to be fused into an overall occupancy grid. Therefore

we need to develop a similar algorithm for the variance values that allows variances from multiple sensor readings to be combined into an overall variance map.

We start with the super-Bayesian combination formula used in the occupancy grid method.

$$P_3 = \frac{P_1 P_2}{P_1 P_2 + (1 - P_1)(1 - P_2)} \tag{3.2}$$

The P_1 is the probability value contained in the occupancy grid. The P_2 is the probability value obtained from the additional sensor reading. The P_3 is the updated probability value and takes the place of P_1 in the occupancy grid. Elfes [47] explains the derivation of this equation.

We define the variance combination formula to be similar to the total derivative of the occupancy grid combination formula.

$$V_3 = \frac{\partial P_3}{\partial P_1} V_1 + \frac{\partial P_3}{\partial P_2} V_2 \tag{3.3}$$

The V_1 is the variance stored in the variance map. V_2 is the variance obtained from the additional sensor reading. V_3 is the updated variance value and replaces the V_1 in the overall variance grid. Figure (3.5) show a plot of the $\frac{\partial P_3}{\partial P_1}$ term while figure (3.6) shows a plot of the $\frac{\partial P_3}{\partial P_2}$ term.

Note that the values are the greatest when P_1 and P_2 are the most dissimilar. This is the desired shape; we want the greatest variance to occur when P_1 and P_2 are dissimilar and we want small variance when P_1 and P_2 are similar. Also note the instability when P_1 is near 0 and P_2 is near 1 or vice versa; a small change in either of the probability values can have a big effect on the result of the combination formula. Cheng and Kashyap [7] use the sensitivity defined in equation (3.4) to measure this property.

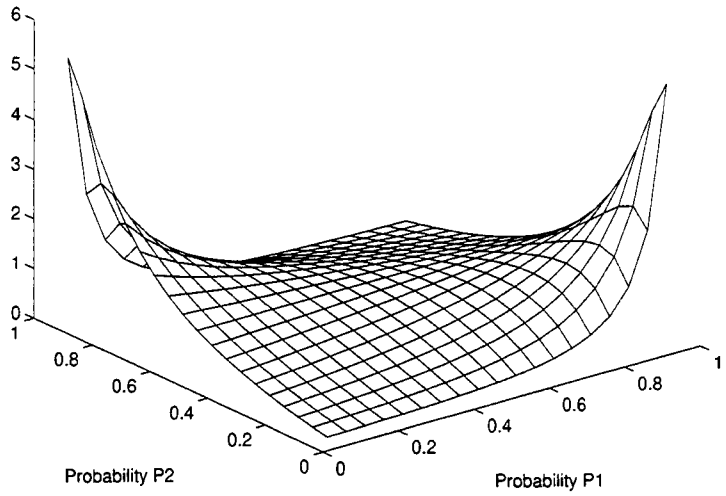


Figure 3.5: *Derivative of combination formula with respect to P1.*

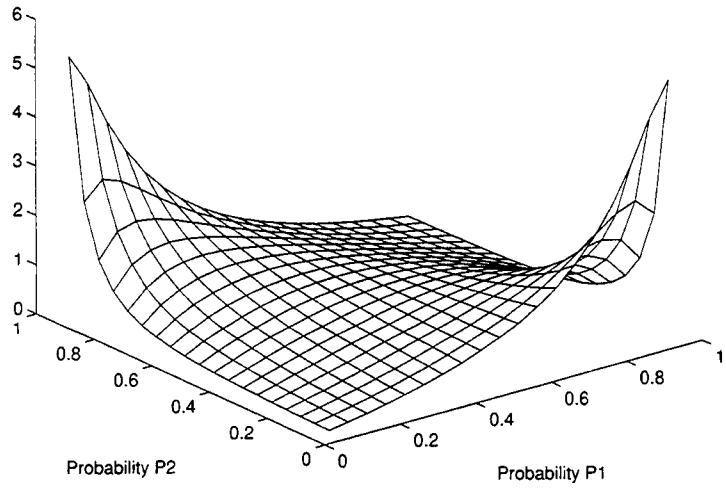


Figure 3.6: *Derivative of combination formula with respect to P2.*

$$S = \sup \left| \frac{\partial f(a,b)}{\partial a} \right| \tag{3.4}$$

The function, $f(a,b)$, is a binary fusion operator that combines the values a and b. Cheng and Kashyap [7] show that all associative, commutative, continuous, monotonic, and symmetric binary operators defined on a finite interval, such as equation (3.2), have $S = \infty$.

Taking the derivatives required by (3.3) leads to

$$V_3 = -\frac{-P_2V_1 + P_2^2V_1 - P_1V_2 + P_1^2V_2}{(2P_1P_2 + 1 - P_2 - P_1)^2}. \tag{3.5}$$

Figure (3.7) shows four plots of this function for various values of P_1 and P_2. Figure (3.7d) shows an important property of the variance combination formula. When $P_1 = 1/2$ and $P_2 = 1/2$, any variance, V, combined with itself yields $2V$. This is undesirable. When the probability of $1/2$ is combined with itself, the result is $1/2$. We would like the variance estimate of the probability of $1/2$ to behave similarly. This is easily accomplished by scaling the variance combination formula by $1/2$. Equation (3.6) shows the final variance combination formula.

$$V_3 = -\frac{-P_2V_1 + P_2^2V_1 - P_1V_2 + P_1^2V_2}{2(2P_1P_2 + 1 - P_2 - P_1)^2} \tag{3.6}$$

Initially the variance grid has to be initialized to some value. We have chosen the value of $1/12$ since that is the variance of the prior. The prior is the information that we have about the initial grid. Other issues relating to the combination formula will be discussed in the " Confidence Map Properties" section.

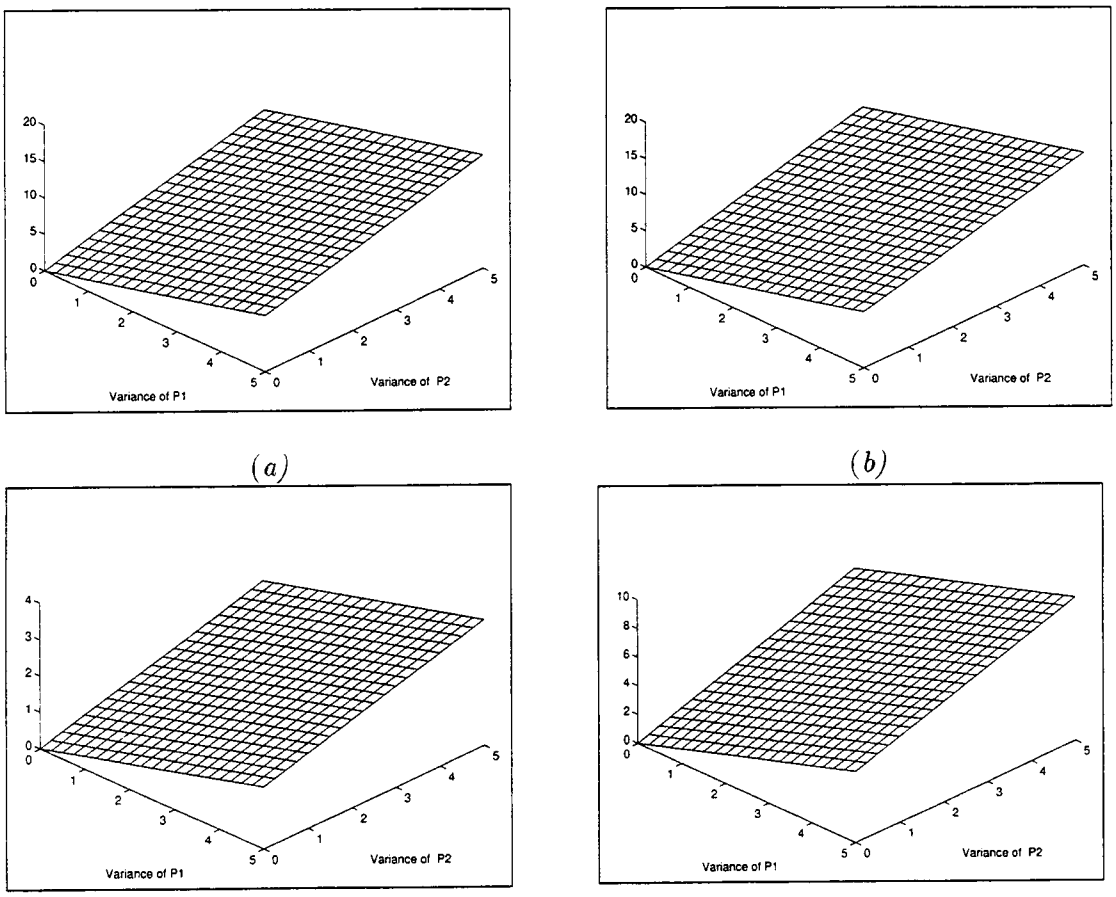


Figure 3.7: *Four plots of equation (3.5) with various values of* P_1 *and* P_2. *(a)* $P_1 = .2; P_2 = .8$. *(b)* $P_1 = .8; P_2 = .2$. *(c)* $P_1 = .8; P_2 = .8$. *(d)* $P_1 = .5; P_2 = .5$.

3.2.2.2 Variance to Confidence

The previous two sections have discussed variance values. The goal of this research is a "Confidence Metric." This section explains how to convert the variance values we have found previously into confidence values. Intuitively, we would like the confidence values to have the following properties:

1. Regions in which we have little or inaccurate sensor information should have a low confidence.
2. Regions in which we have much sensor information should have a high confidence.
3. Confidence values should range from zero to one.

The variance values that we have created have these properties:

1. Regions in which we have little or inaccurate sensor information have a high variance.
2. Regions in which we have much sensor information have a low variance.
3. The variance values range from zero to infinity.

The variance value is opposite of what we want for items 1 and 2. The range of variance values needs to be compressed to meet specification 3. One function that has the desired properties is:

$$C = \exp\left(-\frac{\sigma^2}{\alpha^2}\right) \tag{3.7}$$

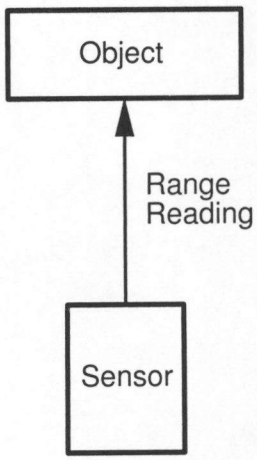
where C is the confidence value, σ^2 is the value from the variance grid, and α is a scaling factor. We select α to set the confidence scale at an intuitive level. Initially we have

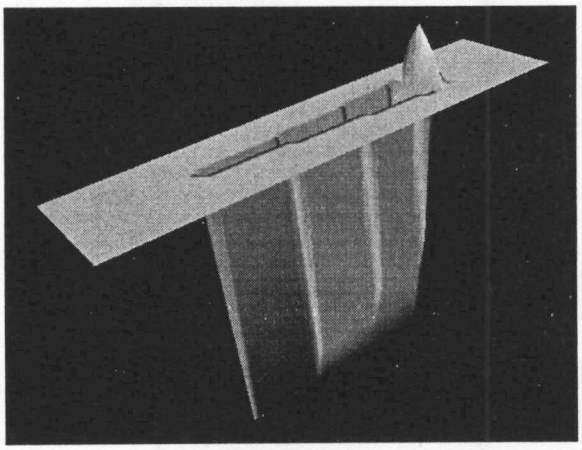
no sensor information and would like to have low confidence. The variance grid in this situation is set to 1/12. Therefore we select α to give a low confidence to the value of 1/12. We opted to have $C = .1$ when $V = 1/12$. This led to $\alpha = .19$.

3.2.3 Confidence Map Properties

This section will show some of the properties of the confidence metric that we have developed through the use of two dimensional plots. We start by considering the situation shown in figure (3.8a). A single sensor reading is taken of an object a fixed distance away. Figure (3.8b) shows the shape of the resulting occupancy grid. The flat region is at the unknown value of 1/2. The peak, which has a maximum value near .6, represents the most likely estimate of the object's location. The triangular region of low probability, which we will refer to as the *trough* or *free space wedge*, is composed of voxels through which the laser sensor had to pass to reflect from the object. For this example we set the standard deviation in the range direction and the standard deviation in the angle direction both to 1 cell width. The height was scaled by 100 to emphasize the shape.

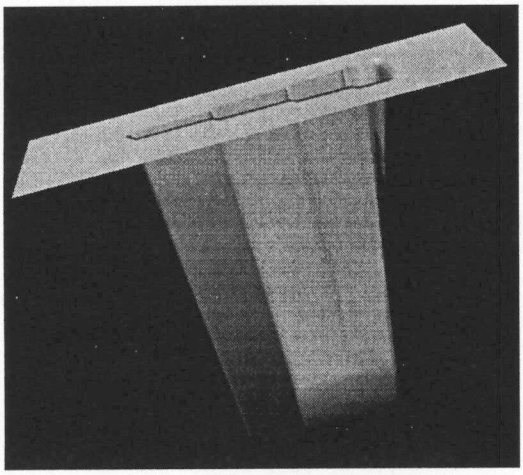
To obtain the variance caused by a single sensor reading, equation (3.1) is evaluated and equation (3.6) is used to combine it into the overall variance grid. For the following examples the overall variance grid was initialized to 1/12, which is the variance of the prior. Figures (3.9a), (3.9b) and (3.9c) show the individual variances due to the uncertainty in the prior, the range variance and the angle variance, respectively. This is the two dimensional version; hence there is only one angle variance to be concerned with. The variance in the angle direction is much smaller than the variances due to the range and prior. Figure (3.9c) was scaled by a factor of 10^{15} more than the other graphs to make the heights noticeable. This clearly shows that the variance in the angular standard deviation has little effect on the overall variance. The same is true in the 3-D case. Neither of the angles contribute much to the overall variance.



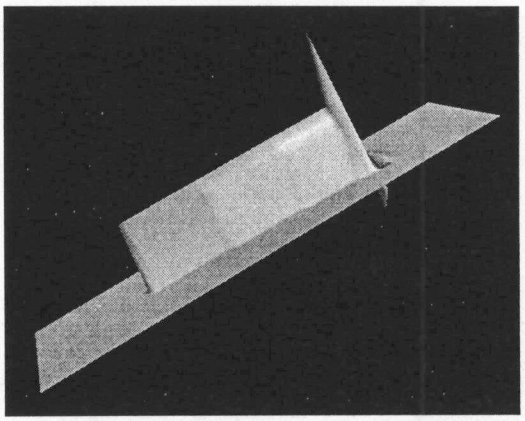


(a) (b)

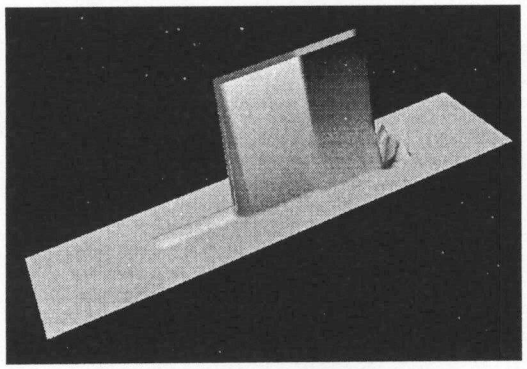
Figure 3.8: *A 2-D occupancy grid example. (a) The setup for a simple 2-D occupancy grid example. We simulate a single sensor reading taken of an object a fixed distance away. (b) The occupancy grid resulting from a single sensor reading.*



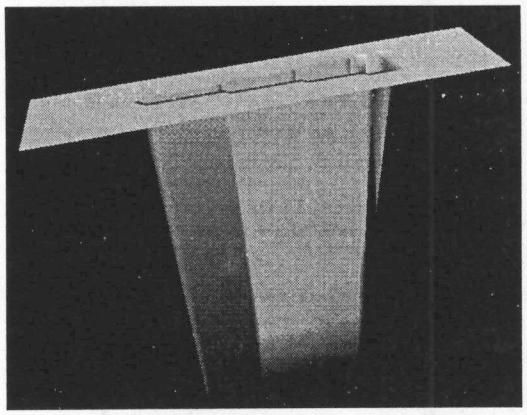
(a)



(b)



(c)



(d)

Figure 3.9: *Variances caused by uncertainties in the occupancy grid formulation. (a) The variance due to uncertainty in the prior. (b) Variance due to uncertainty in the range standard deviation. (c) Variance due to uncertainty in the angular standard deviation. Scaled by* 10^{15} *more than the other graphs. (d) The result of summing the prior, range and angular uncertainties.*

Figure (3.9d) shows the result of combining all the individual variances with the Δb, $\Delta\sigma(r)$, and $\Delta\sigma(\theta)$ all set to 1. Note that since the angular variance is small compared to the range and prior variance, the angular variance has little effect on the overall combination.

Recall that we previously stated that Δb, $\Delta\sigma(r)$, and $\Delta\sigma(\theta)$ are initial estimates of the variance of the parameters b, $\sigma(r)$, and $\sigma(\theta)$ respectively. These can be set to reasonable values. The value of Δb is set to $1/12$. This is because b can vary from zero to one; $1/12$ is the variance of a uniform distribution in this range. The choices of $\sigma(r)$ and $\sigma(\theta)$ reflect the accuracy at which the variance of the sensor is known. We have set them at $.2\sigma(r)$ and $.2\sigma(\theta)$. This was done with the assumption that the values of these parameters are well characterized by the sensor model. Figure (3.10) shows the result of the combination using these values. This plot was scaled by 1000.

Comparing the occupancy grid, figure (3.8), and and the variance map, figure (3.10), gives an intuitive feel for the meaning of the variance on the probability. The sharp downward curve of variance map corresponds to the bump on the occupancy grid. This

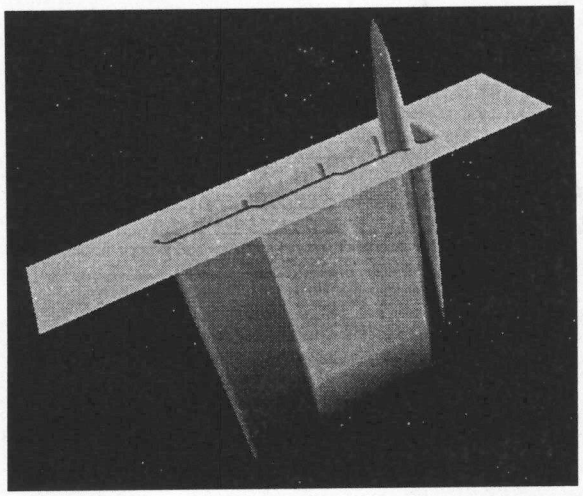


Figure 3.10: *The overall variance map. It results from combining all the individual variances with the* $\Delta b = 1/12$, $\Delta\sigma(r) = .2\sigma(r)$, $\Delta\sigma(\theta) = .2\sigma(\theta)$

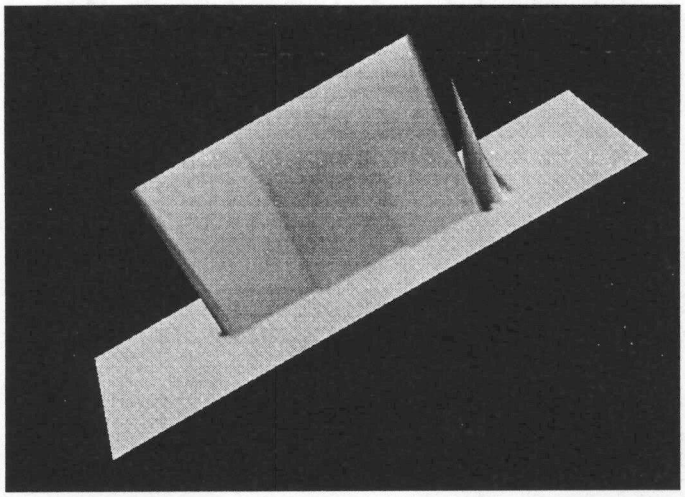


Figure 3.11: *Result of using the variance to confidence equation on the combined variance waveform.*

represents a low variance for the region where the sensor reading was returned. The hump on the variance map is related to the upward sloping region on the occupancy grid. This is caused primarily by the range variance uncertainty. Changing the range variance causes the width of the mound on the occupancy grid to vary. The transition region from low to high probability is most affected by this. The trough in the variance grid corresponds to the trough in the occupancy grid. Regions far from the location of the sensor reading are not affected much by the change in the prior assumption or the range standard deviation. It might seem that a change in the angular standard deviation should affect the trough region near the edges. Figure (3.9c) shows this somewhat, but remember the scale of this plot is 10^{15} times greater than the others.

After the variance map is derived, it has to be converted to a confidence value by using equation (3.7). Figure (3.11) shows the result of using this equation on the overall variance grid.

Note that the result has high confidence in regions with low variance and low confidence in regions that had high variance; this satisfies the requirements that we mentioned

previously. There is a high confidence in the free space wedge and at the location of the sensor reading. The flat region outside the sensor wedge is the constant that we selected for the unknown regions. The lowest confidence region is in the area where the occupancy grid transitions from low probability to high probability. The confidence in this region is slightly below the unknown value, although this is difficult to see in the figure.

Figure (3.12) shows how the confidence changes when multiple consistent sensor readings are obtained. These plots show the occupancy grid with the confidence values color coded onto it. Red is low confidence while magenta represents a high confidence. Figure (3.12a) shows a single sensor reading and its confidence values. Figure (3.12b) shows the result if another identical sensor reading is obtained. Figures (3.12c) and (3.12d) show how the occupancy grid and confidences are affected by 5 and 10 identical readings. Note on (3.12a) that the transition region is reddish which indicates low confidence. The trough region, color coded blue, has the highest confidence. As more sensor readings are acquired, the trough region becomes maximally confidence and is colored magenta. The peak of the probability hump becomes higher and more confident with additional sensor readings.

Another important property is related to the variance combination formula (3.6) and the super-Bayesian combination formula (3.2). Note that in the super-Bayesian combination formula, if $P_2 = 1/2$, then $P_3 = P_1$. This means that any voxel that has a value of $1/2$ cannot affect the values of other voxels in the combination process. This seem intuitively correct, since $1/2$ corresponds to the unknown condition and we would not expect voxels that we have no information about to affect those that we do.

As can be seen by examining the variance combination formula, there is no V_2 that will result in $V_3 = V_1$. This means that "unknown" region in the overall variance grid will affect the shape of individual sensor readings combined into these regions. Also the "unknown" regions surrounding the individual sensor readings will affect the values of cells throughout the overall variance grid. This means that the confidence combination formula

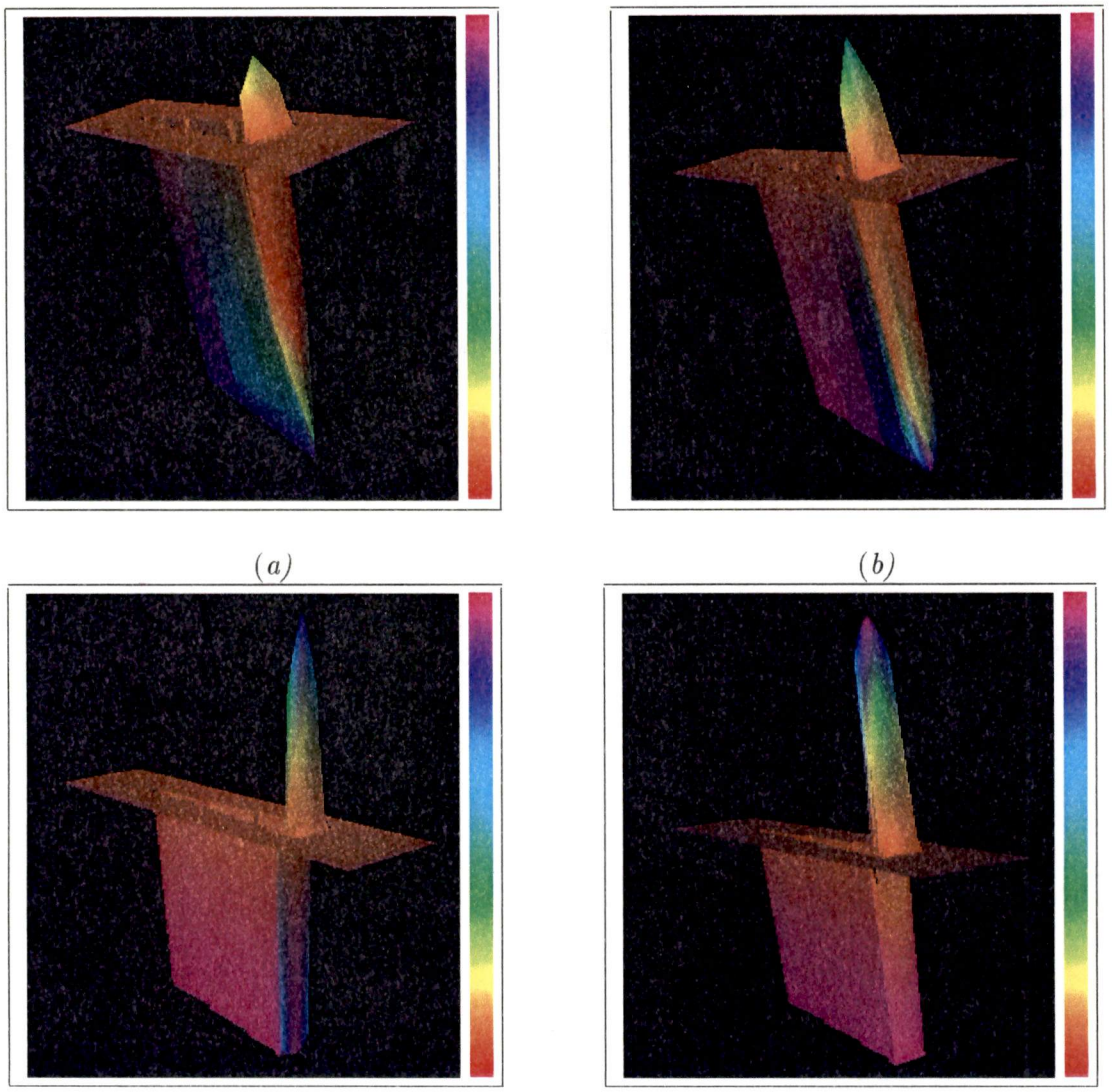


Figure 3.12: *The result of multiple consistent sensor readings in a 2-D grid. Occupancy grids with the confidence color coded onto it. Magenta represents a confidence of 1 and red represents a confidence of 0.(a) Result for a single sensor reading. (b) Result when two identical readings are obtained. (c) Result when five identical readings are acquired. (d) Result when ten identical readings are obtained.*

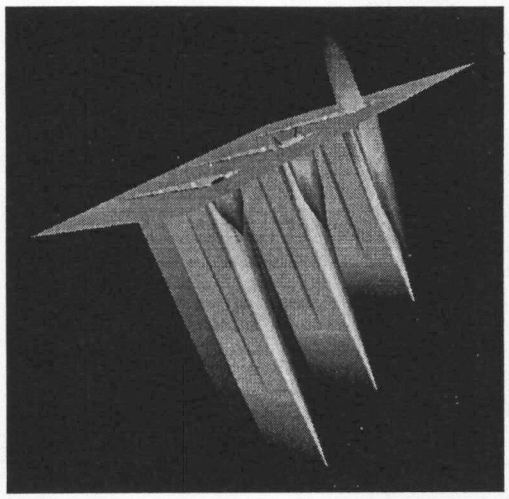


Figure 3.13: *The confidence map created by three nonoverlapping sensor readings. Shows how the sensor readings in different locations affect each other.*

is not associative. Figure (3.13) shows how the sensor readings in different locations affect each other. The readings were taken from bottom to top. Note that the variance in the free space wedge in decreased for each successive reading. The high variance mound in the transition regions is nearly eliminated by additional readings. The variance in the region where the sensed object is most likely to be is increased.

The fact that voxels in the "unknown" region can affect the values of voxels in sensed regions at first does not seem to be intuitively valid due to the reasoning mentioned above. However, we argue that the value of 1/2 in an occupancy grid voxel does provide some information, namely it represents the prior assumption about the percent of the workspace that is occupied. The variance of the 1/2 value is based on the variance of the prior distribution which we have assumed to be uniform. Since the prior assumption does provide some information about the space, it is valid to have the confidence in unsensed regions affect the confidence in sensed regions.

Even though this effect may be explainable, it may not always be desirable. One reason is that the value of 1/2 is not selected to accurately reflect the percentage of space

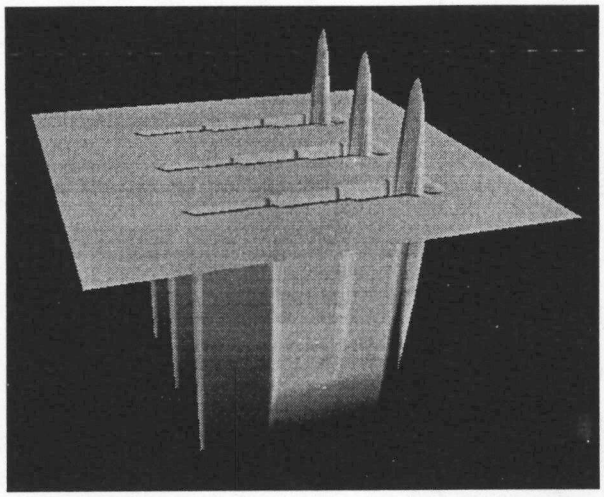


Figure 3.14: *The confidence map created by three nonoverlapping sensor readings when only updating the region where new sensor information is obtained.*

occupied; it is selected primarily because it simplifies the occupancy grid combination formula [26]. If the prior information is not accurate, it may be undesirable to have it affect the confidence in sensed regions where we have solid sensor information. One possible way to get around this effect is to update only the regions of the variance map where new sensor information is available. This is what we have done for the examples given in the rest of this paper. Figure (3.14) shows the result of this procedure for the same example used for figure (3.13). Note that the three sensed regions all look the same.

Figure (3.15) shows another important property of the occupancy grid method. The super-Bayesian combination formula favors values that are near 1 or 0. That is when a value near 0 is combined with any other value (as long as it is not close to 1) the result is near 0. This means that when the trough of probability from one sensor reading intersects the mound from another sensor reading, the trough dominates and the mound is wiped out. The importance of the this effect is shown in the next section. A similar effect occurs with the confidence values.

The last property that we will consider is a sampling problem that occurs when the

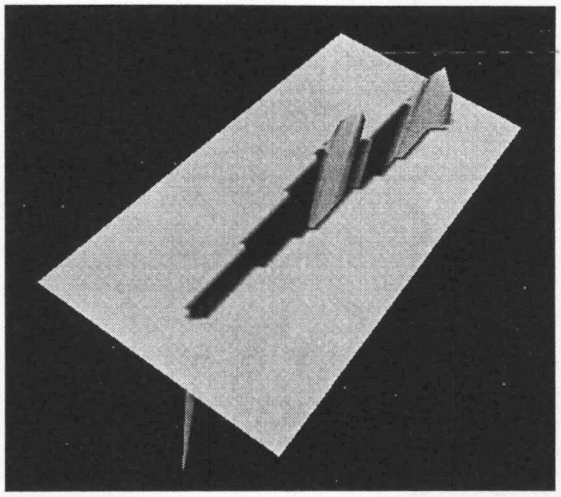


Figure 3.15: *Shows the interference effect when the trough of one sensor reading intersects the mound of another.*

waveform from a single sensor reading is translated and rotated to there correct position in the overall map. The grid size that we use for most of our experiments is less than 150×150 and the probability-of-occupancy waveform is typically less than 8 voxels wide. Translation and rotation on low resolution grid results in waveforms that are somewhat jagged. Figure (3.16) shows a waveform rotated every 30 degrees about a complete circle. The sampling problem is clearly apparent.

3.3 Occupancy Grid and Confidence Map Results

This section shows some results of applying the two dimensional occupancy grid method and the confidence metric to a simple simulated scene. The scene is shown in figure (3.17). It is a room that is 100 units by 100 units with a square block that is 30 units by 30 units in the center of it. Simulated sensor readings are taken from locations near each corner. Each sensor takes one reading every degree throughout a 90 degree arc.

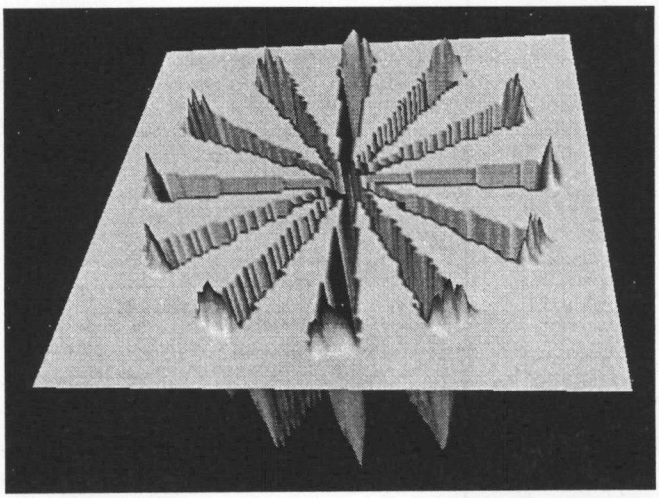


Figure 3.16: *A waveform rotated every* 30 *degrees about a complete circle. The sampling problem is clearly apparent.*

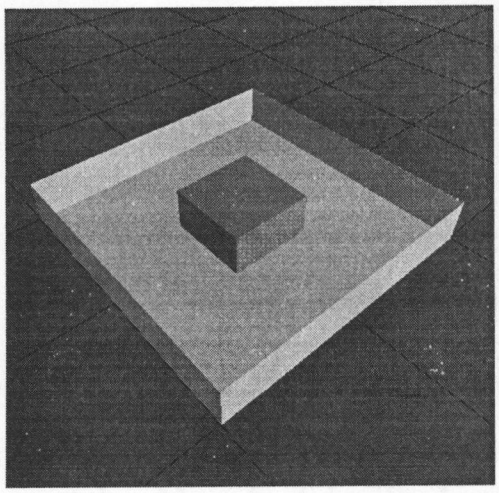


Figure 3.17: *The simulated scene used in the following examples.*

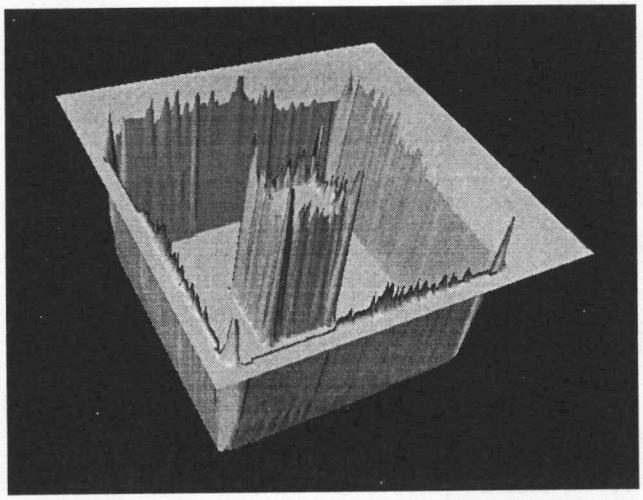
Figure (3.18a) shows the resulting occupancy grid when we assume a standard deviation of one unit in both the range and angular measurement. Figure (3.18b) shows the resulting confidence map from the top while figure (3.18c) shows the bottom view.

Figure (3.19) shows the occupancy grid with the confidence values color coded onto it.

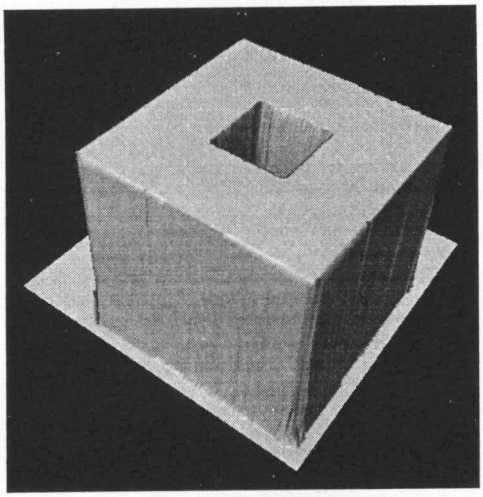
There are a couple of interesting items regarding the occupancy grid. The large number of sensor readings reduces the probability of occupancy in the empty region around the box to nearly zero. The unsensed regions outside the room and inside the box remain at a value of 1/2. The probability near surface locations is greater than 1/2. These are the expected and desired properties of the occupancy grid method. We also note a couple of undesirable properties. The probability is jagged along the surfaces. The probability along certain sections of the wall is nearly 1/2. This second effect is easy to explain. Sensor readings were simulated parallel to the wall (one unit away). The trough region of these readings is wide enough that it wipes out the higher probability values along the wall. This interference effect was discussed in the previous section. We eliminated the sensor readings taken less than three degrees from a wall and ran the data again. Figure (3.20) shows the result. Note that the regions along the walls are much more uniform. The gaps in the corners occur because the corners were the regions mapped by the removed sensor readings.

The jaggedness along the surface of the box may be caused by a couple of factors. It may partially be caused by a lack of resolution. Recall from figure (3.16) that the rotation of the waveform causes a jagged looking result.

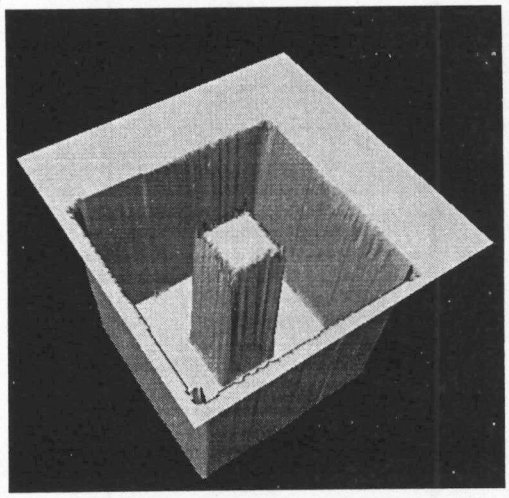
The interference effect may also be partially responsible for the jaggedness. Each successive reading of the box from any given sensor is taken along a sloping flat surface. Each sensor reading creates a probability region that is several voxels wide. When the readings are taken close together along a sloping surface, it is possible that the trough from one sensor reading could intersect the mound of another. This could cause a jagged



(a)



(b)



(c)

Figure 3.18: *Occupancy grid and confidence map with range and angular standard deviation set to 1. (a) The occupancy grid (b) Top view of the confidence map (c) Bottom view of the confidence map.*

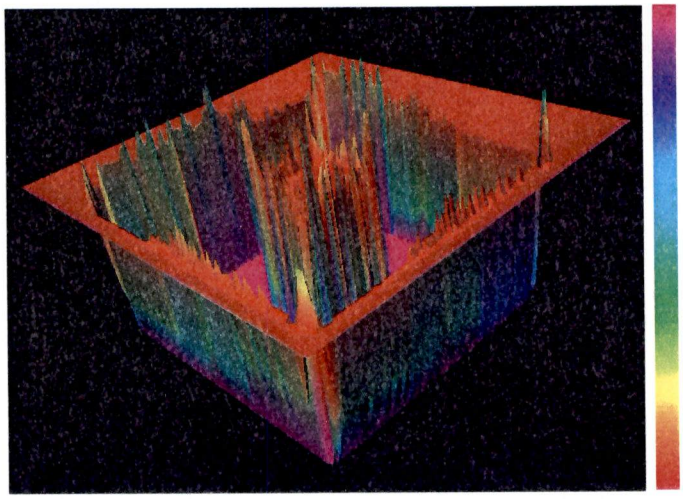


Figure 3.19: *Occupancy grid with the confidences color coded onto it. The confidences are scaled to fill the entire colorbar.*

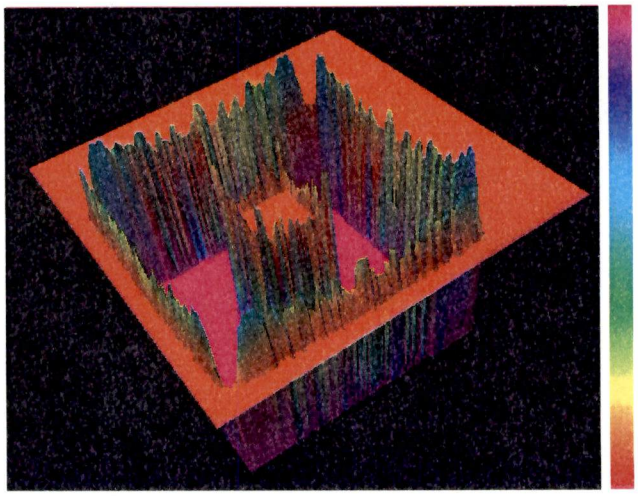


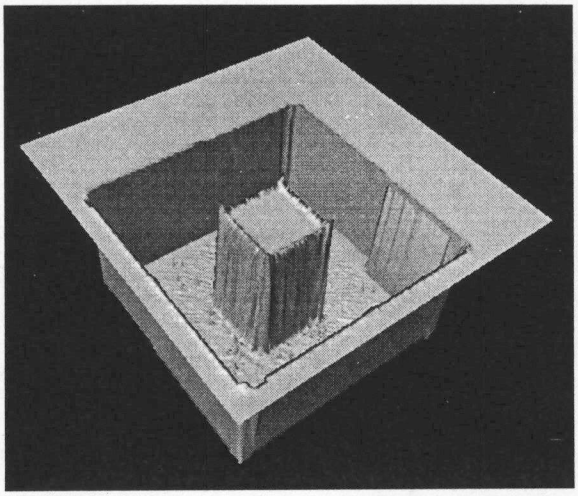
Figure 3.20: *The occupancy grid created for the box in room scene when all sensor readings taken less than 3 degrees from the wall are eliminated. The colors represent relative level of confidence.*

surface.

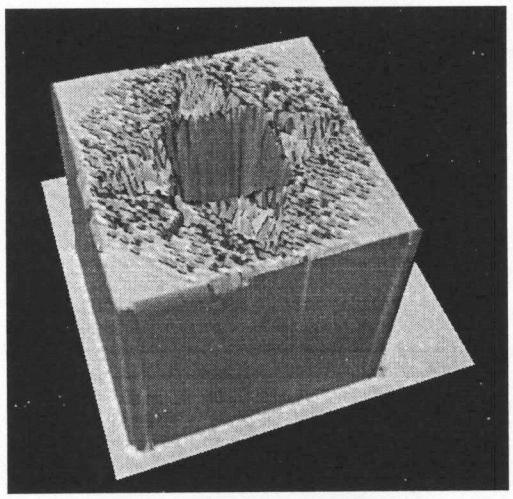
Now consider the confidence map shown in figure (3.18b) and figure (3.18c). The top view shows that the large number of sensor readings have caused the confidence in the empty region around the box to be increased to nearly 1. Regions outside the room and inside the box remain at the .1 value we selected to be the confidence in the prior. The bottom view shows the transition from high confidence to low confidence. Some jaggedness can be seen in the confidence map due to the same reasons that we discussed occupancy grid values. Note also that there are some spikes that have confidences lower than the confidence of the prior. This is caused by the characteristic shape of the confidence waveform for a single sensor reading shown in figure (3.11). The region that corresponds to the transition region of the occupancy grid, has a confidence lower than the confidence of the prior. However, in this example, the interference effect seems to have eliminated most of these values.

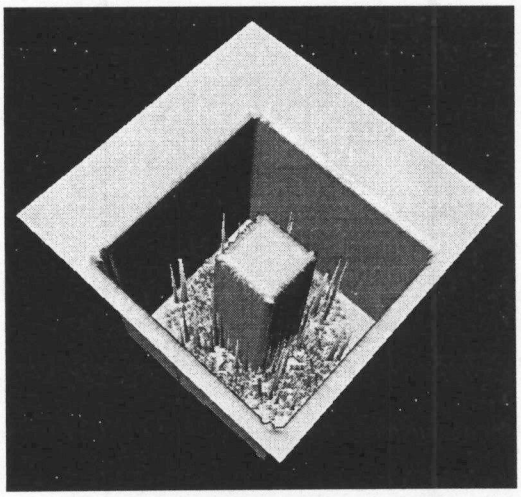
Figure (3.21a) shows the occupancy grid for the same simulated room except with the range standard deviation of the sensor set to 2 units and the angular standard deviation set to .25 units. Figures (3.21b) and (3.21c) show top and bottom views of the confidence map. Figure (3.22) shows the occupancy grid color coded with the confidence values.

The most noticeable change is that the confidence in the empty region around the box is no longer close to 1 everywhere; there are spikes throughout the region. This is because the reduction in the angular standard deviation has decreased the width of the free space wedge for a single sensor reading. This means there is less overlap among the readings. Note that the confidence is still smooth near the corners where there is still plenty of overlap. It is also interesting to note that the occupancy grid also contains some unevenness in this region, but it is less pronounced. This is apparently because it takes fewer readings to drop the probability of occupancy to near zero than it takes to increase the confidence to near one.



(a)





(b) (c)

Figure 3.21: *Occupancy grid and confidence map with range standard deviation set to 2 and the angular standard deviation set to 0.25. (a) The occupancy grid (b) Top view of the confidence map (c) Bottom view of the confidence map.*

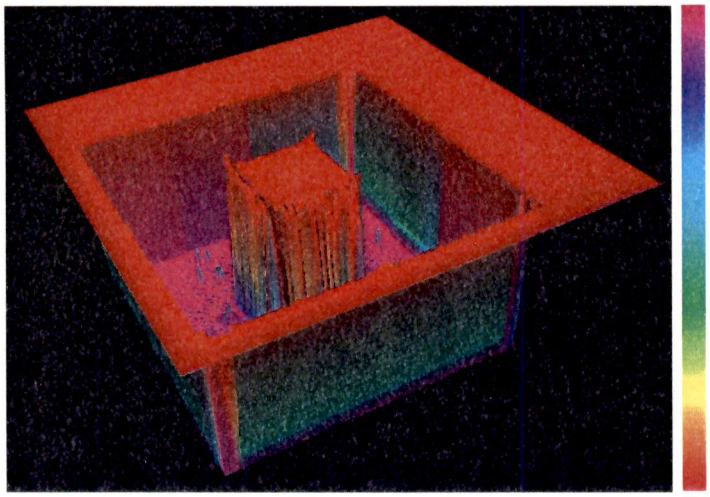


Figure 3.22: *A confidence colored occupancy grid. The range standard deviation of the sensor is set to 2 units and the angular standard deviation is set to .25 units. The colors represent relative level of confidence.*

The other noticeable change can be seen most clearly in figure (3.21a). The probability values along the surfaces are not as large as they were in the previous example. This because the larger range standard deviation is related to a less accurate range measurement and therefore we are less sure that cells near the sensor reading are occupied. It also appears that the surfaces are less jagged. This may simply be because the heights are smaller so it less noticeable. On the other hand, the smaller angular standard deviation should cause less interference between reading. The greater range standard deviation should move the troughs from individual sensor readings further from the surface so would not affect other readings as much.

3.4 Summary

This chapter has presented a review of the occupancy grid method as well as our derivation of a confidence metric to complement it. The occupancy grid method has the

advantage of being statistically based; hence, it can model laser range sensors with a probability density function. This allows it to explicitly handle uncertainty. However, our research has shown several aspects of this formulation that are undesirable for applications using laser range scanners. These drawbacks include interference between neighboring sensor readings and a large calculation time.

The confidence metric that we developed, based on the concept of higher-order uncertainty, shows promise for several robotic navigation tasks. It does a good job of highlighting free space regions which receive the greatest confidence. Regions that have not been explored can also be easily found; these areas have the characteristic *unknown* confidence value. These two types of information are very important for applications in which a robot must move in an unknown environment.

CHAPTER 4

Volumetric Modeling Technique

This chapter describes a new volumetric modeling technique that we have developed. The first section explains the reason we chose to create a new method. Section 2 gives the theory behind the method, including how we model range images, how we combine images from several viewpoints, and how we create the final 3-D model from our volume information. The final section discusses the confidence metric we use for this technique.

4.1 Motivation

The volumetric method that we have developed is intended to overcome several of the drawbacks that we noticed with our implementation of the occupancy grid formulation [46]. One of our biggest concerns is the *interference effect* that we discussed in the previous chapter.

Another disadvantage with the occupancy grid formulation is calculation time. A region around every point has to be computed for every point in the range image. The size of this region is determined by the estimated error variance and the voxel size of the grid. If the error variance is much greater than the size of a single voxel, a huge number of calculations will be needed for a reasonable size (256×256) range image.

Another difficulty with the occupancy grid formulation presents itself in extracting a surface for visualization. In the occupancy grid method, the best estimate of the surface location is where the probability is the greatest (see figure 4.1). This means that a ridge following technique would have to be used to extract a surface from the volumetric data.

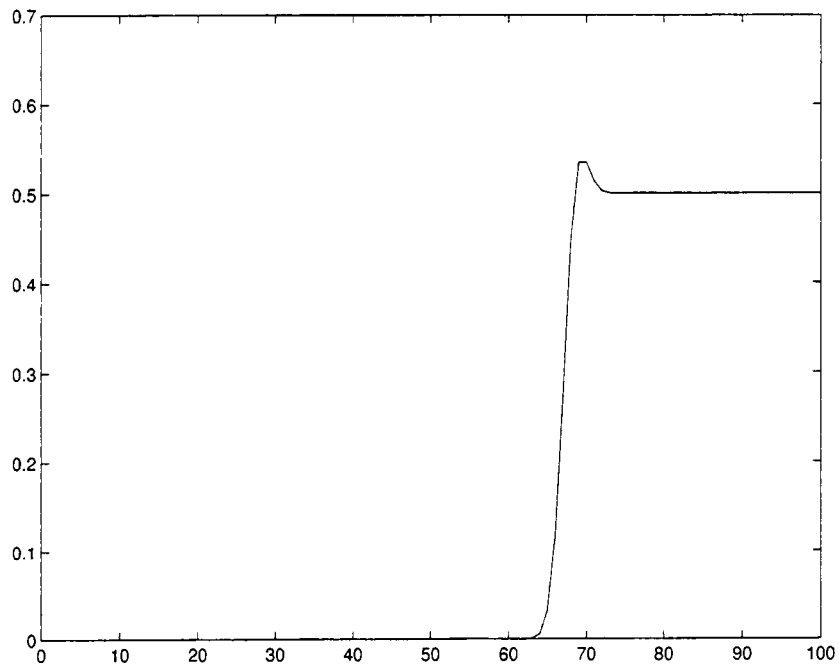


Figure 4.1: *The 1-D occupancy grid resulting from a single sensor reading of 70. Note that the best estimate of the surface, at the reading of 70, is where the probability is the greatest.*

Ridge following is not a trivial task.

4.2 Theory

We have several goals that we are attempting to achieve with our volumetric modeling technique:

- It should be robust enough to handle a reasonable amount of noise with some outliers.
- It should take advantage of redundant information between range images.
- The model should improve as progressively more noisy images are incorporated.
- It should avoid the interference effect mentioned above.
- It should be computationally bearable.
- A surface model should be straightforward to extract.

There are three logical steps to develop the volumetric modeling process. First we must decide how to model the range data for a single range image. Then we need to select a method to combine multiple range images into a single representation. Finally, we have to implement a surface extraction technique on the volumetric data. Naturally these steps are interrelated, so the entire process should be considered when developing each step.

4.2.1 Modeling the Range Image

Figure (4.2) shows a typical range image. The darker regions represent areas that are closer to the sensor and lighter areas correspond to regions further away. A range image can be visualized as a three-dimensional surface as shown in figure (4.3). This gives a good estimate of the true surface when there is no noise present in the range image. However,

Figure 4.2: *A simple range image of a mug. Darker regions correspond to near regions while lighter regions relate to farther regions.*

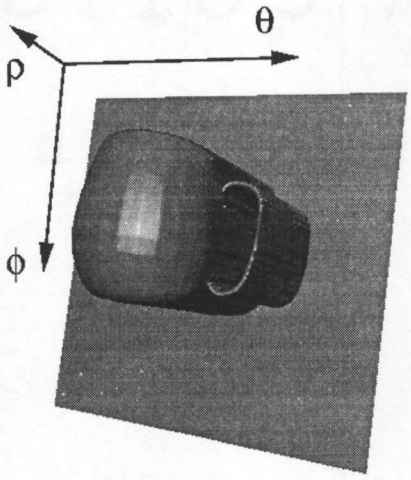


Figure 4.3: *A surface mesh of the range image shown in figure 4.2. This surface data contains no noise.*

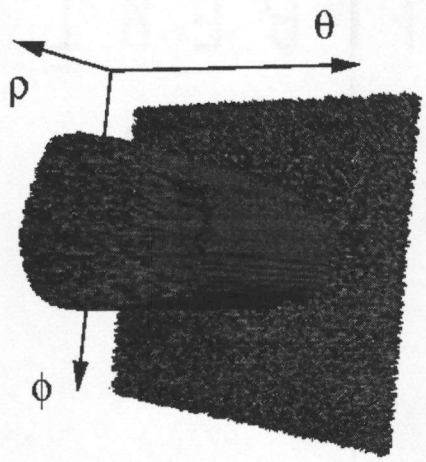


Figure 4.4: *A surface mesh of the range image shown in figure 4.2 with a small amount of noise added. The surface is clearly more jagged.*

as figure (4.4) shows, when the range image contains noise, the resulting surface plot is rather jagged.

When noise is present, the range image gives a representation of the surface that deviates from the true surface. If the characteristics of the noise are known, a probabilistic estimate of the deviation can be found. For example, consider the one-dimensional situation shown in figure (4.5). An object is located a distance of r from the sensor.

Suppose that the sensor gives readings that are corrupted with uniform noise of width 2ϵ. Then the probability density function (pdf) is as shown in figure (4.6). Where r is the reading obtained from the sensor and the true surface in somewhere in the interval $[r - \epsilon, r + \epsilon]$. Taking the integral of the pdf gives the distribution function shown in figure (4.7).

The height of the distribution function gives the probability that the measured surface is between x (along the x-axis) and the sensor (located at $x = 0$).

This shape can be seen intuitively by starting at 0 and moving along the positive x-axis. The maximum error is ϵ; therefore there is no chance of encountering the surface until the point $x = r - \epsilon$. The noise is uniform; hence the chances of having passed the

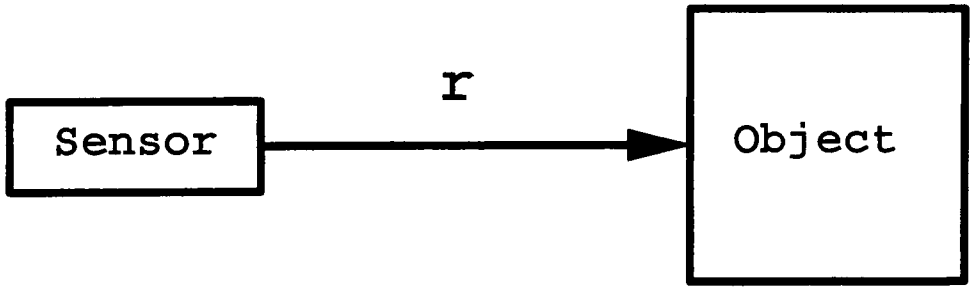


Figure 4.5: *A sensor is located at a distance r from an object.*

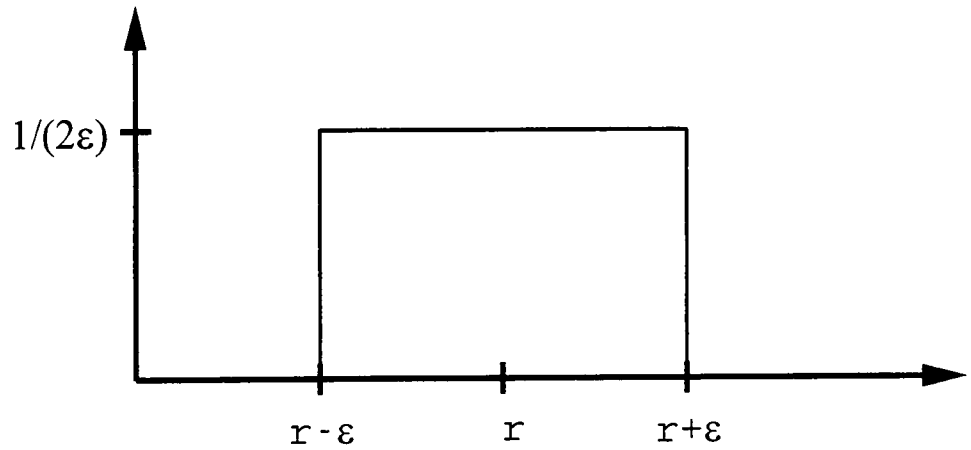


Figure 4.6: *The density function arising from uniform noise with a width of* 2ϵ.

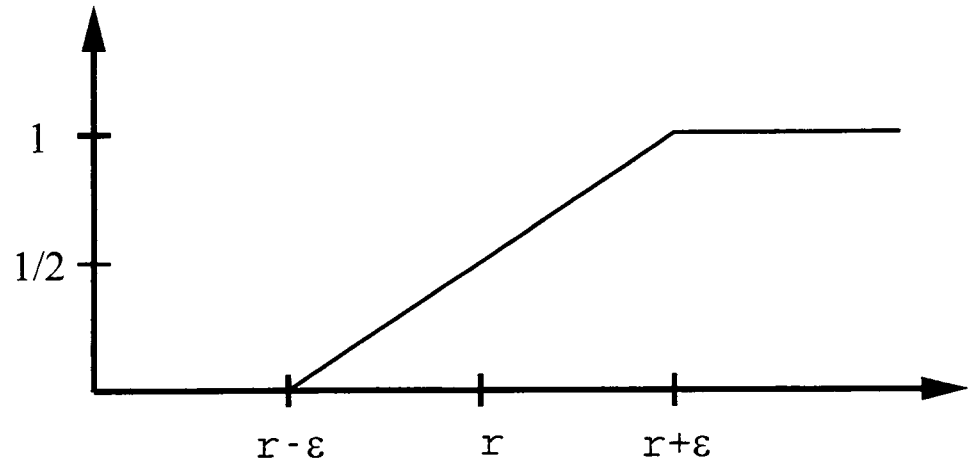


Figure 4.7: *The distribution function arising from uniform noise with a width of* 2ϵ.

true surface increases linearly with x until the point $x = r + \epsilon$ is reached. This is beyond the maximum error; therefore it is certain that the surface has already been encountered.

We find it convenient to reinterpret the distribution function as the probability that x is occluded from the sensor. This is equivalent because if the surface is located at x, everything to the right of x must be occluded.

There are two additional items for which we wish to account, outliers and the possibility of other surfaces. Let us consider outliers first. A range measurement with outliers may have the shape shown in figure (4.8a). There is a small chance of getting readings in the range $[r - \delta,\ r + \delta]$ because of outliers. The distribution function will have the a shape as in figure (4.8b). This example allows for outliers in a small region $[r - \delta,\ r + \delta]$. If we keep the relative number of outliers constant and increase the width of the region in which they may occur, we get the density function shown in figure (4.8c). The corresponding distribution function is shown in figure (4.8d). Figures (4.8e) and (4.8f) show the density and distribution functions resulting from increasing the width of the outlier region even further. Notice that the slope of the distribution function corresponding to the outliers is decreased as the width of the region $[r - \delta,\ r + \delta]$ is increased.

In the limit, as the width of the outlier region approaches infinity, the slope of the distribution function in these regions approaches zero. This is shown in figure (4.9). Intuitively, this is simply saying that we are never completely sure which areas are occluded because of the outliers.

The last consideration is additional surfaces. Figure (4.9) is a good *certainty-of-occlusion* graph for a single sensor from a single location. However, in our situation we will want to take multiple sensor readings from many locations. Figure (4.10) shows the situation.

Sensor 'a' measures the box from the left. Everything from point 'A' to the right is occluded from it. This does not mean that this entire region is occluded from all locations.

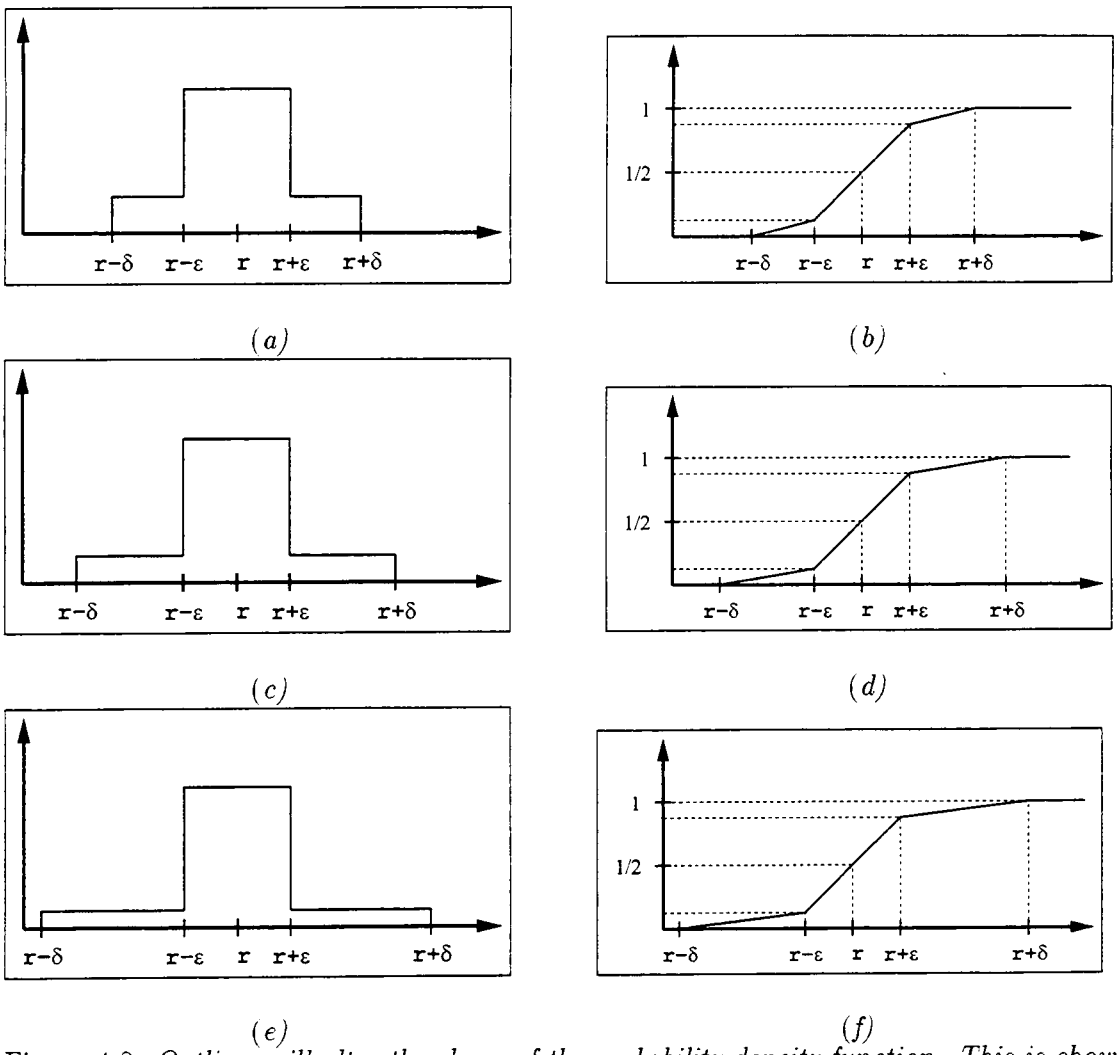


Figure 4.8: *Outliers will alter the shape of the probability density function. This is shown in (a): there is a small chance of getting readings in the range $[r-\delta,\ r+\delta]$ because of outliers. The distribution function will have the a shape as in (b). This example allows for outliers in a small region $[r-\delta,\ r+\delta]$. If we keep the relative number of outliers constant and increase the width of the region in which they may occur, we get the density function shown in (c). The corresponding distribution function is shown in (d). Figures (e) and (f) show the density and distribution functions resulting from increasing the width of the outlier region even further. Notice that the slope of the distribution function corresponding to the outliers is decreased as the width of the region $[r-\delta,\ r+\delta]$ is increased.*

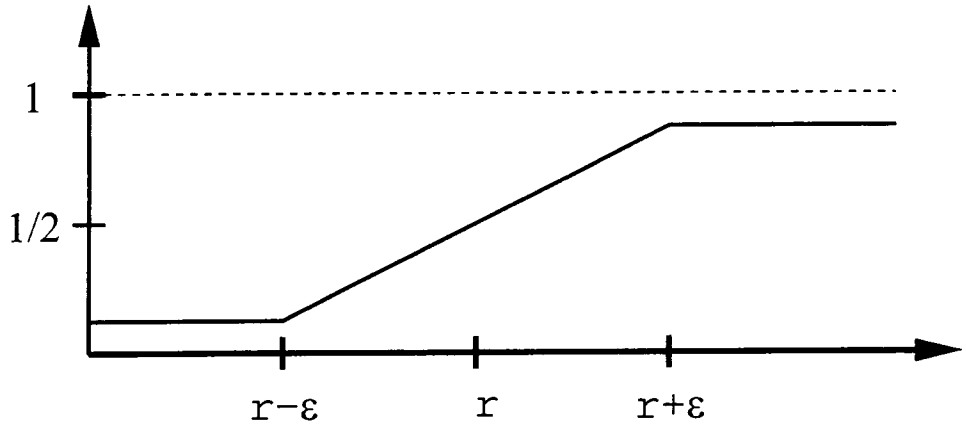


Figure 4.9: *This is the distribution function that results from letting the region in which outliers may occur to* $\pm\infty$.

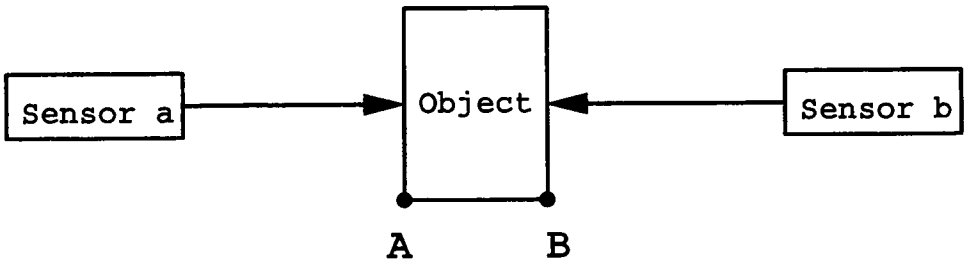


Figure 4.10: *A diagram that shows that a region may be occluded from one sensor but not another. Sensor 'a' measures the box from the left. Everything from point 'A' to the right is occluded from it. This does not mean that this entire region is occluded from all locations. The region to the right of 'B' is not occluded from sensor 'b.'*

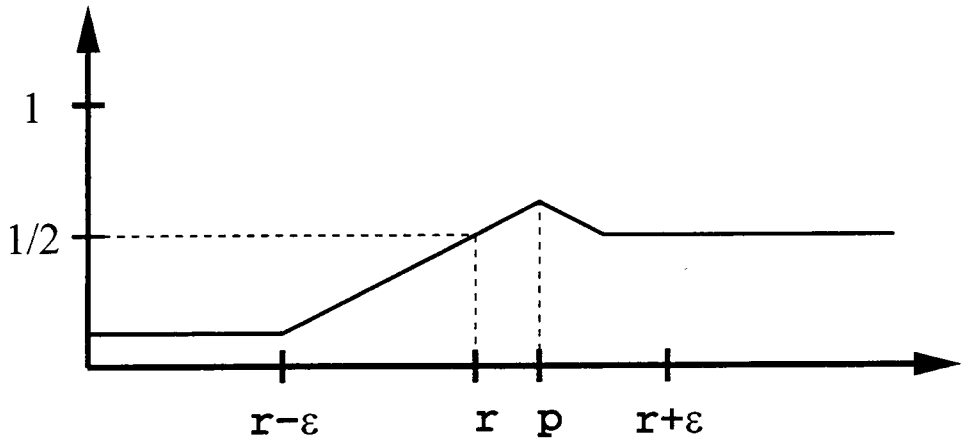


Figure 4.11: *An altered distribution function that takes into account the possibility of other surfaces that may be seen from other directions. The slope of the line segments is determined by the noise level. The location of the peak value p determines the distance between surfaces that can be distinguished.*

The region to the right of 'B' is not occluded from sensor 'b.' Thus we have adjusted our 'certainty-of-occlusion' graph as shown in figure (4.11).

The graph drops down to near 1/2, the unknown value, behind the sensor reading location to account for the possibility of other surfaces. The slope of the line segments is determined by the noise level. Note that the peak value p is behind the position r where the range reading was received. This is because the graph represents the certainty that a voxel is inside the sensed object. Noise in the range scanner means that the true surface location may not be exactly at r. This means that our labeling of voxels (near r) as inside or outside the object is greatly affected by noise; hence these voxels have values near 1/2 to indicate our uncertainty. Voxels further behind the sensor reading have greater certainty because these voxels are less likely to be affected by noise. However, as we discussed earlier, we need to take into account additional surfaces; therefore we reduce the certainty after point p linearly to a value slightly above 1/2. Keeping the certainty

function slightly above 1/2 behind r ensures that the entire interior of the final model is above 1/2. This allows our technique to create a closed surface model.

Note that this shape is similar to the shape that Elfes [11, 47] derived for the one-dimensional occupancy grid (see figure 4.1). There are two important differences. Ours does not drop completely to zero in front of the sensor; this is to account for outliers. In our graph the most likely surface lies at a value of 1/2; this will make the surface extraction step easier. Another difference is that ours is a piecewise linear function, while the occupancy grid function is continuous. This is because we used a uniform distribution in our derivation instead of a Gaussian pdf. We used a uniform distribution for a couple of reasons. It makes the derivation of the certainty function easier to understand. The piecewise linear certainty function that results from a uniform distribution is faster to calculate than the erf-based certainty function that would result from using a Gaussian pdf. A Gaussian pdf, however, would probability more accurately model the true error characteristics of a rangefinder; the derivation done in this section could easily be applied to a Gaussian pdf if the results using a uniform pdf are not accurate enough.

The graph in figure (4.11) is the general shape of the certainty function that we use in 1-D. We extend this to 3-D by taking each vertex in the volumetric grid and projecting a line back to the sensor location, through the range image. We find the point where the line intersects the range image, and bilinearly interpolate to find a range value. We then calculate the distance from the sensor to the considered voxel and determine how close it is to the measured surface. We use the 1-D certainty function to assign a certainty value to the voxel based on its distance from the surface along the projected line. Figure (4.12) shows this graphically.

This process involves several steps. First we have to convert the voxel vertices from world coordinates to the coordinates of the range camera. We do this by multiplying the augmented coordinate vector,

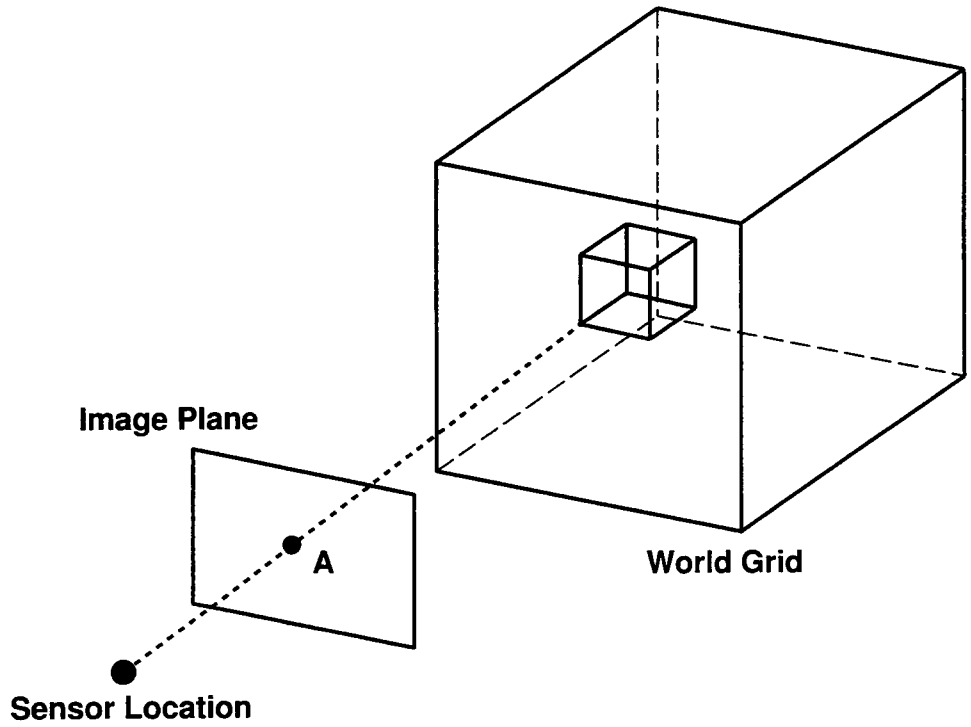


Figure 4.12: *This figure shows how we project voxel vertices onto the range image. We construct a line between the sensor and the voxel; this represents the scanner's line-of-sight. The intersection point, A, is where the line-of-sight intersects the range image.*

$$\mathbf{W} = [W_x \quad W_y \quad W_z \quad 1]^T, \tag{4.1}$$

with a transformation matrix to get the camera coordinates.

$$\mathbf{L} = \mathbf{H}^{-1} \times \mathbf{W} \tag{4.2}$$

Where $\mathbf{L}$ is the vertex point written in the camera coordinate system. We find the inverse transformation matrix, $\mathbf{H}^{-1}$, as described in [51]. The homogeneous transformation matrix, $\mathbf{H}$, which transforms from camera coordinates to world coordinates, is defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{P}_{1\times3} & \mathbf{S}_{1\times1} \end{bmatrix}. \tag{4.3}$$

Where $\mathbf{R}_{3\times3}$ is a 3×3 rotation matrix that describes how the axes of the camera coordinate systems are oriented with respect to the world coordinate system. The position vector, $\mathbf{T}_{3\times1}$, is the position of the origin of the camera coordinate system, written in world coordinates. The perspective transform, $\mathbf{P}_{1\times3}$, is not used in our application; we set it to $\mathbf{0}$. We set the scaling factor, $\mathbf{S}_{1\times1}$, to 1. Hence, we can write the homogeneous transformation matrix as

$$\mathbf{H} = \left[\begin{array}{ccc|c} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{4.4}$$

The vector,

$$\mathbf{o} = [o_x \quad o_y \quad o_z]^T, \tag{4.5}$$

is the y-axis of the camera coordinate system. For our application this is the direction in which the ranger scanner is pointed. The vector,

$$\mathbf{a} = [a_x \quad a_y \quad a_z]^T, \tag{4.6}$$

is the z-axis of the camera coordinate system. We define this as the up direction of the laser scanner. The $\mathbf{o}$ and $\mathbf{a}$ vectors can be obtained from information about the range camera. The remaining vector, $\mathbf{n}$, is obtained using the vector cross product

$$\mathbf{n} = [n_x \quad n_y \quad n_z]^T = \mathbf{o} \times \mathbf{a}. \tag{4.7}$$

This ensures that the camera coordinate system is orthogonal. The inverse of $\mathbf{H}$, which will transform from world coordinates to camera coordinates, is easily found using the formula

$$\mathbf{H^{-1}} = \begin{bmatrix} n_x & n_y & n_z & -(\mathbf{p} \cdot \mathbf{n}) \\ o_x & o_y & o_z & -(\mathbf{o} \cdot \mathbf{n}) \\ a_x & a_y & a_z & -(\mathbf{a} \cdot \mathbf{n}) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.8}$$

Applying equation (4.2) gives the location of the voxel in camera Cartesian coordinates. The geometry of laser range scanners typically do not allow them to create range images in Cartesian coordinates [52]. Hence to determine the relationship between voxel points and pixels in the range image, we find it convenient to convert the camera coordinates from Cartesian to the coordinate system used by the laser scanner. This is done using the transformation function

$$\mathbf{C} = \mathbf{g}(\mathbf{L}). \tag{4.9}$$

Where $\mathbf{g}$ represents a function that transforms from Cartesian coordinates, $\mathbf{L}$, to the laser scanner's native coordinates, $\mathbf{C}$. The geometry of lasers scanners typically cause them to create images in a coordinate system that is roughly spherical. Besl [53] describes several of these. We give the transformation function, $\mathbf{g}(\cdot)$, and its inverse, $\mathbf{g}^{-1}(\cdot)$, for several cameras that we will use in examples later in this thesis.

The first is the orthogonal-axis system that we used when we took our synthetic range images.

$$\mathbf{g}(\cdot) = \begin{cases} C_r &= \sqrt{x^2+y^2+z^2} \\ C_\theta &= \tan^{-1}(x/z) \\ C_\phi &= \tan^{-1}(y/z) \end{cases} \tag{4.10}$$

$$\mathbf{g}^{-1}(\cdot) = \begin{cases} C_x &= \frac{r\tan\theta}{\sqrt{1+\tan^2\theta+\tan^2\phi}} \\ C_y &= \frac{r\tan\phi}{\sqrt{1+\tan^2\theta+\tan^2\phi}} \\ C_z &= \frac{r}{\sqrt{1+\tan^2\theta+\tan^2\phi}} \end{cases} \tag{4.11}$$

The Perceptron laser scanner uses a modified spherical system.

$$\mathbf{g}(\cdot) = \begin{cases} r &= \sqrt{x^2+z^2} \\ C_\rho &= \sqrt{x^2+y^2+z^2} \\ C_\theta &= \tan^{-1}(-x/z) \\ C_\phi &= \tan^{-1}(y/r) \end{cases} \tag{4.12}$$

$$\mathbf{g}^{-1}(\cdot) = \begin{cases} C_x &= -\rho\sin\theta\cos\phi \\ C_y &= \rho\sin\phi \\ C_z &= \rho\cos\theta\cos\phi \end{cases} \tag{4.13}$$

The Coleman laser scanner uses another variation of the spherical coordinate system.

$$\mathbf{g}(\cdot) = \begin{cases} r & = & \sqrt{y^2 + z^2} \\ C_\rho & = & \sqrt{x^2 + y^2 + z^2} \\ C_\theta & = & \tan^{-1}(x/r) \\ C_\phi & = & \tan^{-1}(y/z) \end{cases} \tag{4.14}$$

$$\mathbf{g}^{-1}(\cdot) = \begin{cases} C_x & = & \rho \sin\theta \\ C_y & = & \rho \cos\theta \cos\phi \\ C_z & = & \rho \cos\theta \sin\phi \end{cases} \tag{4.15}$$

Now we are ready to find the point, A, on the range image shown in figure (4.12). We assume that the pixels in the x-axis and y-axis of the range image correspond to equally spaced values of θ and ϕ. Information about the scanner allows us to determine the true θ and ϕ value for a reference pixel in the range image. We usually chose the center pixel as the reference and assume the scanner is pointed directly toward this pixel. This means the center pixel has θ and ϕ both equal to zero. Once the reference pixel is determined, the θ and ϕ values for the remaining pixel can be found using the field-of-view of the scanner and the number of pixels in the x and y directions of the image to find the fixed θ and ϕ increments. Now we can compare the θ and ϕ components of $\mathbf{C}$ to the θ and ϕ values of the pixels to ascertain where in range image the voxel projects. We represent this projection as the function $\mathbf{h}(\cdot)$. Hence, the point, $\mathbf{A}$, can be found as

$$\mathbf{A} = \mathbf{h}(\mathbf{C}). \tag{4.16}$$

Where $\mathbf{A} = [A_x \quad A_y]^T$ is a 2-dimensional location on the range image. Once we have $\mathbf{A}$, we need to find the associated range value in the range image. Since the range image is discretized sampling of the scene, $\mathbf{A}$ will usually be in an area between pixels in the

range image. We use bilinear interpolation to estimate the range value at $\mathbf{A}$. This can be expressed as

$$r = b(\mathbf{A}). \tag{4.17}$$

Where the function $b(\cdot)$ represents bilinear interpolation and r is the returned range value.

The final step is to compare the distance between the camera and voxel to the value of r; this is the distance from the voxel to the surface created by the range image. This will allow us to properly place the 1-D certainty function along the laser line-of-sight and determine the certainty function for the voxel.

The distance between the camera and the voxel location is

$$d = ||\mathbf{W} - \mathbf{T}||_2. \tag{4.18}$$

Recall that $\mathbf{T}$ is location of the camera in world coordinates, $\mathbf{W}$ is the voxel location in world coordinates, and d is the distance between the two. The function, $||\cdot||_2$, is the vector 2-norm.

Finally, we can use the difference, $d - r$, to find the certainty value, v, for the voxel. Figure (4.13) shows this graphically for the case when $d - r$ is negative. If $d - r$ is positive, we would find v as the certainty value that is $|d - r|$ to the right of r. Functionally this can be represented as

$$v = \text{cert}\,(d - r). \tag{4.19}$$

Where v is the certainty value for the voxel and cert $(\cdot)$ is the function that estimates it from d and r. Hence, the process of finding a certainty value for a single voxel represented in world Cartesian coordinates is simply

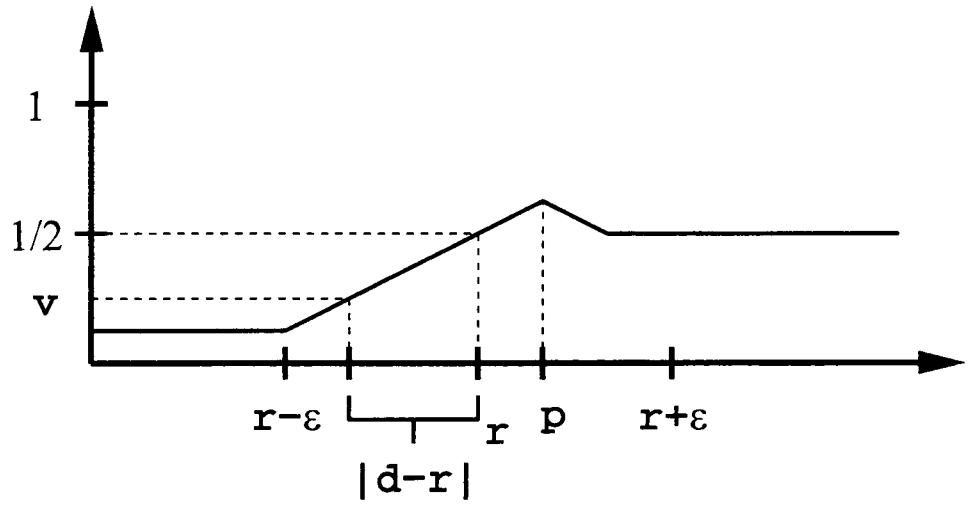


Figure 4.13: *How we find the certainty value for a voxel from the value 'd.' The number $d - r$ is the distance from r that we have to evaluate the certainty function. In this case, $d - r$ is negative; hence we find the certainty value at a distance of $d - r$ to the left of r. If $d - r$ were positive, we would evaluate the certainty function to the right of r.*

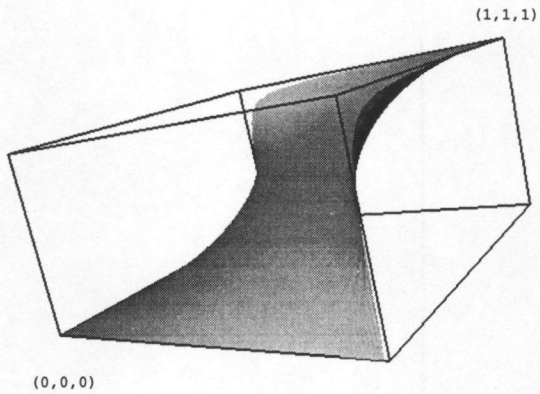


Figure 4.14: *This is a plot of the super-Bayesian operator. Note that any value combined with* 1 *is* 1 *and any value combined with* 0 *is* 0.

$$v_{ijk} = \text{cert}\,(d - b(\mathbf{h}(\mathbf{g}(\mathbf{H}^{-1}\mathbf{W}_{ijk})))) \tag{4.20}$$

Where the subscript ijk is the x, y, z index of the voxel in the world grid.

4.2.2 Fusing Multiple Range Images

The next step is selecting a fusion operator to combine the individual range image representation into an overall volumetric grid. We selected the super-Bayesian combination formula,

$$f(a,b) = \frac{ab}{ab + (1-a)(1-b)}, \tag{4.21}$$

where the a and the b are the certainty values to be fused. Figure (4.14) shows a plot of this function.

The super-Bayesian combination formula is a *binary operator*; hence the certainty

values from the individual range reading must be combined recursively. Let n be the number of range images to be combined and the values i, j, k represent the index of the voxel in the range image. Then the overall certainty for each voxel is

$$V_{ijk} = f(f(\cdots f(f(v_{ijk_1}, v_{ijk_2}), v_{ijk_3}), \cdots), v_{ijk_n}). \tag{4.22}$$

Where $f(\cdot)$ is the super-Bayesian combination formula and the v_{ijk} terms are found from equation (4.20). Substituting equation (4.21) recursively into equation (4.22) yields

$$V_{ijk} = \frac{\prod_{m=1}^{n} v_{ijk_m}}{\prod_{m=1}^{n} v_{ijk_m} + \prod_{m=1}^{n} (1 - v_{ijk_m})}. \tag{4.23}$$

We selected this operator because it has many of the properties that we wanted. Since it is *commutative and associative*, we can combine certainties in any order. This is important because there is no clear reason to favor a certainty from one range image over another; this would make it difficult to develop rules to determine the order of combination.

The super-Bayesian is *monotonic*. This means that

$$f(a, b) \leq f(a, c) \qquad \text{if} \qquad b \leq c. \tag{4.24}$$

This is an intuitive type of ordering. We expect the result of the fusion to be greater if we increase one the certainty values.

Symmetry is another useful property that the super-Bayesian function has. This is the property that

$$1 - f(a, b) = f((1 - a), (1 - b)). \tag{4.25}$$

Technically, symmetry means that evidence that supports a hypothesis refutes its negation and the degree of support for a hypothesis determines the degree that it refutes the negation of the hypothesis [54]. Symmetry is desirable when trying to decide between

two mutually exclusive hypotheses. For our application, symmetry means that if we have a certainty of a that a voxel is inside an object, we have a certainty of $1 - a$ that it is outside. This make sense because we have only two options, the voxel is inside or it is outside.

The super-Bayesian function also has a convenient *identity* at the value of 1/2. This means that any value, a, combined with 1/2 results in the original value, a. In this situation, 1/2 can be used as the "no information" value, meaning that we are completely unsure if the voxel is occluded or not. This relates well to placing a surface at 1/2; it is uncertain whether the surface is inside or outside an object.

Another advantage of the probabilistic-like method of representation and fusion that we use is that it only requires one value to be stored for each voxel. Other methods such as the Dempster-Shafer method and forms of fuzzy reasoning represent values as an interval instead of as a single value. This has some advantages, such a being able to explicitly represent uncertainty, however it requires that two numbers be stored per voxel.

4.2.3 Extracting a Surface

After all the range images have been combined into a single volumetric grid, a surface must be extracted to visualize the data. We use a variation of the popular marching cubes algorithm that was originally developed by Cline and Lorensen [55] to visualize medical data.

The super-Bayesian combination formula that we use for the fusion step tends to push the voxel values toward 1 or 0. This leads to large quantization errors if we simply interpolate linearly between voxel vertices to find the surface location as is done in the standard marching cubes algorithm. Therefore we subsample the vertex edges that we know pass through the surface and use the original range data to find the certainty values at each new subsample. We use the additional information to find a more accurate estimate

of where the surface crosses the edge than we could by a mere linear interpolation.

4.3 Confidence Metric

This section describes a couple of confidence metrics that we have implemented for our volumetric approach.

4.3.1 Number of Sensor Readings

The first method simply counts the number of times that each voxel has been included as a possible surface location. More specifically, it looks at the certainty function for each range image and and adds one to each voxel that is in a non-zero sloped region. The non-zero sloped regions are near the sensor reading; voxels in this region are most likely to contain the surface that the sensor measured. Although this method is simple, it is effective at showing which regions have not been sensed and which regions have been measured often enough that additional readings would have little effect.

4.3.2 Slope-Based Confidence

The confidence metric based simply on the number of times a voxel is measured is useful in some situations; however, it assumes that each range image contains the same amount of noise. This may not always be the case. The amount of noise in a range image can be effected by factors, such as electronic's temperature and illumination, that may change from one scan to the next.

The slope-based confidence metric allows us to weight the confidence obtained from scans differently. We refer to it as the slope-based confidence because we use the slope of the certainty function as a basis for the confidence. Recall, that the shape of the certainty function was determined by an estimate of the probability density function of the noise

for a scan. We have used an assumption of uniform noise; hence the certainty function is linear,

$$\text{cert} = mx + b, \tag{4.26}$$

in the region corresponding to random noise. The slope, m, is determined by the width of the probability density function. This means that scans with greater amounts of noise will have a certainty function with a more gradual slope. Hence, this slope is a natural choice for a confidence measure, since scans with less noise will automatically have a greater slope.

We have chosen to obtain an overall confidence, for each voxel, by adding the slopes of the certainty functions that affect the voxel.

$$\text{con}_{ijk} = \sum_{s=1}^{n} m_s \tag{4.27}$$

Where con_{ijk} is the confidence of the voxel at location (i, j, k) in the world grid. The value, n, is the number of range images to be combined and m_s is the slope of certainty function corresponding to the sth range image.

Note that the certainty function has a slope of zero far from the sensor reading. This means that the confidence for voxels in these regions will not be affected by this certainty function. This is desirable because these regions give little information about the true surface location.

For visualization purposes, the results of this method may need to be scaled to emphasize the relative differences between confidences in various regions. Note that if all the range images are assumed to have the same noise level, the slope-based confidence method can be scaled to be equivalent to the confidence measure based on the number of sensor readings.

4.3.3 Slope-Normal Confidence

The slope-normal confidence builds upon the slope-based confidence by incorporating normal information. The premise is that the sensor gives better information about regions that are nearly perpendicular to its line of sight. Hence, for each voxel, we calculate the line-of-sight vector from the voxel to the range scanner,

$$\mathbf{l} = \mathbf{W} - \mathbf{T}. \tag{4.28}$$

Where $\mathbf{W}$ is the location of the voxel in world coordinates, $\mathbf{T}$ is the location of the camera in world coordinates, and $\mathbf{l}$ is the camera line-of-sight vector to the voxel.

Then we determine the location,A, in the range image to which the voxel maps as described by equation (4.16). The position, A, will most likely map to a region between pixels as shown in figure (4.15). We take the neighboring pixels (x_1, x_2, x_3) and convert them to Cartesian coordinates using the $g^{-1}(\cdot)$ function described in equations (4.11, 4.13, 4.15). We use these 3-D coordinates to calculate the vectors

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{x}_2 - \mathbf{x}_1 \\ \mathbf{v}_2 &= \mathbf{x}_3 - \mathbf{x}_1. \end{aligned}$$

This allows us to estimate the normal in this region as the cross product of $\mathbf{v}_1$ and $\mathbf{v}_2$,

$$\mathbf{N} = \mathbf{v}_1 \times \mathbf{v}_2. \tag{4.29}$$

Once we have the normal vector and the line-of-sight vector, we calculate the cosine of the angle between them.

$$\cos\alpha = \frac{\mathbf{N} \cdot \mathbf{l}}{\|\mathbf{N}\|_2 \|\mathbf{l}\|_2} \tag{4.30}$$

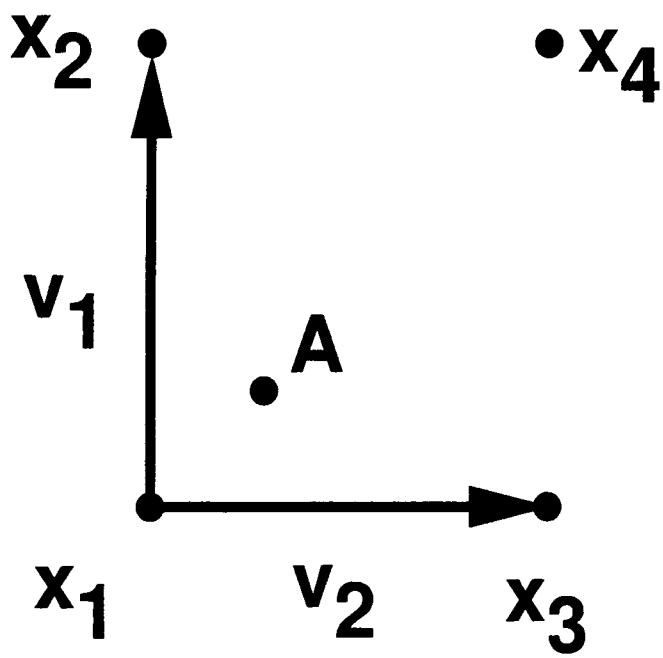


Figure 4.15: *A voxel vertex maps to point 'A.' This is typically between pixels in the range image. The range values at these pixels can be used to estimate the vectors v_1 and v_2. The cross product of these vectors will provide an estimate of the surface normal at 'A.'*

We use this to weight the slope value we obtain from the certainty function,

$$\mathrm{con}_{ijk} = \sum_{s=1}^{n} m_s \cos\alpha_s. \tag{4.31}$$

4.4 Summary

This chapter has presented our derivation of a new volumetric modeling technique. It is statistically based, as is the occupancy grid method; hence it can explicitly model uncertainty using the error model of the sensor. Our method has been designed especially for laser time-of-flight scanners and we will show in the next chapter that it overcomes several of the drawbacks that we noticed with our implementation of the occupancy grid method. This chapter also shows a couple of confidence metrics for our new volumetric technique. We will demonstrate the situations in which these techniques are useful in the next chapter.

CHAPTER 5

Evaluation and Discussion

This chapter shows how our volumetric technique performs. We start by showing results using synthetic data. This is a useful test because we do not have to worry about registration, camera geometry, or noise; we can focus entirely on the technique itself. However, we must show that the technique is valid on data taken from a real range finder as well. This chapter presents results using data from two types of laser range finders. First, we demonstrate how our method reconstructs objects from images taken with a Perceptron model P5000 scanner. Then we use range images taken with a Coleman Coherent Laser Radar scanner to reconstruct a scene. We indicate the advantages and disadvantages of the method throughout the chapter.

5.1 Reconstruction of Synthetic Models

This section presents some of the results that we obtained using the volumetric modeling technique on synthetic images with simulated noise. The general steps are:

1. Acquire registered range images.
2. Initialize 3-D grid to 'unknown' value of 1/2.
3. For each range image:
   (a) Apply a 5×5 mean filter to the foreground.
   (b) Create the certainty grid.
   (c) Combine into overall volumetric grid.

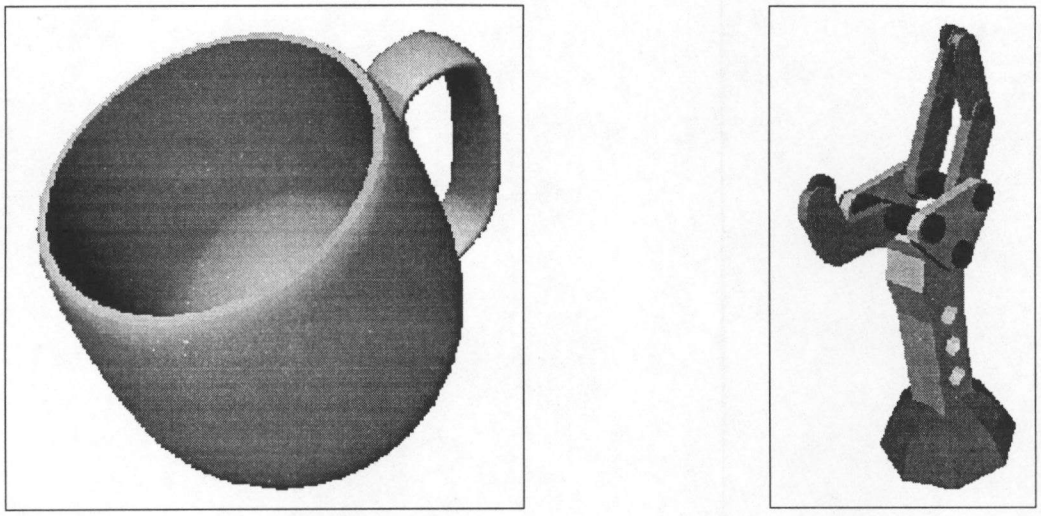
4. Extract a surface at the certainty value of 1/2.

Step (3a) may require further elaboration. We know that the object we are imaging is well in front of the background; therefore we threshold the image to obtain the foreground, containing the object, and the background. Then we apply a 5×5 mean filter to the foreground. Separating the object from the background before filtering prevents the edges of the object from be blurred away. We omitted this step when working with the noiseless data.

Step (3a) assumes that the foreground and background can be easily separated. This is a reasonable assumption in certain controlled situations. One of these is object modeling tasks where the object is placed in a known environment. In other applications, such as modeling an entire scene, segmenting the foreground from the background can be quite complex. In these cases, step (3a) may be eliminated or replaced with a more general, edge-preserving filtering technique such as that of Perona and Malik [56].

We also need to make a comment about step (3b). We notice that artifacts may occur in the object model due to the discrete nature of the range images. Range images are composed of pixels that cause a stairstep effect around the edges of objects. Our volumetric technique will attempt to reconstruct this in the object model. To avoid this, we ignore voxels that map to object edges in the range image. Range images from other views will determine the values for these voxels. As with step (3a), this process may have to be modified for certain applications.

Finally, we need to make a comment about incremental updating. Ideally, one would like to have a method where it is possible to combine several views, look at the model and then add additional views to get an updated version without having to restart the program. This is desirable, however, it means that the entire 3-D grid must be held in computer memory throughout the process; a large 3-D array of floating point numbers



(a) (b)

Figure 5.1: *The ideal mug and gripper models.*

can rapidly consume computer memory. Therefore, we have sacrificed the incremental updating ability in our program to be able to work with larger volumes. We assume that we have all the views initially which allows us to work with the 3-D grid two layers at a time, saving significant computer memory.

Consider the ideal models of a mug and robotic gripper pictured in figure (5.1). We acquired twelve synthetic range images of each shown in figures (5.2) and (5.3). The reconstructions, done on a 80×80×80 grid, are shown in figure (5.4).

The next several examples will show how our method performs with the addition of noise. For this experiment we added uniform noise that had an average value that was 13% of the typical object distance. The reconstruction of the 12 images is shown in figure (5.5a,b) and (5.6a,b). The addition of noise has clearly made the objects rougher, however, they are still recognizable.

Next we wish to show that the model improves with additional range images. Figure (5.5c,d) and (5.6c,d) show the reconstruction from cycling through the 12 synthetic images ten times to simulate 120 range images. Uniform noise with the same characteristics as

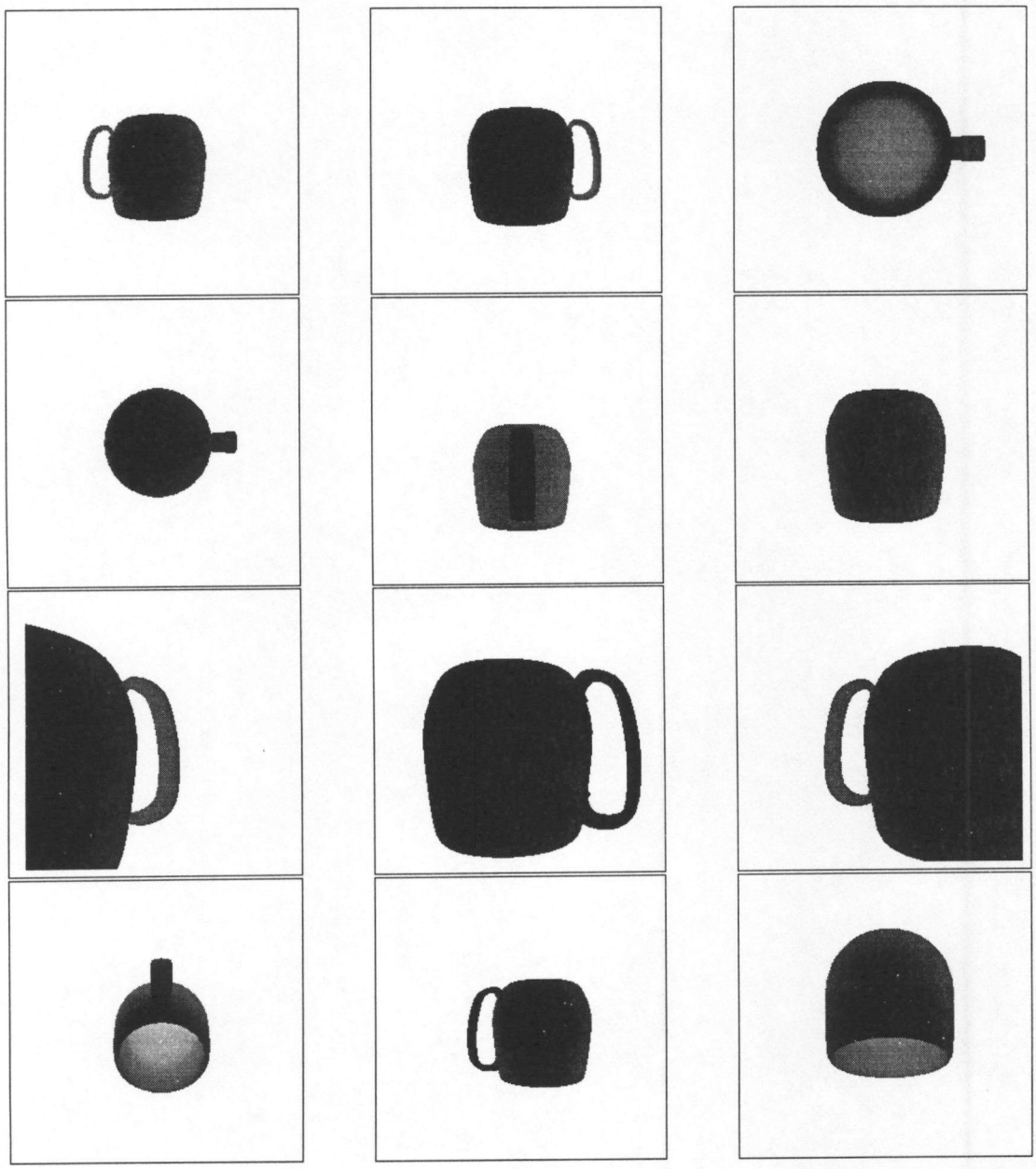


Figure 5.2: *Twelve synthetic range image of the ideal mug model.*

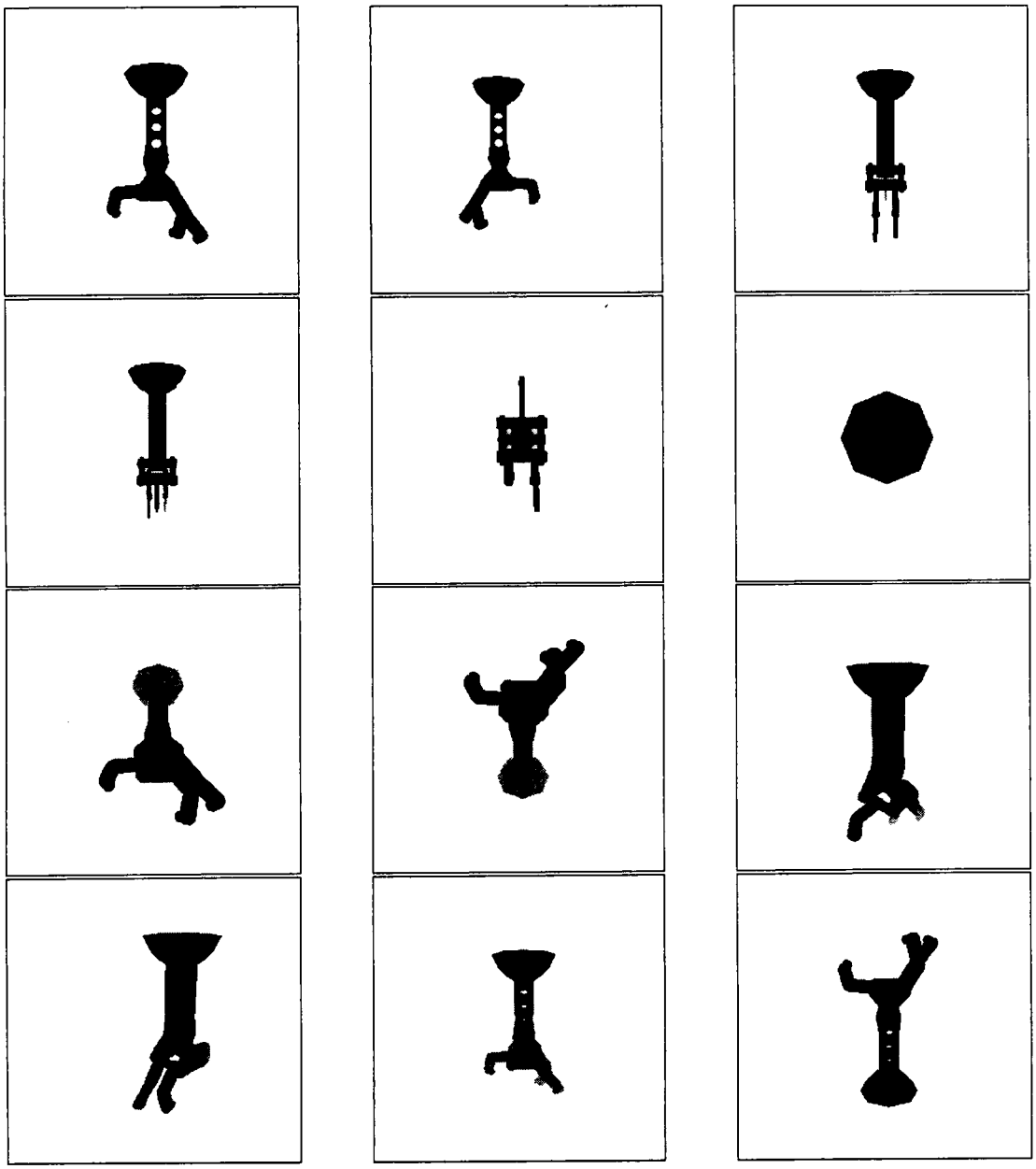


Figure 5.3: *Twelve synthetic range image of the ideal robotic gripper model.*

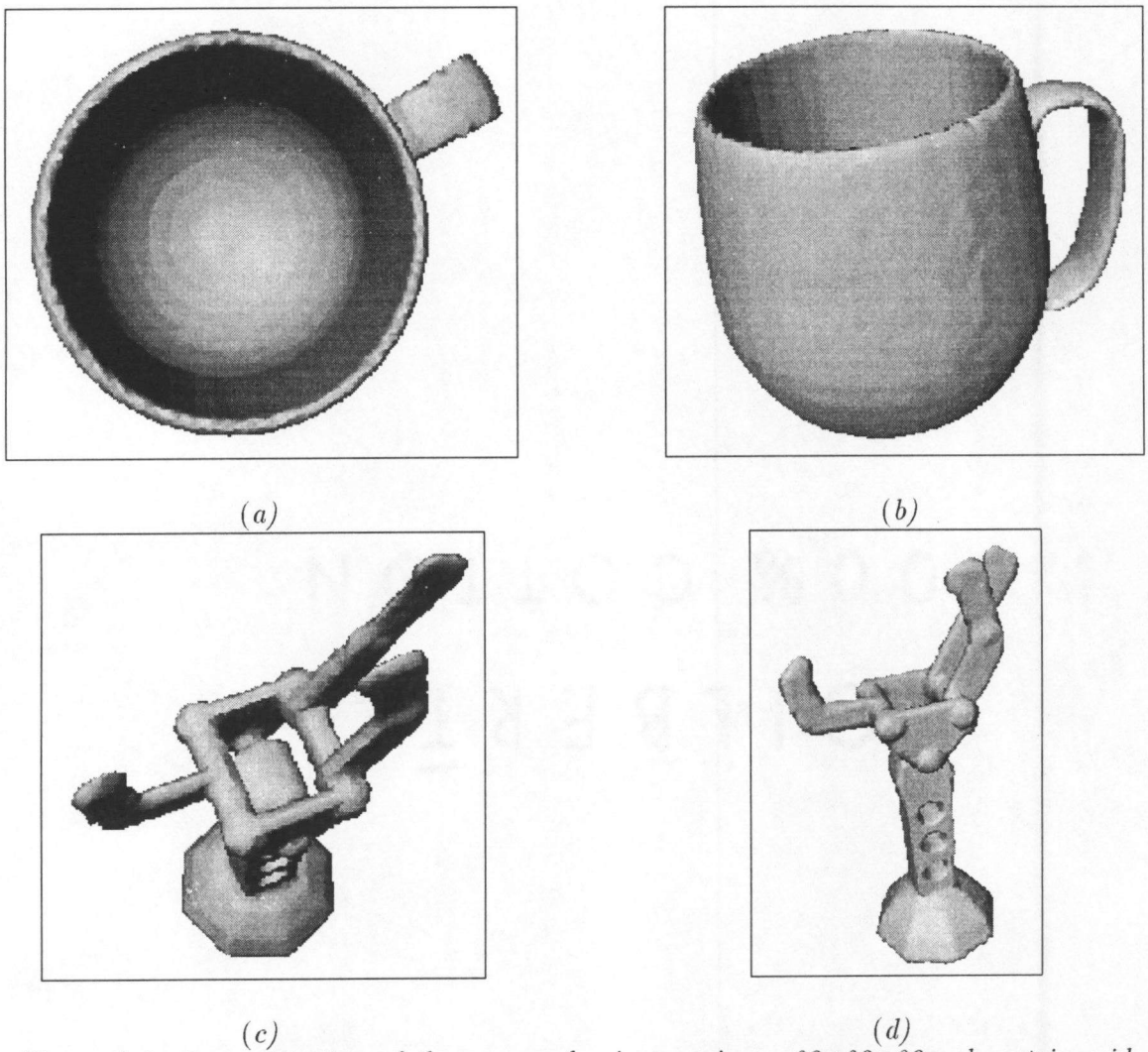


Figure 5.4: *Reconstruction of the mug and gripper using a 80×80×80 volumetric grid. Each model was constructed with 12 noiseless range images.*

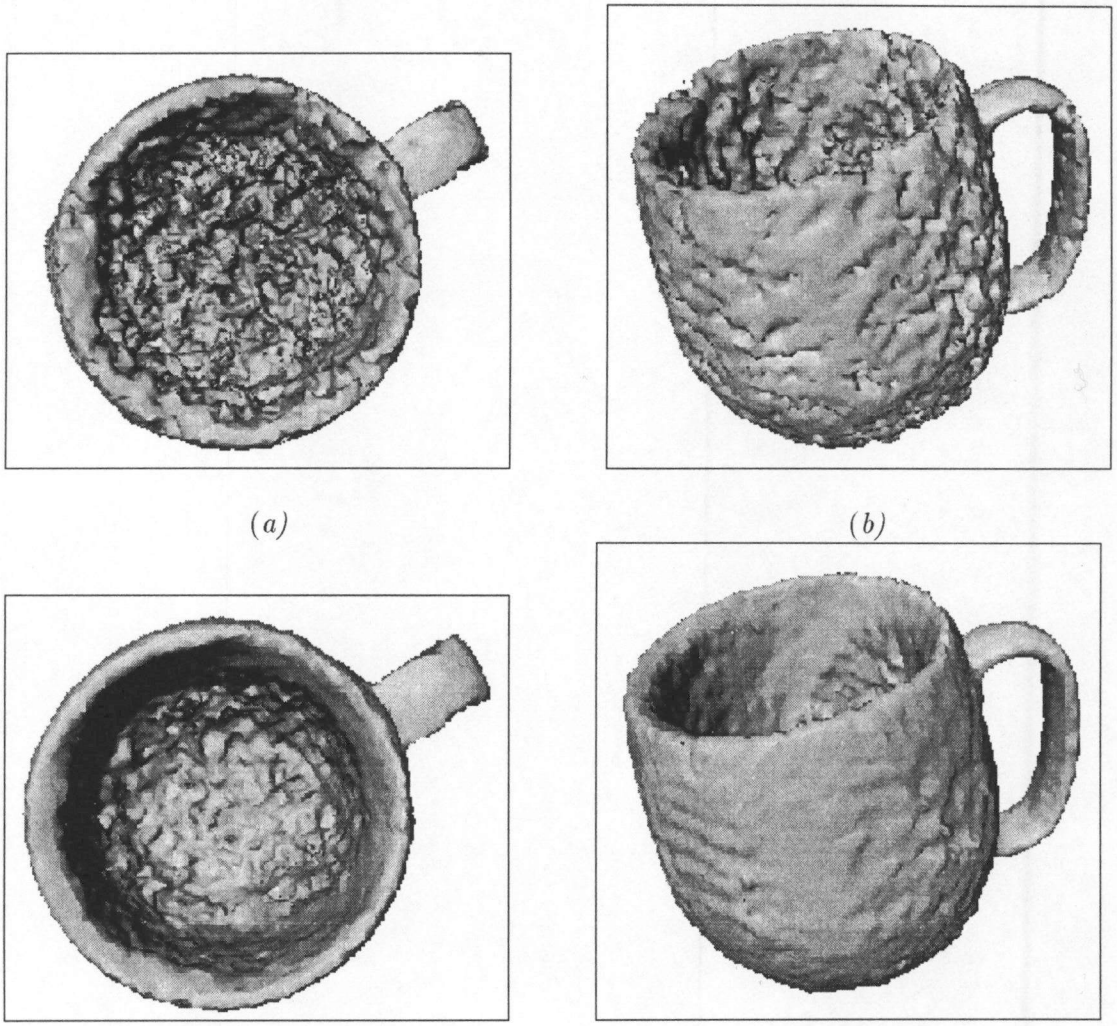


Figure 5.5: *A reconstruction of the mug on a 80×80×80 grid with uniform noise added. The noise has an average of 13% of a typical range measurement. 12 range images were used to create this model shown in (a) and (b). The 12 images were cycled through ten times for the model shown in (c) and (d).*

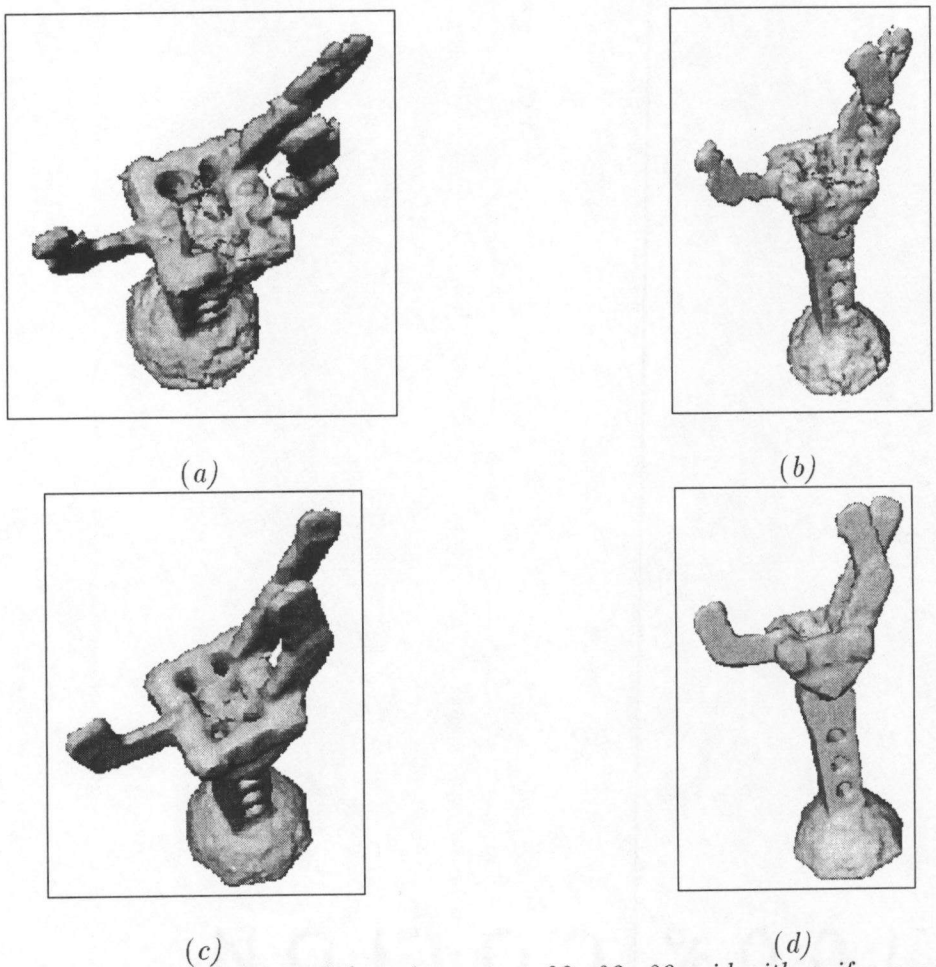


Figure 5.6: *A reconstruction of the gripper on a 80×80×80 grid with uniform noise added. The noise has an average of 13% of a typical range measurement. 12 range images were used to create this model shown in (a) and (b). The 12 images were cycled through ten times for the model shown in (c) and (d). The model improves with additional information.*

Figure 5.7: *The ideal reconstruction of the mug model using the number of times confidence measure. The colors represent relative confidence level.*

the noise in the previous example is used. The noise in each cycle has the same density function, however the individual noise values are different to simulate random noise that varies from range image to range image.

5.1.1 Confidence Measure on Synthetic Data

This section presents a few results of the confidence measures that we described in the previous chapter. The first example, figure (5.7), is the ideal reconstruction of the mug model using the number-of-times confidence measure. Notice the red spot on the lip of the mug. This is a region that has not been mapped because it is along an edge in the views that we have taken. This information could be filled in by an additional view. This shows that the number-of-times confidence measure is valuable for deciding which regions require additional imaging.

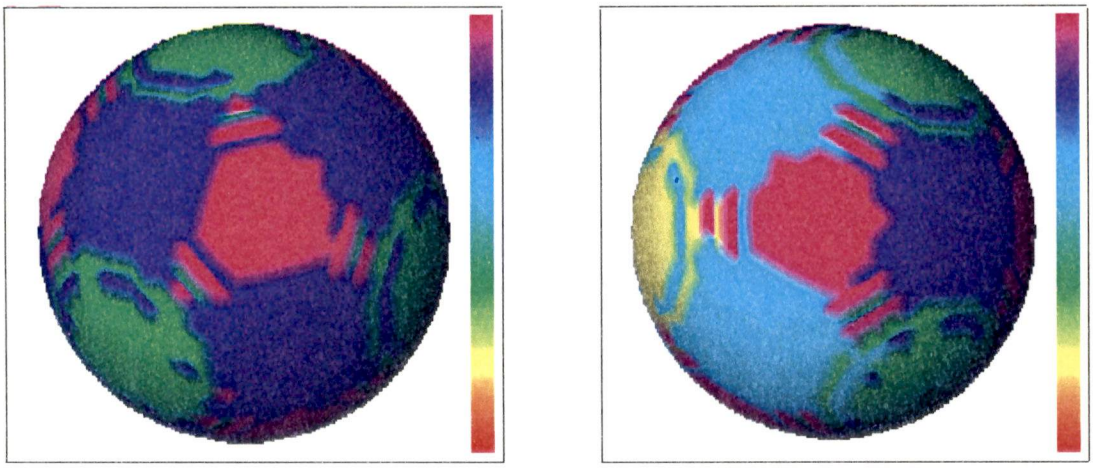
The next example uses the slope-based confidence. This example uses six synthetic range images of a sphere. We use a $60 \times 60 \times 60$ volumetric grid for the reconstruction.

Figure (5.8a) shows the result using the same noise estimate for each range image, while (5.8b) uses one noise estimate for the first three images and another for the last three. In (5.8a) the regions where the range images overlap is clearly evident. In (5.8b) the higher noise estimate for three of the images causes the confidence to be lower for half of the sphere.

The third confidence example shows the slope-normal approach. This method is most clearly demonstrated using a reconstruction made from only one view. Figure (5.9a) shows the result when the range image contains no noise. Regions that are perpendicular to the scanner line-of-sight have a higher confidence. Figures (5.9b) and (5.9c) show effect of noise on the result. Figure (5.9b) is reconstructed from an image with uniform noise with a width of .1 units (the width of the mug is approximately 4.5 units). Figure (5.9c) used a range image with uniform noise with a width of .2. Two effects are noticeable with increasing noise. Firstly, increasing the noise in the range image makes the estimated surface normal more random; hence the confidence does not transition as smoothly. When a sufficient amount of noise is added, the confidence is similar in all regions despite an apparent difference in orientation. Secondly, the increasingly random nature of the surface normals tends to reduce the overall confidence of a region. These two effects lead us to conclude that this method is most useful for range scanners that have low noise levels. We will show one such example later in this chapter.

5.2 Reconstruction using Real Data

In the previous section, we showed how our volumetric method performs on synthetic data. Synthetic data is useful for testing an algorithm, however, data from a real range finder must be used to demonstrate that a technique is valid for actual applications. This



(a) (b)

Figure 5.8: *The effect of differing noise levels in the slope-based confidence metric. The colors represent relative level of confidence. (a) is a reconstruction using six range images with the same noise estimate. (b) uses one noise estimate for the first three images and another for the remaining three.*

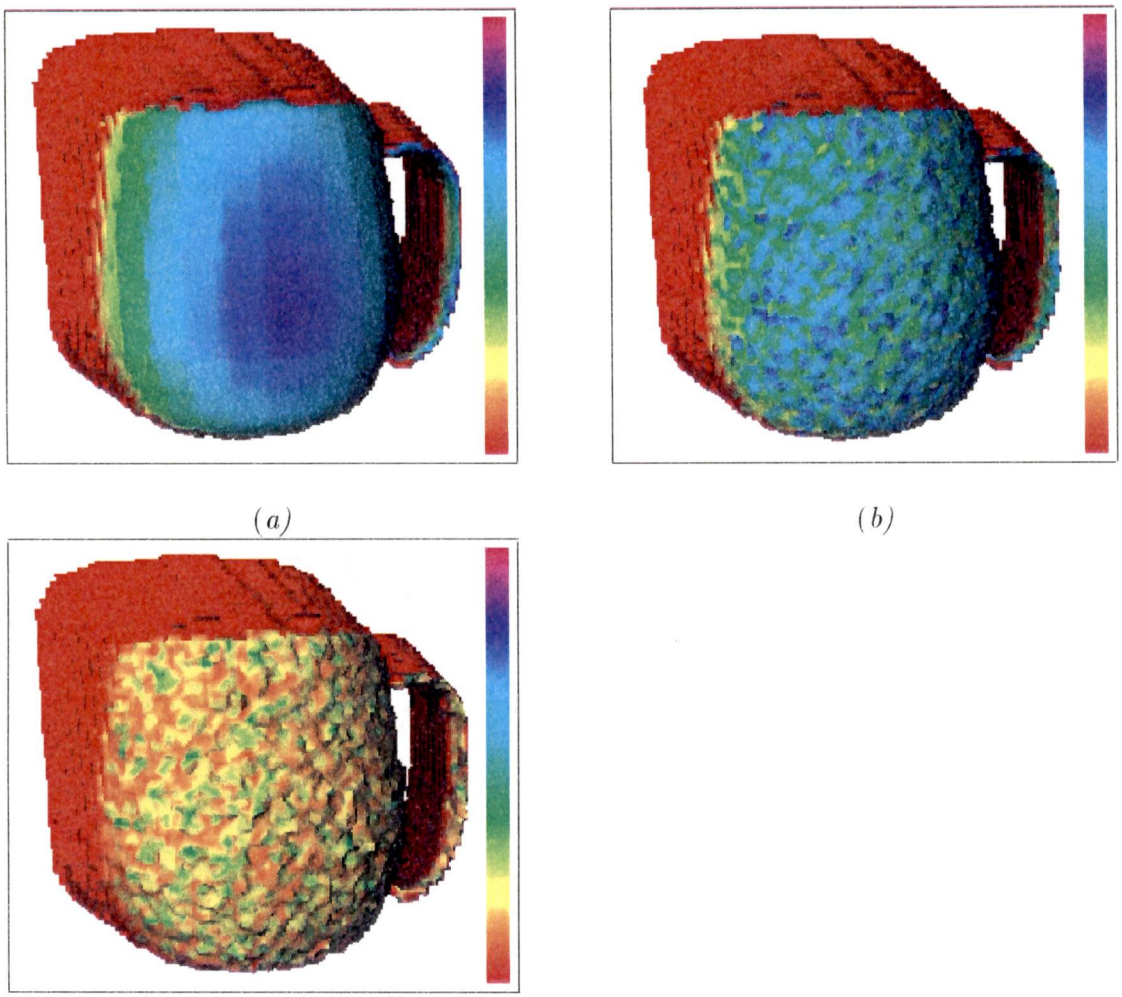


Figure 5.9: *The effect of varying levels of noise on the slope-normal confidence metric. The colors represent relative level of confidence.*

section shows how our technique fares on real data from two different scanners. First, we use range images taken with a Perceptron scanner to reconstruct objects. Then, we demonstrate how our method reconstructs a scene from images taken with a Coleman Coherent Laser Radar scanner.

5.2.1 Object Modeling

This section presents how our method can be used to reconstruct real objects from data taken with a Perceptron laser range finder. We use the same general steps as with the synthetic data, however, we do not apply the median filter described in step (3a).

Working with real data presents several challenges that we did not face with synthetic data:

1. Synthetic data is precisely registered. This is not true for real data.
2. The camera geometry for synthetic data is known exactly, while the geometry for a real camera will not be.
3. The pdf of the noise in a real range image may be unknown.

There are a couple of ways to overcome challenge (1). We can take range images from arbitrary views and then apply a registration algorithm to them. Another solution is to take views from known positions so the transformation matrix between views can be determined easily. We use a variation of this idea; we keep the range camera fixed and rotate the object a known amount. We use an optical tripod that has graduated degree marks to rotate the object a known amount θ. The distance d from the range camera to the center of rotation of the optical tripod must be determined. The optical tripod that we use has a hole in the center of the platform about which it rotates. We insert a pencil in this hole and take a reference scan (figure 5.10) to approximate this distance. We use the center of rotation of the optical tripod as the origin of the world coordinate

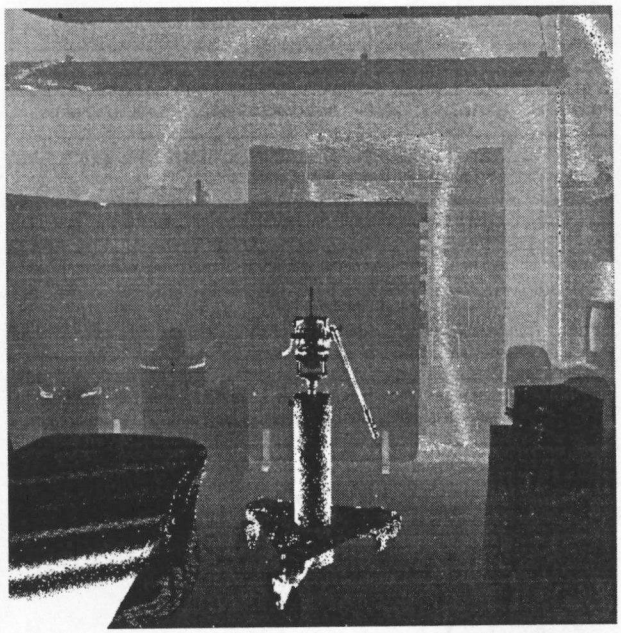


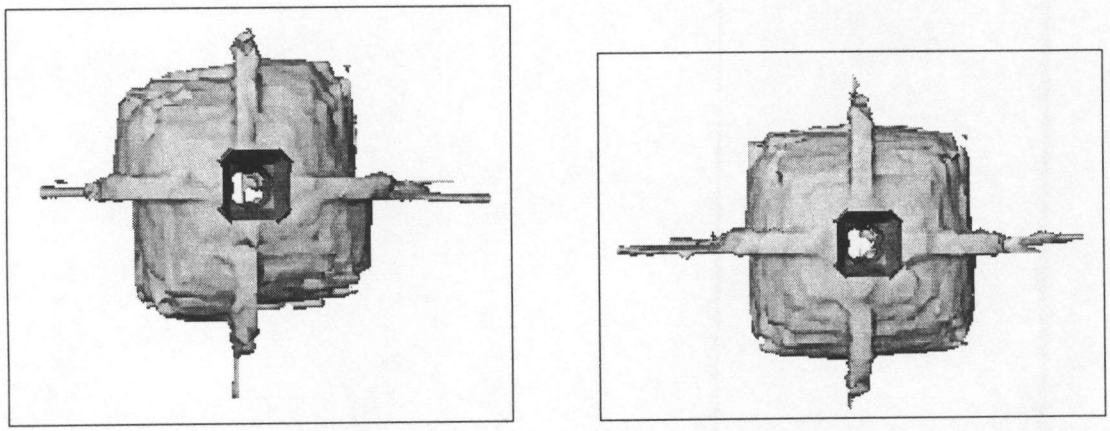
Figure 5.10: *A reference range image that we used to find the distance from the range finder to the center of rotation. We inserted a pen in a hole in the center of the optical tripod. This represents the axis about which the tripod rotates.*

system. Although the object is rotating while the camera remains still, we calculate the transformation as if the object is still and the camera moved to positions about the it; the results are equivalent.

We obtain an initial estimate of the camera transformations using the known distance d and the rotation angle θ. We also assume that the initial camera transformation has the same orientation as the world coordinate system. This gives a good first estimate of the transformations, however, the optical tripod or the laser rangefinder may be tilted and we need to correct for this. The tilt is made up of two components with respect to the camera; one component is a forward/backward tilt while the other component is a left/right tilt.

We corrected the left/right tilt by using reference scans of a basketball. We adjusted the distance d to be greater than its actual value so the reconstruction will only use silhouette information. We modify our program to only carve away free space to a distance of d from the camera. This will allow us to see the contribution of the individual range scans. We know that scans from opposite sides of the basketball should align in the reconstruction. If they do not we adjust the orientation of the initial camera transformation and recalculate the remaining transformations until they do. Figure (5.11a) shows the reconstruction of a basketball from four scans before correcting for the left/right tilt. Figure (5.11b) shows the result after the correction. Note that the views from opposite sides of the ball are more closely aligned after the correction. We correct for the forward/backward tilt by adjusting the camera orientation to make vertical sections of the object appear vertical on the model.

The next challenge to overcome is determining the geometry of the rangefinder. We used a Perceptron model P5000 laser range finder for these experiments. In [52], Chase characterizes this type of rangefinder and finds that spherical coordinates closely approximates the camera geometry.



(a) (b)

Figure 5.11: *This figure shows how we align misregistered range images. Figure (a) shows a top view of a reconstruction of a basketball using four misregistered range images. Notice that the opposite sides of the ball do not line up as they should. This is caused by a left/right tilt in the optical tripod or range finder. Figure (b) shows the result after a tilt correction is made.*

The third challenge is the noise in the rangefinder is unknown. We have used a certainty graph that would be derived, as discussed in chapter 4, from uniform noise with an average value of 7.5 centimeters. The true pdf of the Perceptron scanner is likely to be closer to Gaussian, however, the uniform approximation seems to give reasonable results and requires less calculation time.

5.2.1.1 Results

This section shows the results of our volumetric method on four objects. The first test object is five reams of printer paper stacked on top of each other. We took the six range images shown in figure (5.12). We have applied an upper and lower threshold to all of the range images of objects shown in the remainder of this document. This enhances the contrast of the objects and makes them easier to see. However, we used the original range data in our algorithm. Figure (5.13) shows a surface plot of the first two range images of five reams of paper. They show the types noise that our method must overcome. These were also thresholded to emphasize the object. Figure (5.14) shows two views of the reconstructed model. Some noise is present, however the basic shape is correct. There are some flaws near the edges of the model. This is because we ignore certain step edges as we explained in the previous section. In this, case we do not have enough views and poses to fill in the missing data.

The remaining two images in figure (5.14) are two views of the reconstruction colored coded with confidence. We used the number-of-times confidence measure for these reconstructions for a couple of reasons. Firstly, images from the Perceptron scanner contain significant amounts of noise. As we mentioned earlier, the slope-normal method is best suited for low noise images. Secondly, we have no information to suggest that one range image contains any more noise than any other in the set; hence the slope-based confidence would be merely a scaled version of the number-of-times method.

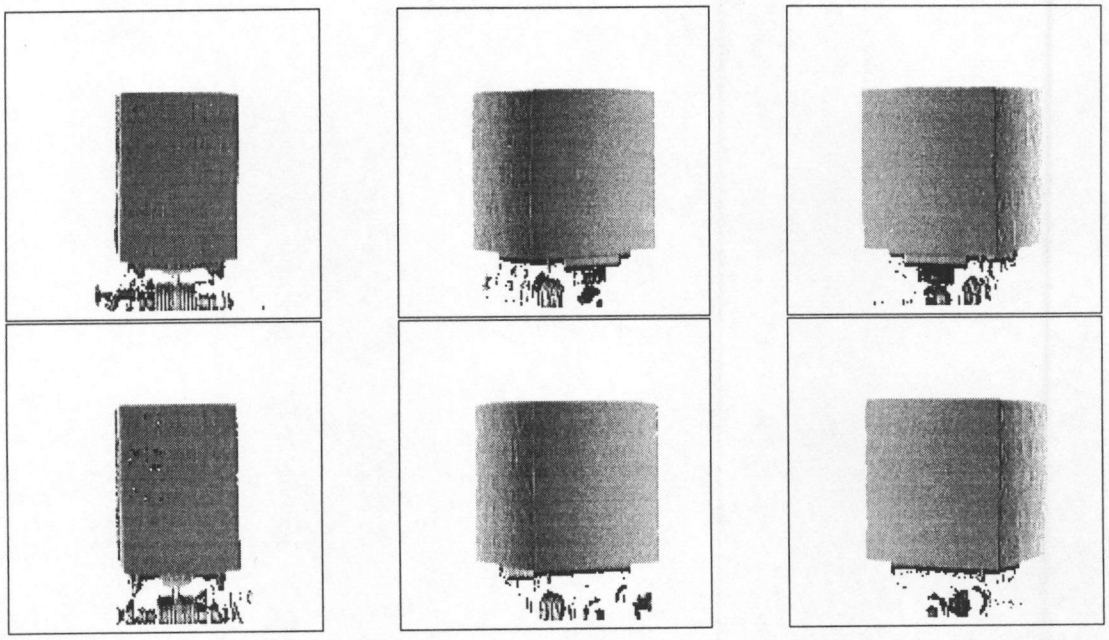
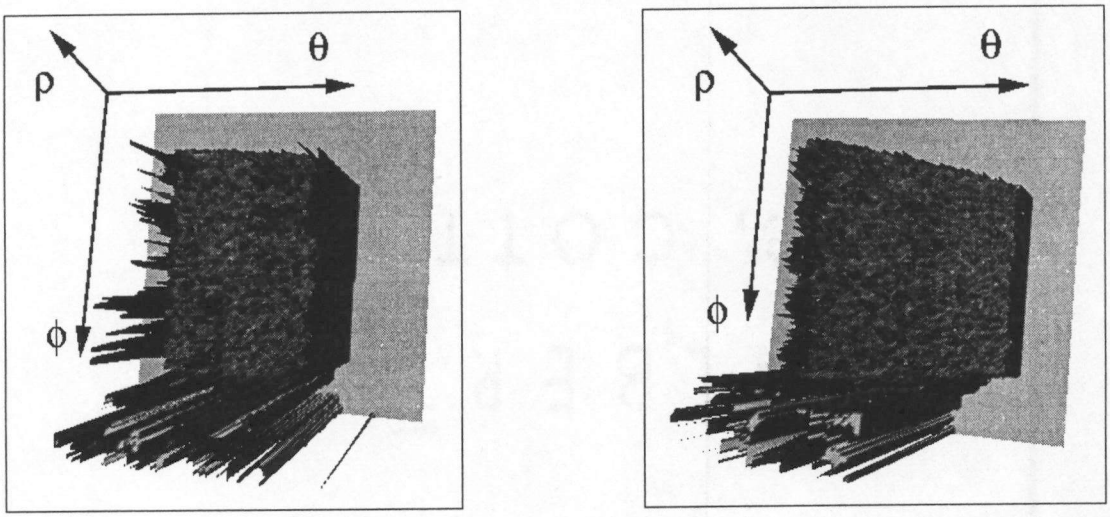
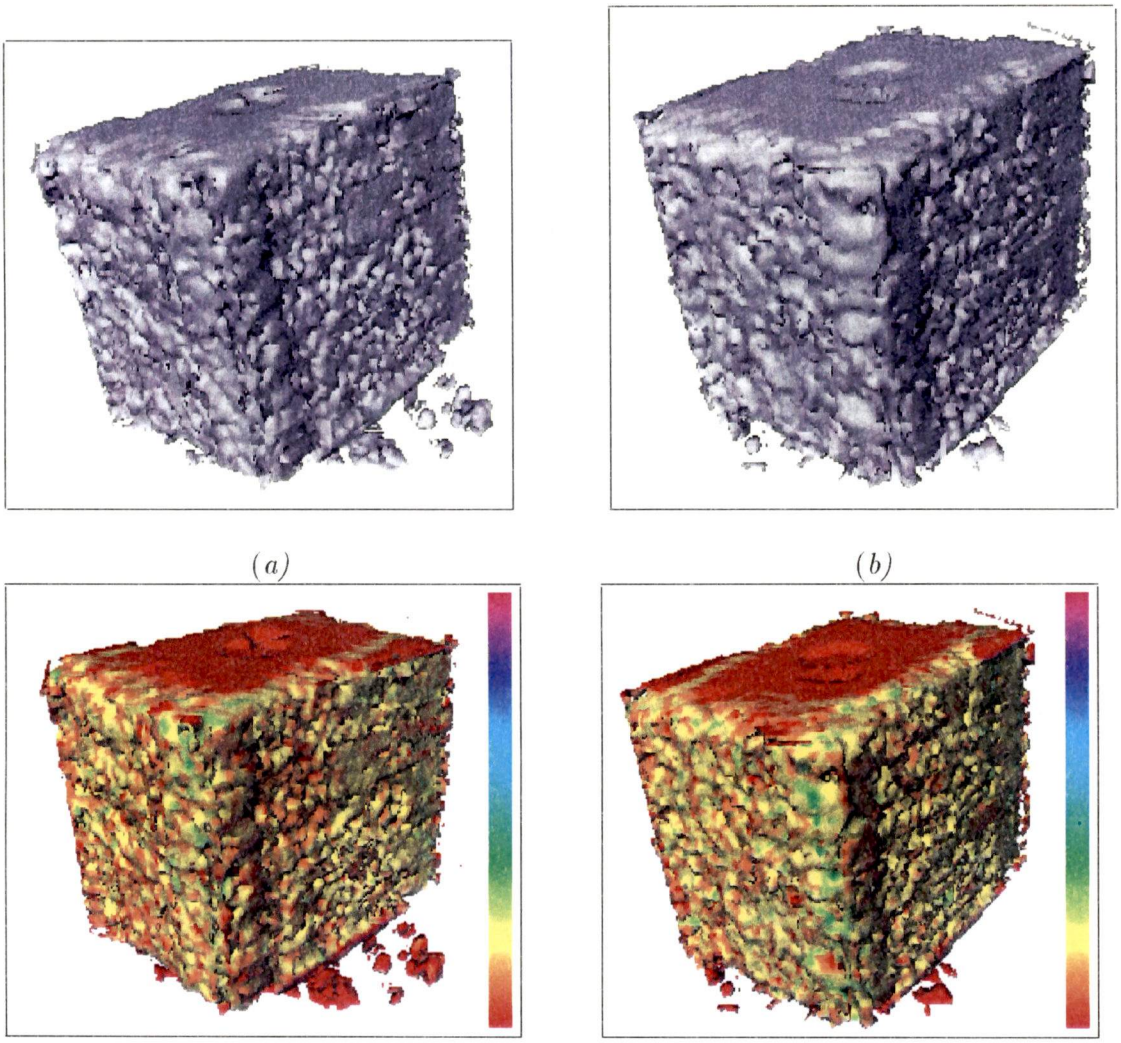


Figure 5.12: *Six range images taken of five reams of printer paper stacked on top of each other. We rotated the paper 60 degrees between each image. We applied an upper and lower threshold to these images for this paper to enhance the contrast.*



(a) (b)

Figure 5.13: *Surface plots of the first two paper range images. They show the types of errors that our method must overcome. We applied an upper and lower threshold for clarity.*



(a) (b) (c) (d)

Figure 5.14: *Two views of the model reconstructed from the paper images on an 80× 80× 80 grid. Some noise is present, however the basic shape is correct. Figures (c) and (d) are two views of the confidence coded reconstructions.*

We constructed the second test object from four reams of printer paper and several blocks of wood. The first block is a 2×4 segment a couple of inches long. It has a 3/4 inch hole drilled part way though its width and a 1/4 inch drill part way through its depth. The second block is a 2×4 with a triangle shape cut out of it. The center object is make up of a small pyramid sitting on a small cube. We took the twelve range images shown in figure (5.15). Surface plots of the first two are shown in figure (5.16). We thresholded these to emphasize the object. Figure (5.17) shows the model reconstructed from the range data. The basic shape is true. The larger hole in the 2×4 is evident; there is a slight indentation where the smaller hole is, but it is hardly noticeable.

The third test object is a styrofoam head similar to the ones used to hold wigs. We took 12 range images of the head, rotating it 30 degrees between each image. Figure (5.18) shows the range images. Figure (5.19) is a surface plot of the first two range images. Again, we applied an upper and lower threshold for clarity. Figure (5.20) shows two views of the reconstructed model. The reconstruction is far from ideal, however, much of the noise evident in the surface plots has been eliminated. The shape of the head is clearly apparent, as is the shape of the ream of printer paper on which the model is sitting. Even the black base that the head is mounted on can be seen, even though our laser scanner typically does not get good returns from black objects.

The last test object is a computer monitor sitting on a ream of printer paper. Figure (5.21) shows the 12 range image that we acquired. The black spot in the first image is noise from the sensor reflecting off the computer screen. Figure (5.22) shows a surface plot of the first two range images. Note that the spike on the screen has been clipped. We used the original data when we constructed our model. Figure (5.23) shows the reconstruction on an 80×80×80 grid. Notice that our technique has eliminated the large spike from the computer screen. This is a case where carving away free space is useful. The power cable was not reconstructed very well; however, notice in figure (5.21) that some of the range

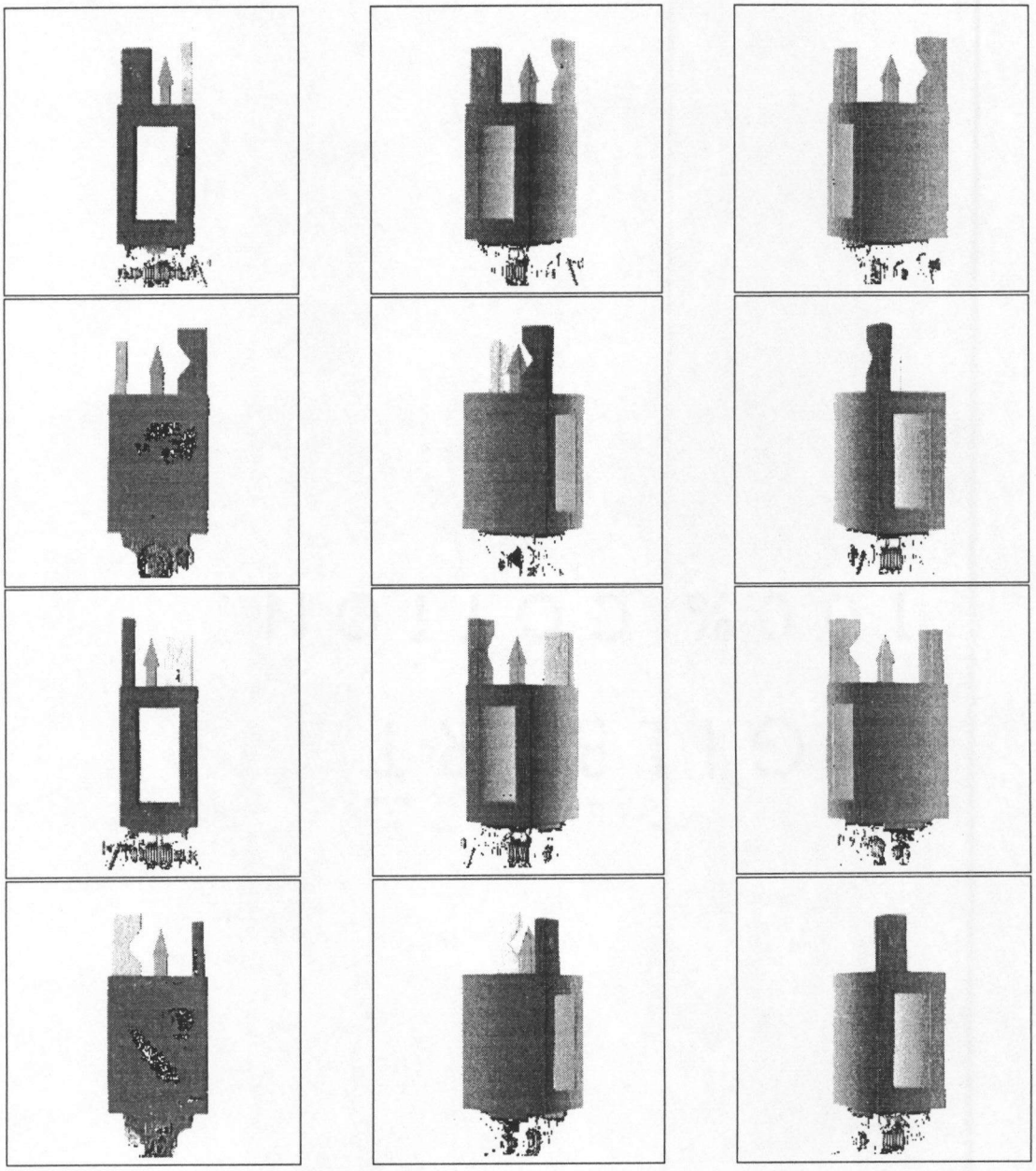
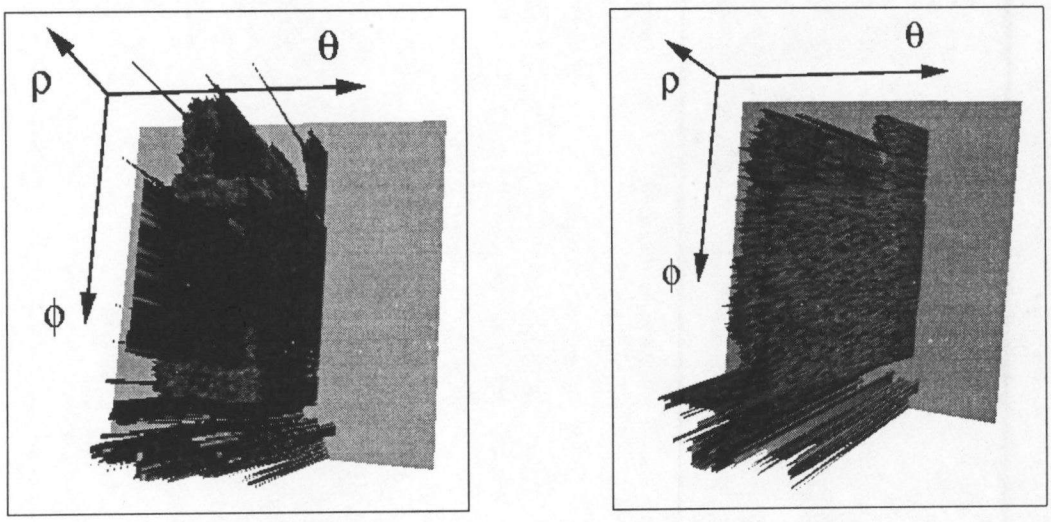


Figure 5.15: *Twelve range images taken of the 'shapes' test object. We rotated the object 30 degrees between each image. We applied an upper and lower threshold to these images for this paper to enhance the contrast.*



(a) (b)

Figure 5.16: *Surface plots of the first two 'shapes' range images. They show the types of errors that our method must overcome. We applied an upper and lower threshold for clarity.*

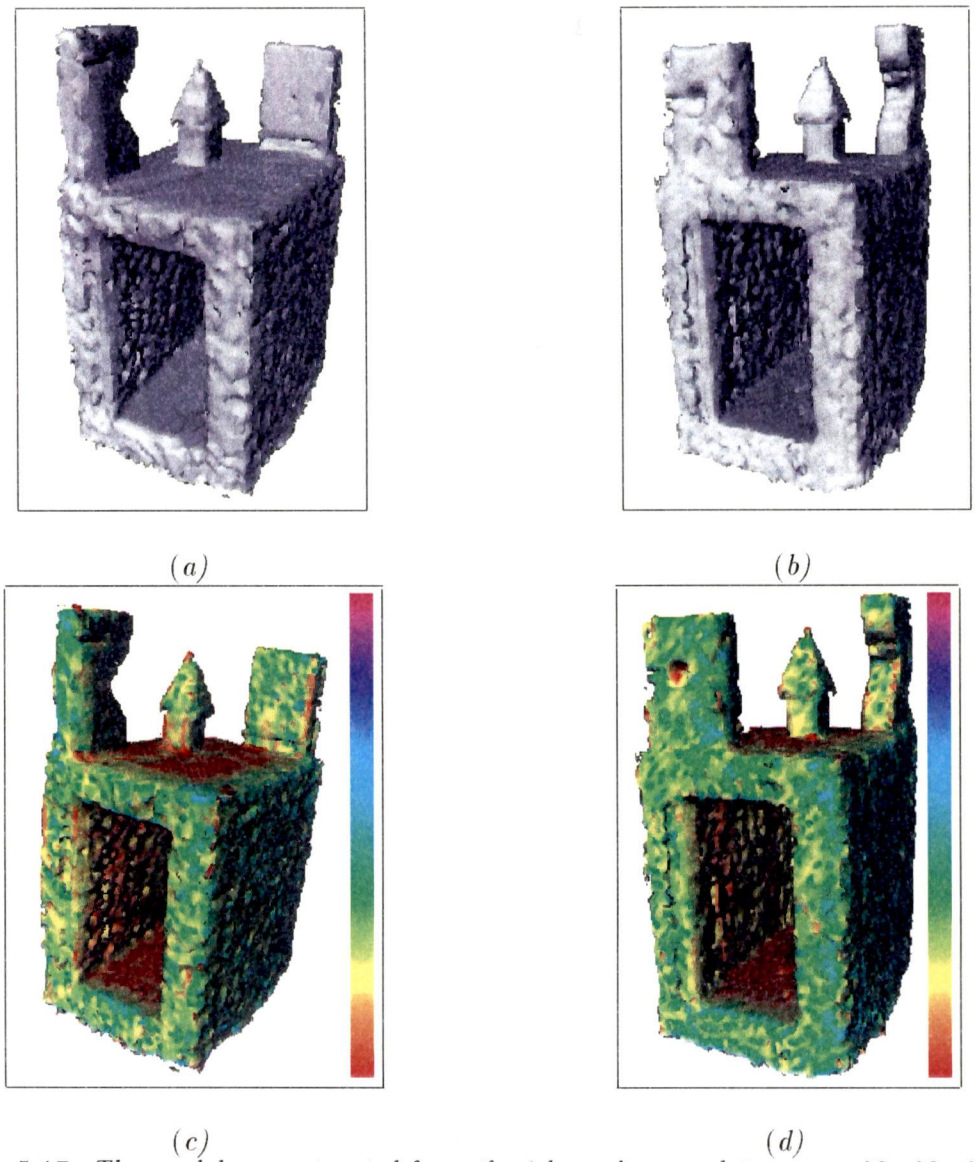


Figure 5.17: *The model reconstructed from the 'shapes' range data on an 80×80×80 grid.*

Figures (c) and (d) are two views of the confidence coded reconstructions.

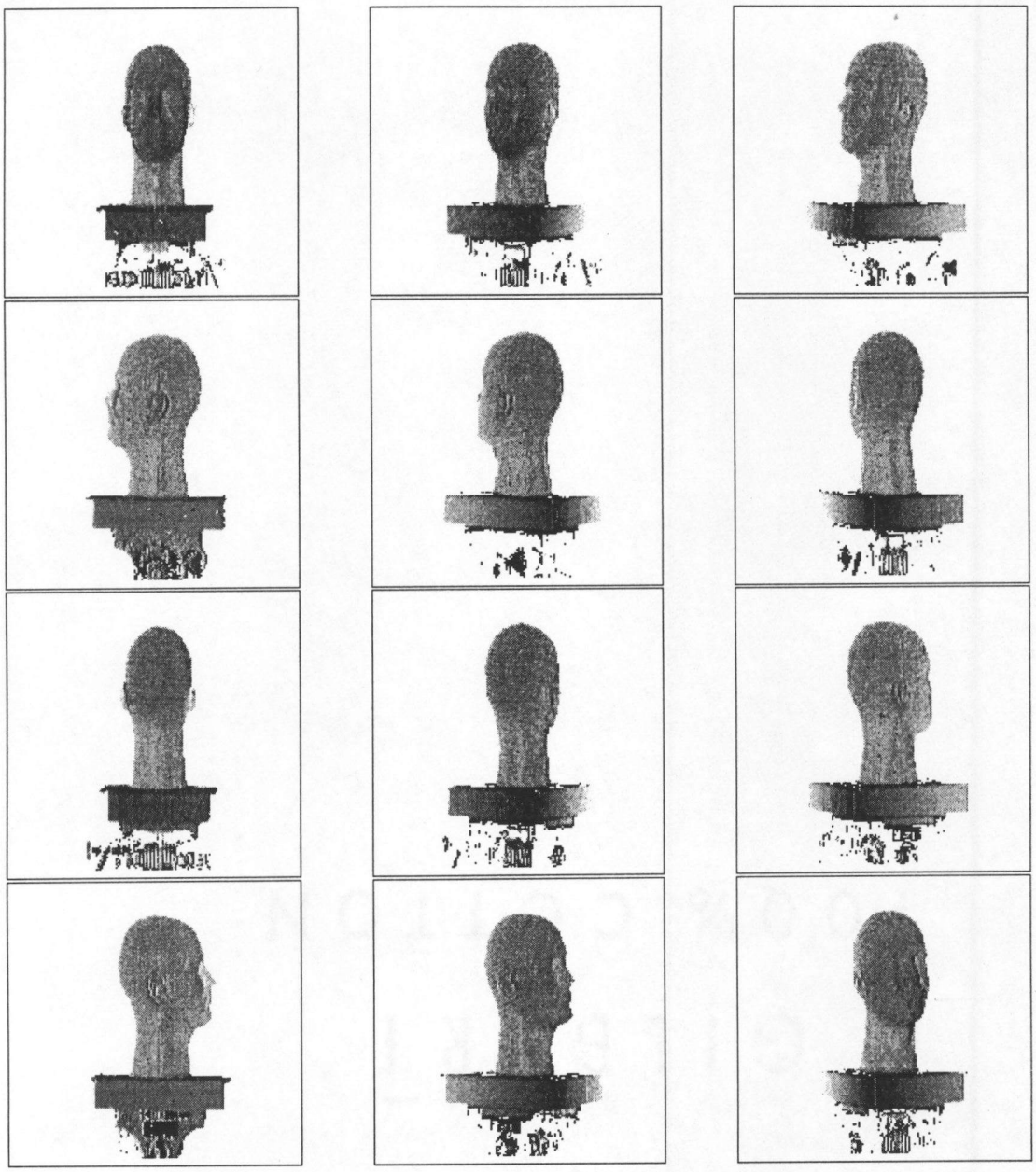
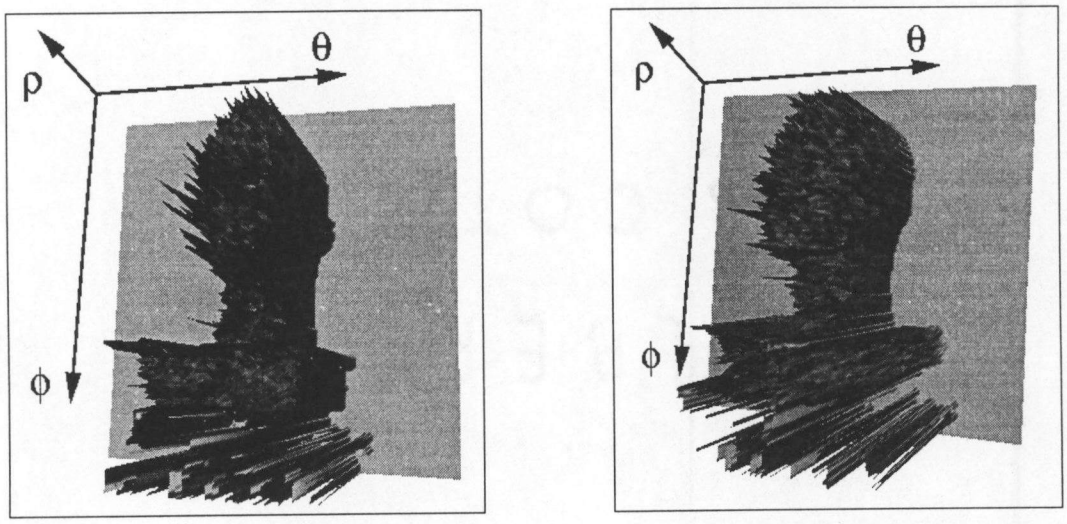


Figure 5.18: *Twelve range images taken of a styrofoam head sitting on a ream of printer paper. We rotated the head 30 degrees between each image. We applied an upper and lower threshold to these images for this paper to enhance the contrast.*



(a) (b)

Figure 5.19: *Surface plots of the first two head range images. They show the types of errors that our method must overcome. We applied an upper and lower threshold for clarity.*

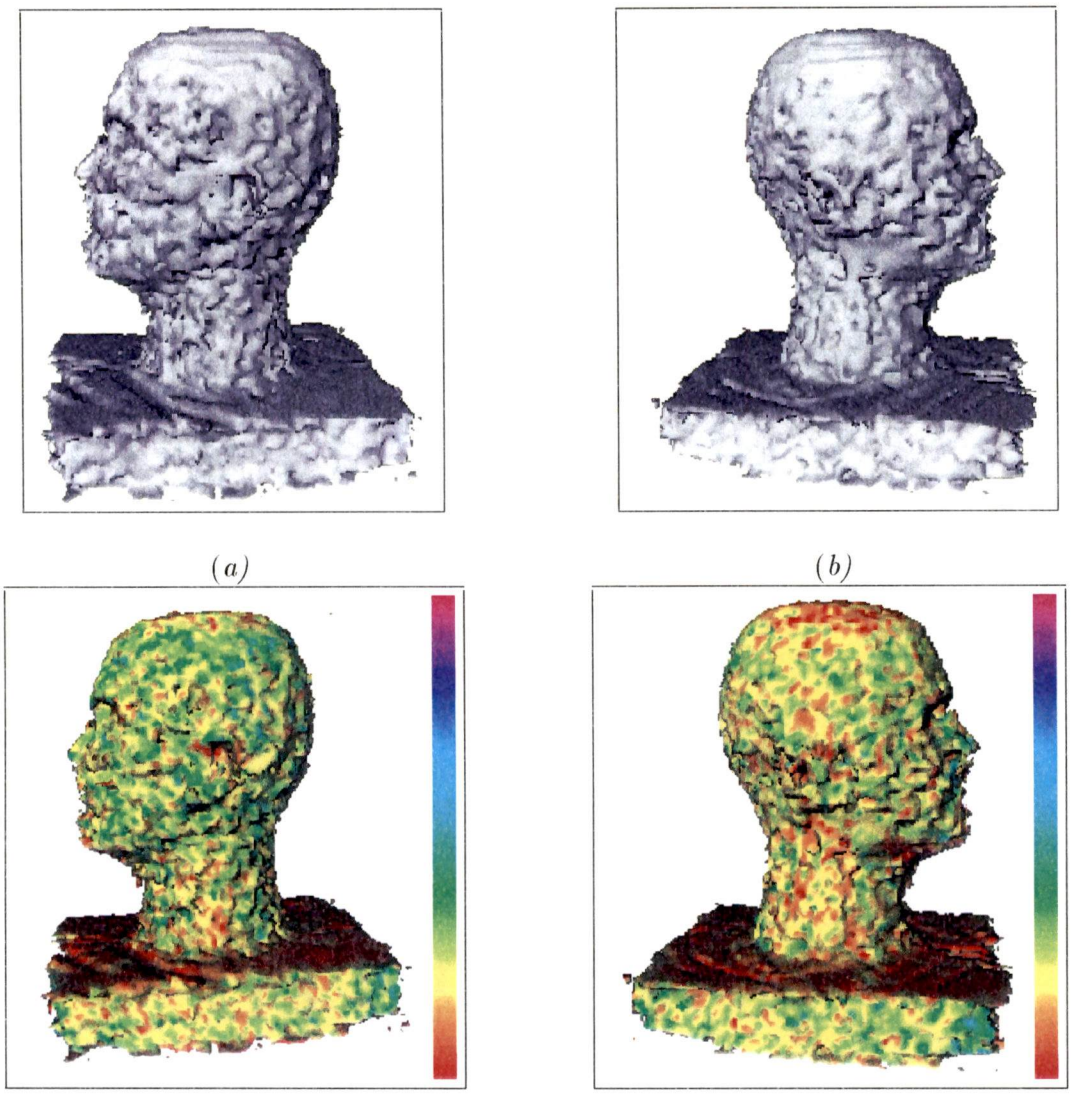


Figure 5.20: *Two views of the reconstructed head model on an 80×80×80 grid. The shape of the head is clearly apparent, as is the shape of the ream of printer paper on which the model is sitting. Figures (c) and (d) are two views of the confidence coded reconstructions.*

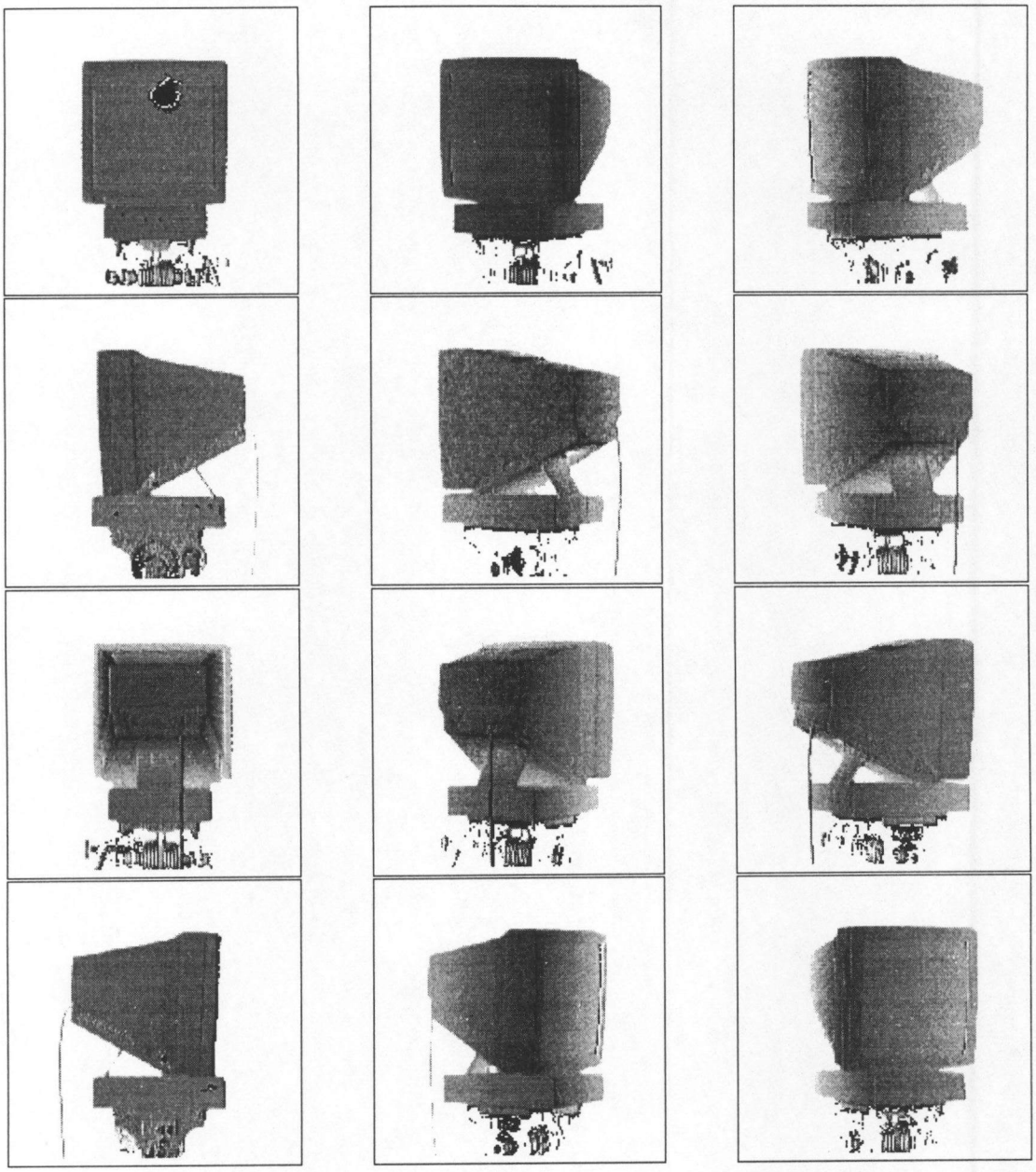
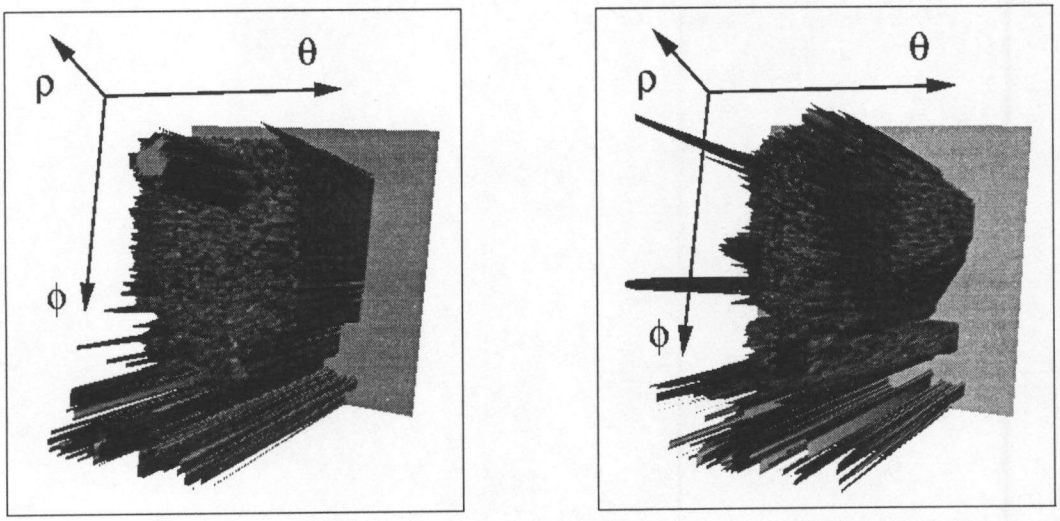
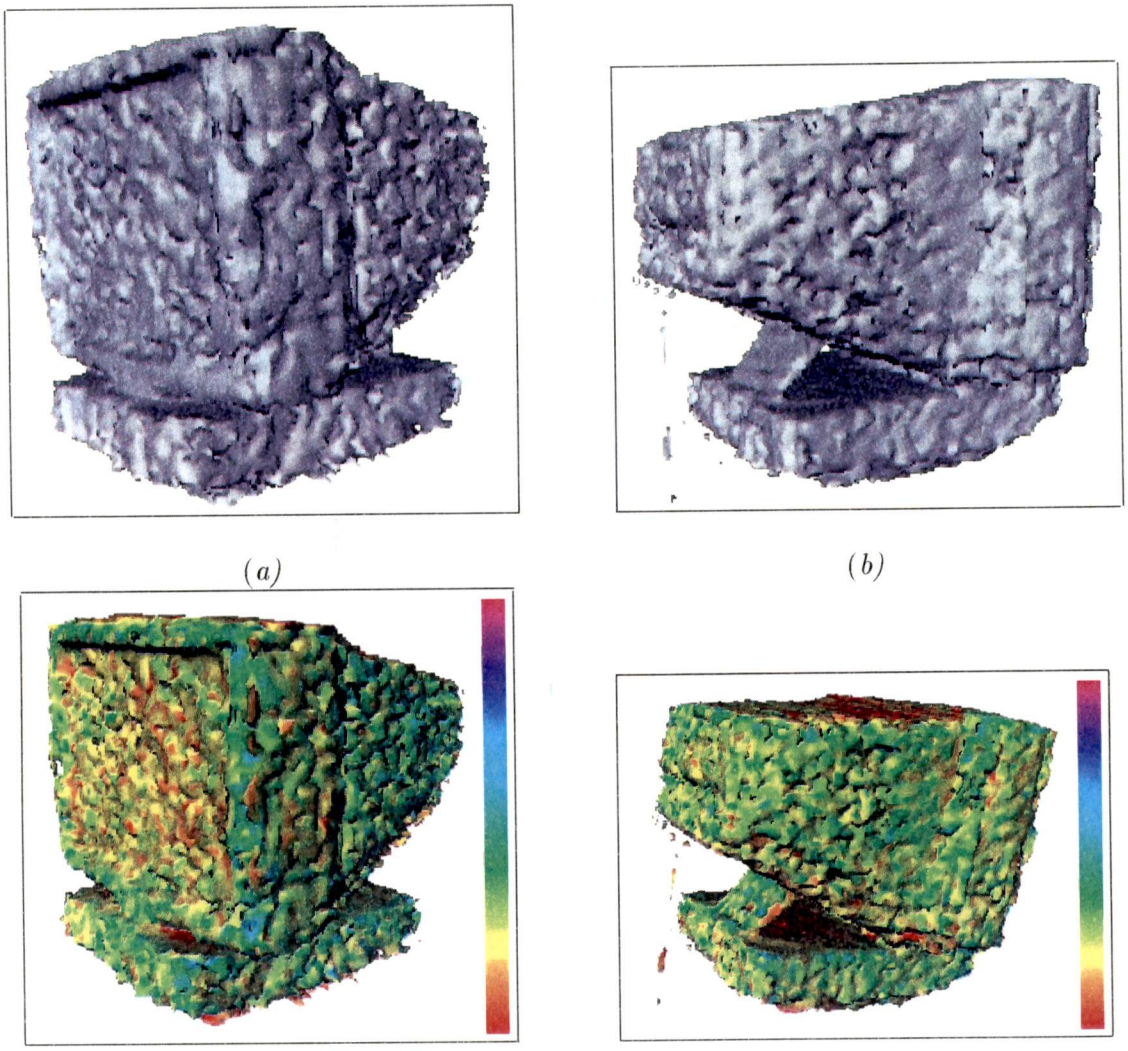


Figure 5.21: *Twelve range images taken of a monitor sitting on a ream of printer paper. We rotated the monitor 30 degrees between each image. The black spot in the first image is noise from the sensor reflecting off the computer screen. We applied an upper and lower threshold to these images for this paper to enhance the contrast.*



(a) (b)

Figure 5.22: *Surface plots of the first two monitor range images. They show the types of errors that our method must overcome. We applied an upper and lower threshold for clarity. Note that the spike in the first image is clipped.*



(a) (b) (c) (d)

Figure 5.23: *Two views of the reconstructed monitor model on an 80×80×80 grid. Figures (c) and (d) are two views of the confidence coded reconstructions.*

images have gaps in the cable that may have hindered the reconstruction. Overall the model is good, especially the reconstruction of the thin stand that supports the monitor.

5.2.2 Scene Modeling

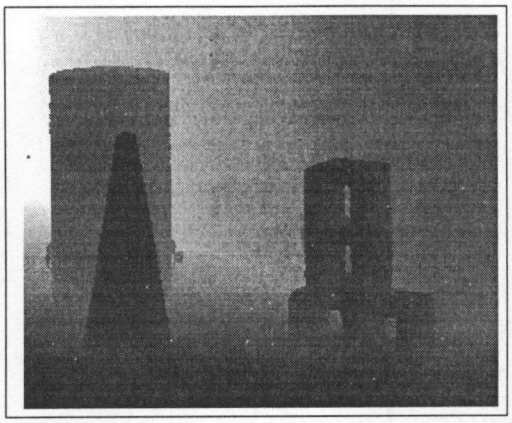
In this section we use our volumetric technique to model a scene. We use the same steps as with the synthetic data, except we do not apply the median filter in step (3a). Since we are doing an entire scene, we also do not try to threshold the foreground from the background as we do with object modeling. We will discuss the result of this later in this section.

This example shows our method applied to three range images taken with a Coleman Coherent Laser Radar scanner of a simple scene containing a traffic cone. Figure (5.24) shows the three images that we have.[1] We have registration and camera geometry information for these images as well.

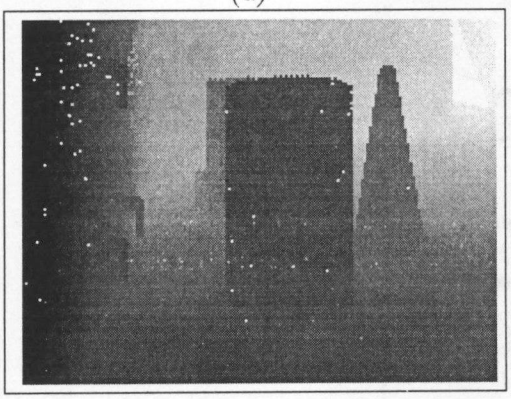
Figure (5.25) shows the reconstruction on a $200\times200\times200$ grid of the scene using the two least noisy images (5.24a) and (5.24c). The low noise level in the images allows our method to capture such details as the handles and vents on the lockers along the back wall. Regions that are occluded from both views are evident as the shadow-like effect behind objects. The odd shape between the cone and the blocks is also caused by occlusion. Labeling occluded regions as occupied by an object is desirable for robotic navigation tasks where a robotic should only be sent into areas that have been verified to be free from obstacles.

Figure (5.26) shows the reconstruction of the cone scene on a $200\times200\times200$ grid using all three range images. The third image contains more noise than the others; hence we

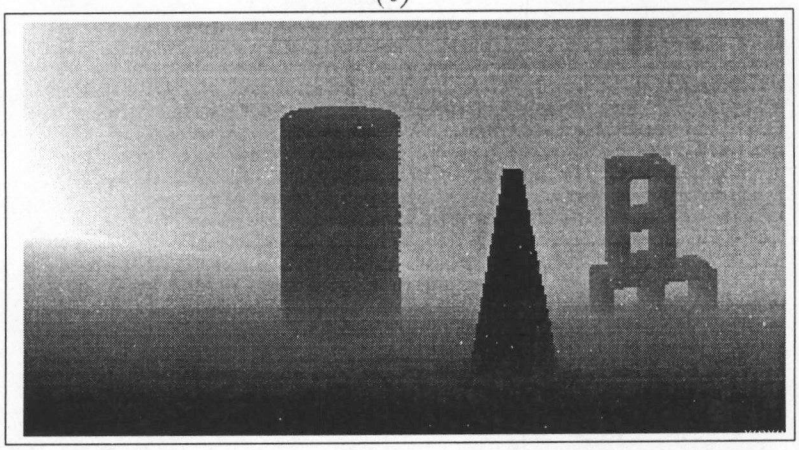
[1] The laser range data files were provided by the Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, Managed by Lockheed Martin Energy Research Corp. for the U.S. Department of Energy under contract DE-AC05-96OR22464.



(a)



(b)



(c)

Figure 5.24: *Three range images taken from a Coleman scanner.*

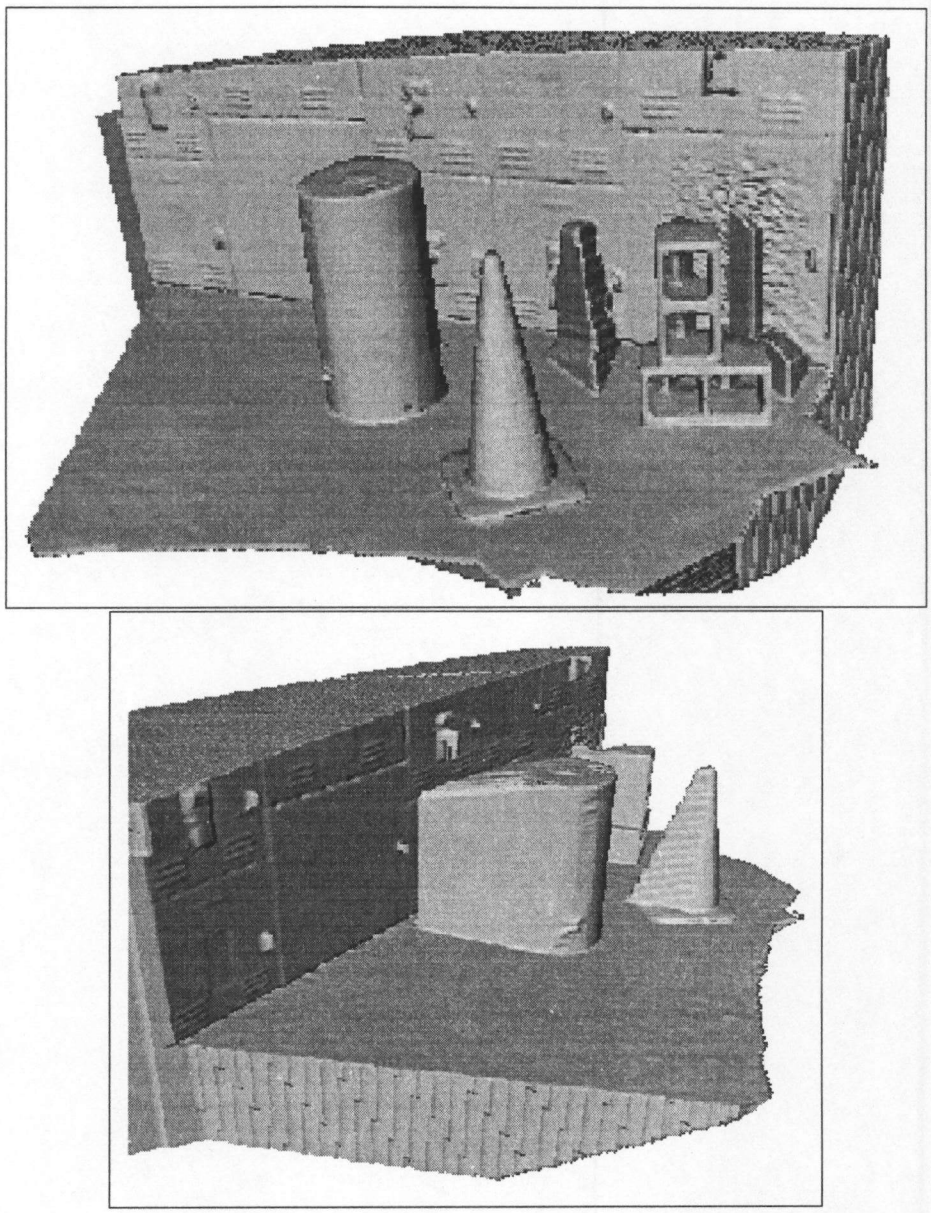


Figure 5.25: *The reconstruction of the cone scene using two of the three range images on a 200×200×200 grid. Regions that are occluded from both viewpoints are evident in the shadow of objects and between the cone and the blocks.*

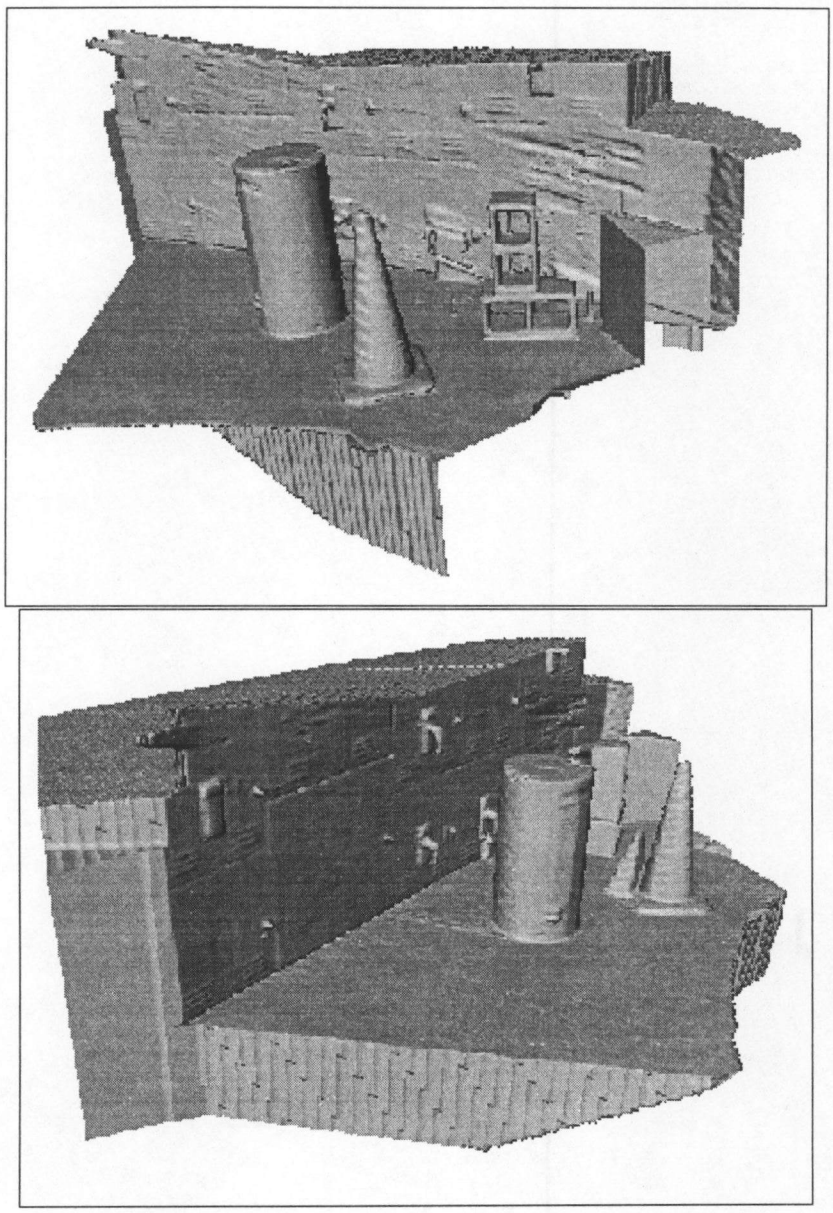


Figure 5.26: *The reconstruction of the cone scene using all three of range images on a 200×200×200 grid. The additional range image contains more noise than the other two. This caused the scarring on the back wall. The third image has filled in some of the occluded regions.*

applied a 3×3 median filter to it before we added it to our model. The higher noise level caused the scarring especially noticeable along the wall behind the blocks. Also notice that there are small ridges on the traffic cone. This is the stairstep effect that results from the discrete nature of the range image. Recall that we avoided this in object modeling by ignoring voxels that map back to edges in the range image. However, in situations such as scene reconstruction, finding edges is not as straightforward as in well controlled object modeling environments. Therefore we chose to let our technique operate across edges in this case. The effect is not very pronounced in this reconstruction; it is not noticeable on the barrel, for example. The stairstep effect is related to the voxel size to pixel size ratio. When the voxel size is large enough, our method will tend to average across several pixels and make the effect less noticeable. When the voxel size is small, our method will reconstruct the stairstep effect in the range image more precisely.

There are two solutions to the stairstep problem. One can select the voxel size to be large enough to average across several pixels. The other approach is to apply a step edge detector to the range image and ignore voxels that map to the edges. The second solution could be problematic for noisy range images, but may be feasible in other situations.

The Coleman scanner creates low noise images which are well suited for the slope-normal confidence measure that we described earlier. Figure (5.27) shows the result using only the first 'cone' image. Regions that are sloping away from the scanner clearly have a lower confidence. Regions occluded from the scanner have the lowest confidence (red).

Figure (5.28) shows the reconstruction using all three range images. The second range image has more noise than the other two; we median filtered it before we added it to our reconstruction and we used a higher noise estimate for it. Regions that are occluded from all viewpoints have the lowest confidence while regions in all views have the highest.

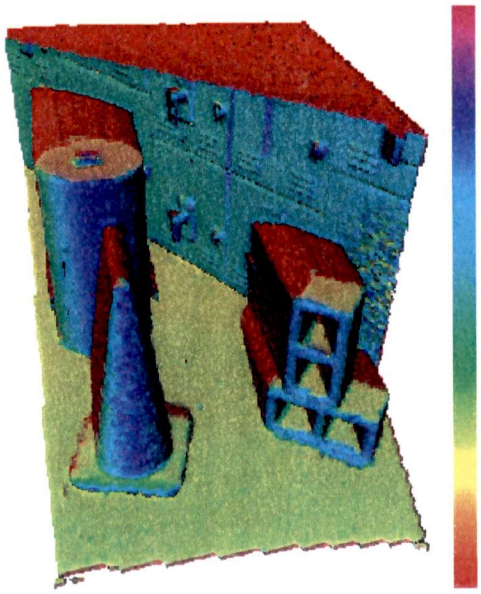


Figure 5.27: *The slope-normal confidence metric used with one 'cone' image from the Coleman scanner. Note that region that were occluded in all three views has a low confidence (red).*

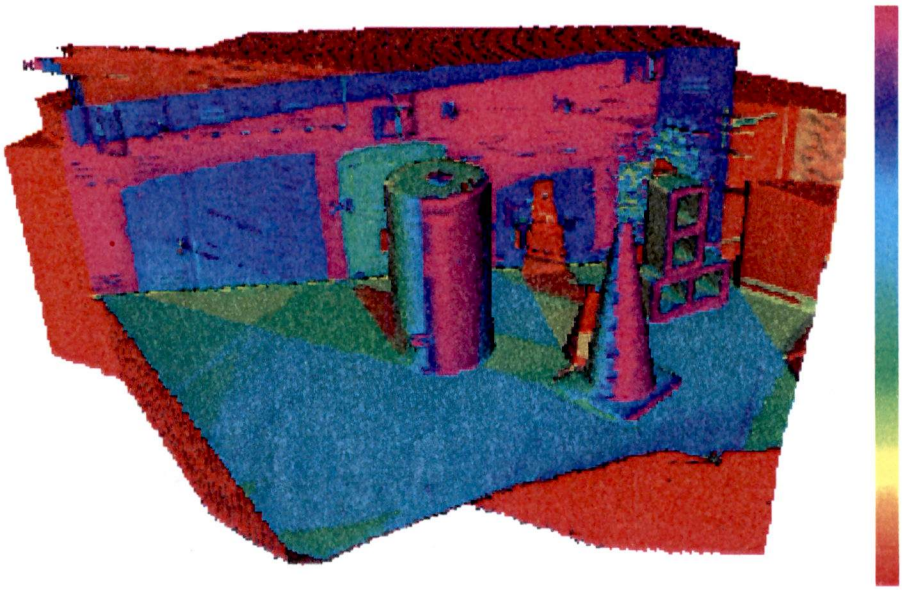


Figure 5.28: *The reconstruction with three views color coded with confidence created with the slope-normal method. Note that regions that were occluded have a low confidence (red).*

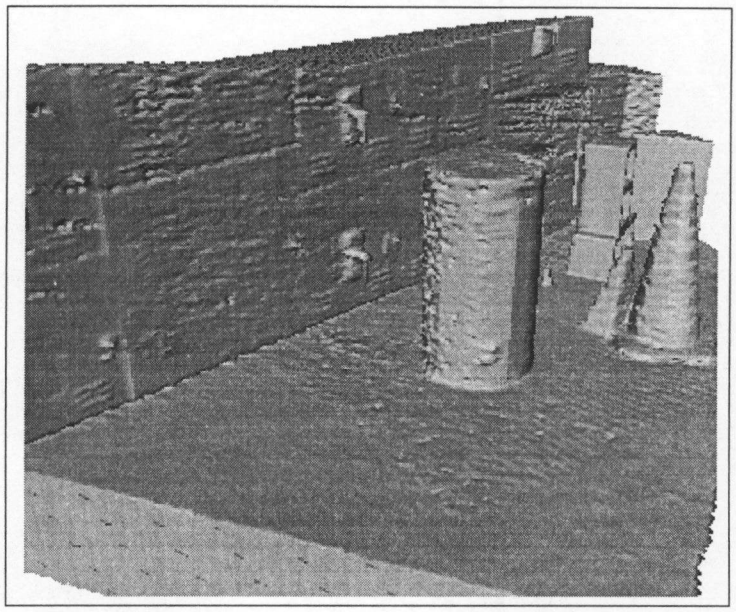
5.2.3 Pixel-by-Pixel Noise Estimate

All the results that we showed up to now have used the same noise estimate and hence the same certainty function throughout an entire range image. However, there are times when one has data that provides information about the relative noise level of individual pixels in an image. For example, the noise in many range scanners is dependent on surface properties such as reflectivity or color. If these properties are known, it is desirable to use the additional information to get a better reconstruction. Pixel-by-pixel noise information can be used by storing a *noise image* that contains the noise data in each pixel. When the voxel is projected onto the range image to obtain a range value, the corresponding noise value is also found. Then we calculate the certainty function as before.

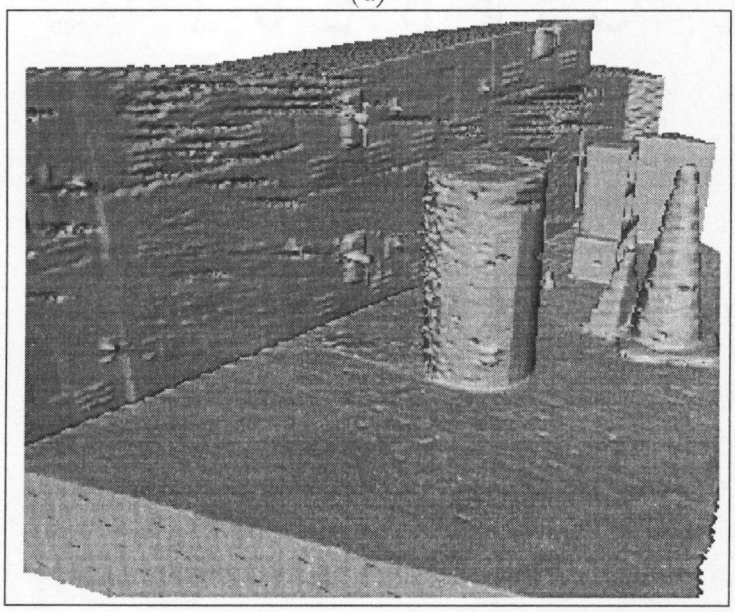
An example where pixel-by-pixel noise information is available is the *quality value* that the Coleman scanner provides with its range data. The quality value is a number based on the intensity of the laser returned from an object. The higher the number, the greater the return, and presumably, the better the accuracy.

To take advantage of this quality value, we need to convert it to a noise estimate. We use a rule-based approach to do this. If the quality value is above a certain value we assume it is an accurate reading and set it to a small constant noise value. If it is below a certain threshold, we assume it is an outlier. In this case we want to ignore the pixel; we do this by replacing the pixel with an average of its neighbors and setting the noise level to a high value. If the quality value is between the upper and lower threshold, we linearly interpolate between the maximum and minimum noise levels to get our noise value.

Figure (5.29) shows a comparison of a reconstruction of the cone scene using the pixel-by-pixel noise estimate (5.29a) with a reconstruction using only image-wide noise information (5.29b). Figure (5.29b) has many tiny holes caused by outliers in the noisy second image even though we set its noise level to be five times greater than the other



(a)



(b)

Figure 5.29: *A comparison of two types of noise estimates. (a) uses a pixel-by-by estimate, while (b) uses an image wide noise estimate.*

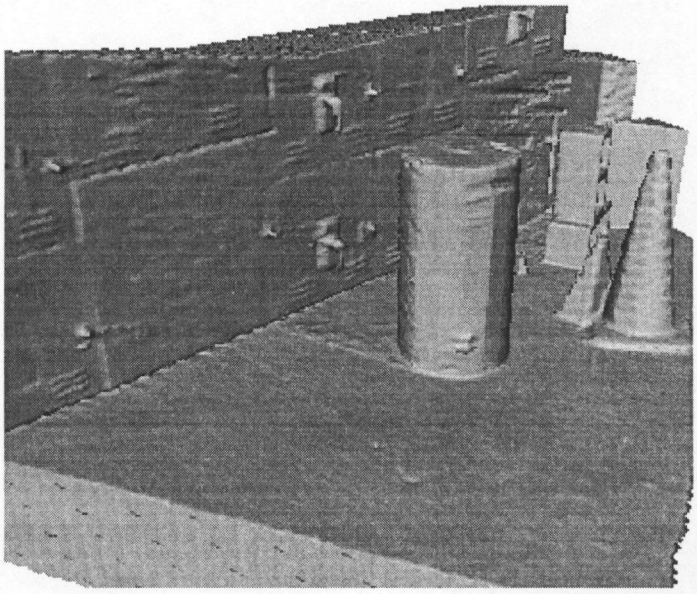


Figure 5.30: *For this reconstruction we median filtered the second image and set its noise estimate to five times the noise estimate of the other two images*

two. Figure (5.29a) uses the quality value to eliminate many of the holes, although much scarring still remains.

This example shows an improved reconstruction using a pixel-by-pixel noise estimate instead of an image-wide estimate. However, recall that in a previous example we used a median filter on the second image before the reconstruction. A close-up of the region of interest is shown in figure (5.30). For this reconstruction we median filtered the second image and set its noise estimate to five times the noise estimate of the other two images. This reconstruction appears smoother than the one in figure (5.29a). However the processing done to the images to create (5.29a) is done automatically, while the user must manually select which images to median filter for (5.30). This trade-off may make using a pixel-by-pixel noise estimate attractive for applications, such as robotic exploration, that needs to be done automatically.

CHAPTER 6

Conclusions

This thesis has presented research in volumetric modeling techniques for 3-D reconstruction. It makes two main contributions. The first is a confidence metric to compliment the occupancy grid method. We define the confidence metric to be similar to a sensitivity analysis; it gives a measure of how sensitive the probability values are to changes in some of the occupancy grid parameters. This is especially useful for robot navigation tasks such as path planning. The operator can ensure that the robot takes routes through areas that have been adequately mapped.

The second contribution is a new volumetric modeling technique that has many of the advantages of the occupancy grid method, but borrows from other volumetric modeling techniques that are more targeted toward range images. Like the occupancy grid method, our technique has a statistical foundation and uses the super-Bayesian combination formula to merge multiple range images. This allows our method to create a certainty function whose shape can be tailored to model the error characteristics of the sensor.

Other non-binary volumetric methods typically use information from only points on the object's surface and ignore background points. Our method (and the occupancy grid method) uses background points to carve away free space around an object. This is especially useful when working with data with large errors; carving away free space can help remove spikes on the surface. Our method labels each point in the grid as either outside or inside; the surface extraction will create a continuous surface between the two. Some other methods (sometimes intentionally) leave holes where there is no sensed data.

Our method has several similarities to the occupancy grid method, but it overcomes

several of its drawbacks. We treat the range image as a whole instead of point-by-point. This avoids the interference effect that we noticed with the occupancy grid. It also makes our method much faster since we do not have calculate a 3-D probability-of-occupancy function for each point.

We have tested our volumetric approach on several sets of real and synthetic range images. The synthetic data sets show that our method correctly reconstructs objects from range data. The reconstructed models are good representations of the original objects. However there are some artifacts due to edge effects in the range images. The real data sets demonstrate that our approach is applicable to many real applications that other volumetric methods may not be. Many volumetric techniques have been demonstrated using data from structured light imaging systems, while we show ours using a time-of-flight scanner. Structured light systems provide range images with less noise, but the size of volume that they can scan is limited. Time-of-flight scanners, similar to the Perceptron range finder we used, are more applicable for situations in which larger areas need to be scanned. These range finders typically have greater noise; other volumetric methods have not been demonstrated to work with these levels of noise.

We have also investigated several confidence measures for our volumetric modeling technique. The number-of-times method is useful for range data, such as that from the Perceptron scanner, that has significant amounts of noise. The slope-based method is valuable in situations where each range image has a different level of noise. The slope-normal method is applicable for low noise range images, such as those from the Coleman scanner. This method uses an estimate of the surface normal to create a more intuitive confidence that decreases in regions tilted away from the scanner. All three of these methods do a good job of highlighting regions that have not been sensed. This is important for planning additional sensor readings.

The work described in this thesis represents the foundation of a 3-D volumetric mod-

eling technique. However, there are still several areas for possible improvement. This method could be used to model larger scenes, but our current approach would create too many triangles display in a reasonable amount of time. One solution would be to apply a mesh reduction algorithm to the final model. As with most volumetric approaches, the time and memory requirements of our technique increases rapidly with grid size. If large grids are needed for an application, one would have to develop time and memory conservation techniques; since this approach can be run on the grid layer-by-layer, one could implement the algorithm in parallel.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America. A, Optics and image science* **4**, pp. 629–642, 1987.

[2] M. W. Walker, L. Shao, and R. A. Voltz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: Image Understanding* **54**, pp. 358–367, 1991.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**, pp. 698–700, 1987.

[4] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, February 1992.

[5] M. Soucy and D. Laurendeau, "A general surface approach to the integration of a set of range views," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(4), pp. 344–358, 1995.

[6] M. Soucy, G. Godin, R. Baribeau, F. Blais, and M. Rioux, "Sensors and algorithms for the construction of digital 3-d colour models of real objects," in *Proceedings of the 1996 IEEE International Conference on Image Processing*, 1996.

[7] Y. Cheng and R. L. Kashyap, "Recursive Fusion Operators: Desirable Properties and Illustrations," in *Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzalez, eds., pp. 245–265, Academic Press, Inc, 1992.

[8] Y. Chen and G. Medioni, "Description of complex objects from multiple range images using an inflating balloon model," *Computer Vision and Image Understanding* **61**(3), pp. 325–396, 1995.

[9] F. P. Ferrie, J. Lagarde, and P. Whaite, "Recovery of volumetric object descriptions from laser rangefinder images," in *Computer vision–ECCV 90 : First European Conference on Computer*, pp. 387–396, Springer-Verlag, 1990.

[10] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Computer Graphics Proceedings, SIGGRAPH*, 1994.

[11] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer* **22**, pp. 46–57, June 1989.

[12] H. P. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar," in *Proceedings of the 1885 IEEE International Conference on Robotics and Automation*, pp. 116–121, March 1985.

[13] L. 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*, vol. 2, pp. 727–733, April 1988.

[14] A. Elfes and L. Matthies, "Sensor Integration for Robot Navigation: Combining Sonar and Range Data in a Grid-Based Representation," in *Proceedings of the 26th IEEE Conference on Decision and Control*, vol. 3, pp. 1802–1807, December 1987.

[15] H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *AI Magazine* **9**, pp. 61–77, Summer 1988.

[16] J. H. Lim and D. W. Cho, "Specular Reflection Probability in the Certainty Grid Representation," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control* **116**, pp. 512–520, September 1994.

[17] P. G. Auran and K. Malvig, "Realtime Extraction of Connected Components in 3D Sonar Range Images." Submitted to CVPR'96 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[18] P. G. Auran and K. Malvig, "Clustering and Feature Extraction in a 3D Real-Time Echo Management Framework." Accepted for presentation at AUV '96.

[19] P. G. Auran and K. Malvig, "3D Sonar Based Sensing for AUVS: Realtime Experiments within the Uncertainty Grid Framework." Accepted for presentation at IFAC'96.

[20] P. G. Auran and O. Silven, "Ideas for Underwater 3D Sonar Range Sensing and Environmental Modelling," in *Proceedings of the IFAC CAMS '95 (Control Applications in Marine Systems) Workshop*, 1995.

[21] P. G. Auran and O. Silven, "Underwater Sonar Range Sensing and 3D Image-Formation," *Control Engineering Practice* 4, pp. 393–400, March 1996.

[22] D. Jung and K. K. Gupta, "Spatial Occupancy Recovery form a Multiple-View Range Imaging System for Path Planning Applications," in *Proceedings of the International Conference on Cybernetics and Society*, vol. 1, pp. 472–477, 1995.

[23] A. Stopp and T. Riethmuller, "Fast Reactive Path Planning by 2D and 3D Multi-Layer Spatial Grids for Mobile Robot Navigation," in *Proceedings of the 1995 IEEE International Symposium on Intelligent Control*, pp. 545–550, 1995.

[24] F. Vacherand, "Fast Local Path Planner in Certainty Grid," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2132–2137, May 1994.

[25] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile Robots," *IEEE Transactions on Systems, Man, Cybernetics* **19**, pp. 1179–1187, September/October 1989.

[26] M. C. Martin and H. P. Moravec, "Robot Evidence Grids," Tech. Rep. CMU-RI-TR-96-06, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1996.

[27] B. Schiele and J. L. Crowley, "A Comparision of Position Estimation Techniques Using Occupancy Grids," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1628–1634, May 1994.

[28] L. Moreno, M. A. Salichs, and D. Gachet, "Fusion of Proximity Data in Certainty Grids," *Parallel and Distributed Computing in Engineering systems: Proceedings of the IMACS/IFAC International Symposium on Parallel and Distributed Computing in Engineering Systems* , pp. 269–274, 1992.

[29] A. P. Tirumalai, B. G. Schunck, and R. C. Jain, "Evidential Reasoning for Building Environment Maps," *IEEE Transactions on Systems, Man, and Cybernetics* **25**, pp. 10–20, January 1995.

[30] E. A. Puente, L. Moreno, M. A. Salichs, and D. Gachet, "Analysis of Data Fusion Methods in Certainty Grids Application to Collision Danger Monitoring," in *Proceedings IECON '91. 1991 International Conference on Industrial Electronics, Control and Instrmentation*, vol. 2, pp. 1133–1137, November 1991.

[31] J. Zurada and J. H. Graham, "Sensory Integration in a Neural-Network-Based Robot Safety System," *International Journal of Human Factors in Manufacturing* **5**, pp. 325–340, Summer 1995.

[32] M. Poloni, G. Ulivi, and M. Vendittelli, "Fuzzy Logic and Autonomous Vehicle: Experiments in Ultrasonic Vision," *Fuzzy Sets and Systems (Netherlands)* **69**, pp. 15–27, January 1995.

[33] G. Oriolo, G. Ulivi, and M. Vendittelli, "Fuzzy maps: A new tool for mobile robot perception and planning," *Journal of Robotic Systems* **14**(1), pp. 179–197, 1997.

[34] V. Santos, J. G. M. Gonclaves, and F. Vaz, "Perception Maps for the Local Navigation if a Mobile Robot: a Neural Network Approach," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2193–2198, May 1993.

[35] Y. Sakaguchi, H. Kato, K. Sato, and S. Inokuchi, "Acquistion of Entire Surface Data Based on Fusion of Range Data," *IEICE Transactions* **E74**, pp. 3417–3422, October 1991.

[36] A. Croteau, D. Laurendeau, and J. Lessard, "Building a space occupancy model for a power-line maintenance robot using a range data sensor," in *Intelligent Robots and Computer Vision X : Algorithms and Techniques*, 1991.

[37] G. H. Tarbox and S. N. Gottschlich, "Ivis: An integrated volumetric inspection system," *Computer Vision and Image Understanding* **61**(3), pp. 430–444, 1995.

[38] G. H. Tarbox and S. N. Gottschlich, "Planning for complete sensor coverage in inspection," *Computer Vision and Image Understanding* **61**(1), pp. 84–111, 1995.

[39] T. Wada, T. Nakamura, and K. Suzuki, "A Robust Object Modeling System Using a Range Finder," in *3rd International Conference in Central Europe*, vol. 2, pp. 329–337, 1995.

[40] A. Li and G. Crebbin, "Octree encoding of objects from range images," *Pattern Recognition* **27**(5), pp. 727–739, 1994.

[41] C. H. Chien, Y. B. Sim, and J. K. Aggarwal, "Generation of volume/surface octree from range data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1988.

[42] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images." To appear in SIGGRAPH 1996, Available from http://www-graphics.stanford.edu /papers/volrange/.

[43] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Computer Graphics* **26**(2), pp. 71–78, 1992.

[44] A. Hilton, "On Reliable Surface Reconstruction from Multiple Range Images." In VSSP-TR-5-95. ftp://ftp.ee.surrey.ac.uk/pub/vision/papers /hilton-vssp-tr-5-95.ps.Z, 1995.

[45] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt, "Reliable surface reconstruction from multiple range images," in *ECCV*, Springer-Verlag, 1996.

[46] D. L. Elsner, "Confidence metric for the occupancy grid framework." University of Tennessee, Knoxville, Technical Report #TR-EE-1101-1996, 1996.

[47] A. Elfes, *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation.*
PhD thesis, Carnegie Mellon Unverisity, Pittsburgh, PA, May 1989.

[48] P. E. Lehner, K. B. Laskey, and D. Dubois, "An Introduction to Issues in Higher Order Uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **26**, pp. 289–293, May 1996.

[49] A. Mosleh and V. M. Bier, "Uncertainty About Probability: A Reconciliation with the Subjectivist Viewpoint," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **26**, pp. 303–310, May 1996.

[50] R. E. Neapolitan, "Is Higher-Order Uncertainty Needed?," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **26**, pp. 294–302, May 1996.

[51] R. D. Klafter, T. A. Chmielewski, and M. Neqin, *Robotic Engineering An Integrated Approach*, Prentice Hall, 1989.

[52] B. Chase, "Calibration of scanning laser range cameras with applications for machine vision," Master's thesis, University of Tennessee, Knoxville, TN, Unpublished, expected 1998.

[53] P. J. Besl, *Surfaces in Range Image Understanding*, Springer-Verlag, 1988.

[54] Y. Cheng and R. L. Kashyap, "Recursive fusion operators: Desirable properties and illustrations," in *Data Fusion in Robotics and Machine Intelligence*, M. A. Abidi and R. C. Gonzalez, eds., ch. 6, pp. 245–265, Academic Press, Inc., 1992.

[55] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *Computer Graphic* **21**(4), pp. 163–169, 1987.

[56] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis Machine Intelligence* **12**, pp. 629–639, 1990.

[57] J. Wagner, "Appendix: Computation of Probability of Occupancy for Cells in the Point Bundle." Unpublished, Prepared for Mechanical Technology Incorporated.

[58] R. D. Luce and H. Raiffa, *Games and Decisions*, Wiley, New York, 1957.

[59] H. E. Kyburg, Jr. and M. Pittarelli, "Set-Based Bayesianism," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **26**, pp. 324–339, May 1996.

[60] D. Dubois, H. Prade, and P. Smets, "Representing Partial Ignorance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **26**, pp. 361–377, May 1996.

[61] B. de Finetti, *Probability, Statistics, and Induction: The Art of Guessing*, Wiley, New York, 1972.

APPENDICES

APPENDIX A

Derivation of Variance Equations

This appendix provides the details of the derivation of the variance equations discussed in chapter 3. Recall that we are considering the following four parameters:

b - prior distribution

$\sigma(r)$ - standard deviation in the range direction

$\sigma(\theta)$ - standard deviation in the θ direction

$\sigma(\phi)$ - standard deviation in the ϕ direction

The variance for a single sensor reading is estimated similarly to a sensitivity analysis using the equation

$$V_1 = \frac{\partial P_1}{\partial b}\Delta b + \frac{\partial P_1}{\partial \sigma(r)}\Delta\sigma(r) + \frac{\partial P_1}{\partial \sigma(\theta)}\Delta\sigma(\theta) + \frac{\partial P_1}{\partial \sigma(\phi)}\Delta\sigma(\phi), \tag{A.1}$$

where V_1 is the estimated variance of the probability. P_1 is probability of occupancy obtained using the occupancy grid formulas. The values Δb, $\Delta\sigma(r)$, $\Delta\sigma(\theta)$, and $\Delta\sigma(\phi)$ are estimates of the variance of the parameters b, $\sigma(r)$, $\sigma(\theta)$ and $\sigma(\phi)$ respectively. The values $\Delta\sigma(r)$, $\Delta\sigma(\theta)$ and $\Delta\sigma(\phi)$ reflect the accuracy at which the variance of the sensor is known.

Before we start the derivation we want to give the definition of the term *point bundle* which is used throughout the remainder of this appendix. Since the sensor error is modeled with a Gaussian, the occupancy grid created by a single sensor reading theoretically

extends throughout the grid. However, to limit the number of calculations, each sensor reading is assumed to influence a small region around it. This region is called the point bundle and its dimensions are based on the variance of the sensor model used.

This section uses the notation of [57]. Here is a brief description of the variables:

r - The range value returned from the sensor.

j, n, v - x, y, z coordinates. Used to indicate the location of the sensor reading.

r_{jnv} - Indicates that the sensor reading was returned from the cell located at (j, n, v) in the occupancy grid.

S - A cell in the occupancy grid.

i, m, u - x, y, z coordinates. Used to indicate the location of cell for which we are calculating the probability of occupancy.

S_{imu} - Indicates that a cell for which we are calculating the probability of occupancy is located at (i, m, u) in the occupancy grid.

k, q, w - x, y, z coordinates. Used to indicate a location within the point bundle.

S_{kqw} - A cell located at (k, q, w), which is within the point bundle.

emp - Abbreviation for empty. Usually used to indicate that a cell is assumed to be empty.

occ - Abbreviation for occupied. Usually used to indicate that a cell is assumed to be occupied.

$C(\cdot)$ - A weighting function that is based on probability of the configuration in which the cells are arranged.

$C_{kqw}(\cdot)$ - The weighting function at location (k, q, w).

Δ - The cell size.

θ_0 - Defined as $\tan\theta_0 = 1/2^i$. Used as an angular step size.

$\Re, \delta$ - A region with the dimensions of $\Re \times \delta \times 1$ that is assumed to be occupied or empty during the calculation of the probability values. This region is moved throughout the point bundle region during the calculation.

$\sum_{k,q,w}$ - A summation throughout the point bundle.

From equation (A.1) it can be seen that we now have to find, $\frac{\partial P_1}{\partial b}$, $\frac{\partial P_1}{\partial \sigma(r)}$, $\frac{\partial P_1}{\partial \sigma(\theta)}$, and $\frac{\partial P_1}{\partial \sigma(\phi)}$. We will start with $\frac{\partial P_1}{\partial \sigma(r)}$.

The formula to calculate P_1 for a single sensor reading is

$$P(S = occ|r) = \frac{P(r|S = occ)P(S = occ)}{P(r|S = occ)P(S = occ) + P(r|S = emp)P(S = emp)}. \tag{A.2}$$

For a prior assumption of 1/2, $P(S = occ) = P(S = emp) = 1/2$, and the equation simplifies to

$$P(S = occ|r) = \frac{P(r|S = occ)}{P(r|S = occ) + P(r|S = emp)}. \tag{A.3}$$

We need to calculate the derivative of this equation with respect to $\sigma(r)$. In the three dimensional case,

$$\begin{aligned} P(r|S = occ) &= P(r_{jnv}|S_{imu} = occ) \\ &= \sum_{w,q,k} P(r_{jnv}|S_{kqw} = occ)C_{kqw}(S_{imu} = occ) \end{aligned} \tag{A.4}$$

and

$$
\begin{aligned}
P(r/S = emp) &= P(r_{jnv}|S_{imu} = emp) \\
&= \sum_{w,q,k} P(r_{jnv}|S_{kqw} = occ) C_{kqw}(S_{imu} = emp).
\end{aligned} \tag{A.5}
$$

Note that the $P(r_{jnv}|S_{imu} = emp)$ is a function of $\sigma(r)$ while the $C_{kqw}(S_{imu} = occ)$ and $C_{kqw}(S_{imu} = emp)$ are not. For simplicity, let us make the following substitutions:

$$
\begin{aligned}
P_{occ}(\sigma_r) &= P(r_{jnv}|S_{kqw} = occ) \\
C_o &= C_{kqw}(S_{imu} = occ) \\
C_e &= C_{kqw}(S_{imu} = emp)
\end{aligned}
$$

These are also implicitly functions of w, q, and k so they cannot be pulled out of the summations later. Taking the necessary derivative yields

$$
\begin{aligned}
\frac{\partial}{\partial\sigma(r)} P(S = occ|r) = &\left[\left(\sum_{w,q,k} \frac{\partial}{\partial\sigma(r)} P_{occ}(\sigma_r) C_o\right)\left(\sum_{w,q,k} P_{occ}(\sigma_r) C_e\right) - \right. \\
&\left.\left(\sum_{w,q,k} P_{occ}(\sigma_r) C_o\right)\left(\sum_{w,q,k} \frac{\partial}{\partial\sigma(r)} P_{occ}(\sigma_r) C_e\right)\right] \Big/ \\
&\left(\left[\sum_{w,q,k} P_{occ}(\sigma_r) C_o + \sum_{w,q,k} P_{occ}(\sigma_r) C_e\right]^2\right).
\end{aligned} \tag{A.6}
$$

This is made up of four terms,

$$
\sum_{w,q,k} P_{occ}(\sigma_r) C_o, \tag{A.7}
$$

$$
\sum_{w,q,k} P_{occ}(\sigma_r) C_e, \tag{A.8}
$$

$$\sum_{w,q,k} \frac{\partial}{\partial\sigma(r)} P_{occ}(\sigma_r) C_o, \tag{A.9}$$

$$\sum_{w,q,k} \frac{\partial}{\partial\sigma(r)} P_{occ}(\sigma_r) C_e. \tag{A.10}$$

The first two are the same ones used to calculate the occupancy grid. The last two are new terms. However the C_o and the C_e are the same as used in the occupancy grid formulation. To calculate equations (A.9) and (A.10), we have to find $\frac{\partial}{\partial\sigma(r)} P_{occ}(\sigma_r)$. For the 3D case the $P_{occ}(\sigma_r)$ is given in [57].

$$\frac{1}{8}\left[\text{erf}\left(\frac{j-k+1/2}{\sqrt{2}\sigma(r)/\Delta}\right) - \text{erf}\left(\frac{j-k-1/2}{\sqrt{2}\sigma(r)/\Delta}\right)\right] \left[\text{erf}\left(\frac{n-q+1/2}{\sqrt{2}\sigma(\theta)/\theta_0}\right) - \text{erf}\left(\frac{n-q-1/2}{\sqrt{2}\sigma(\theta)/\theta_0}\right)\right]$$

$$\left[\text{erf}\left(\frac{v-w+1/2}{\sqrt{2}\sigma(\phi)/\theta_0}\right) - \text{erf}\left(\frac{v-w-1/2}{\sqrt{2}\sigma(\phi)/\theta_0}\right)\right] \tag{A.11}$$

Taking the necessary derivative results in

$$\frac{\sqrt{2}\Delta}{16\sqrt{\pi}\sigma^2(r)}\left[\exp\left(-\frac{1}{8}\frac{(2j-2k+1)^2\Delta^2}{\sigma^2(r)}\right)(2j-2k+1) - \right.$$
$$\left.\exp\left(-\frac{1}{8}\frac{(2j-2k-1)^2\Delta^2}{\sigma^2(r)}\right)(2j-2k-1)\right]$$
$$\left[\text{erf}\left(\frac{\sqrt{2}}{4}\frac{(2n-2q+1)\theta_0}{\sigma(\theta)}\right) - \text{erf}\left(\frac{\sqrt{2}}{4}\frac{(2n-2q-1)\theta_0}{\sigma(\theta)}\right)\right.$$
$$\left.\text{erf}\left(\frac{\sqrt{2}}{4}\frac{(-2v+2w-1)\theta_0}{\sigma(\phi)}\right) - \text{erf}\left(\frac{\sqrt{2}}{4}\frac{(-2v+2w+1)\theta_0}{\sigma(\phi)}\right)\right]. \tag{A.12}$$

Now we have enough information to find $\frac{\partial}{\partial\sigma(r)}$. The procedure for calculating $\frac{\partial}{\partial\sigma(\theta)}$ and $\frac{\partial}{\partial\sigma(\phi)}$ is the same. Finding the last derivative, $\frac{\partial P_1}{\partial b}$, is slightly different. Since b is the prior, we cannot eliminate $P(S = occ)$ and $P(S = emp)$ from equation (A.2).

Clearly this is true because,

$$P(S = occ) = b,$$

and

$$P(S = emp) = 1 - b.$$

Equation (A.2) becomes,

$$P(S = occ|r) = \frac{P(r|S = occ)b}{P(r|S = occ)b + P(r|S = emp)(1 - b)}. \tag{A.13}$$

$P(r|S = occ)$ and $P(r|S = emp)$ remain the same as equations (A.5) and (A.6) respectively. The $P(r_{jnv}|S_{kqw} = occ)$ term is not a function of b. However the $C_{kqw}(S_{imu} = occ)$ and $C_{kqw}(S_{imu} = emp)$ are functions of b. For simplicity, let us make the following substitutions:

$$P_{occ} = P(r_{jnv}|S_{kqw} = occ)$$
$$C_o(b) = C_{kqw}(S_{imu} = occ)$$
$$C_e(b) = C_{kqw}(S_{imu} = emp)$$

These are implicitly functions of k, q, and w so they cannot be pulled out of the summations later. Equation (A.14) shows the result of taking the necessary derivatives.

$$\frac{\partial}{\partial b}P(S = occ|r) = \left[(b - b^2)\left[\left(\sum_{w,q,k} P_{occ}\frac{\partial}{\partial b}C_o(b)\right)\left(\sum_{w,q,k} P_{occ}C_e(b)\right) - \right.\right.$$
$$\left.\left.\left(\sum_{w,q,k} P_{occ}C_o(b)\right)\left(\sum_{w,q,k} P_{occ}\frac{\partial}{\partial b}C_e(b)\right)\right] + \left(\sum_{w,q,k} P_{occ}C_o(b)\right)\left(\sum_{w,q,k} P_{occ}C_e(b)\right)\right]$$
$$\Big/\left[b\left(\sum_{w,q,k} P_{occ}C_o(b)\right) + (1 - b)\left(\sum_{w,q,k} P_{occ}C_e(b)\right)\right]^2. \tag{A.14}$$

Table A.1: *The values of $C_{kqw}(S_{imu} = occ)$ and $C_{kqw}(S_{imu} = emp)$ in five regions.*

k	q	w	$C_{kqw}(S_{imu} = occ)$	$C_{kqw}(S_{imu} = emp)$
$< i$	all	all	$1/2^k$	$1/2^k$
$= i$	In $\Re$	In δ	$1/2^{k-1}$	0
$= i$	Not in $\Re$	Not in δ	$1/2^k$	$1/2^k$
$> i$	In $\Re$	In δ	0	$1/2^{k-1}$
$> i$	Not in $\Re$	Not in δ	$1/2^k$	$1/2^k$

Source: J. Wagner, "Appendix: Computation of Probability of Occupancy for Cells in the Point Bundle." Internal document for Mechanical Technology Incorporated.

Note that the terms

$$\sum_{w,q,k} P_{occ} C_o(b)$$

and

$$\sum_{w,q,k} P_{occ} C_e(b)$$

are the terms that appear in the occupancy grid equations. The new terms include either $\frac{\partial}{\partial b} C_o(b)$ or $\frac{\partial}{\partial b} C_e(b)$. These are found by taking the derivative of $C_{kqw}(S_{imu} = occ)$ and $C_{kqw}(S_{imu} = emp)$ with respect to b. I will start with $\frac{\partial}{\partial b} C_o(b)$. The first step is to explicitly make $C_{kqw}(S_{imu} = occ)$ a function of b. According to table (A.1) adapted from [57] there are five regions in which $C_{kqw}(S_{imu} = occ)$ can be evaluated.

Case 1: $k < i$ for all q and w

For any cell S_{kqw} to cause reading r_{juv}, cell S_{imu} must be occupied and all cells between S_{kqw} and the sensor must be empty. There are $k - 1$ cells between the cell S_{kqw} and the

sensor. The probability of all of them being empty is $(1-b)^{k-1}$. The probability of S_{imu} being occupied is b. So the total probability is

$$b(1-b)^{k-1}. \tag{A.15}$$

Note that if $b = 1/2$, this agrees with table (A.1). The necessary derivative results in equation (A.16).

$$\frac{\partial}{\partial b}[b(1-b)^{k-1}] = -(1-b)^{k-2}bk + (1-b)^{k-2}b + (1-b)^{k-1} \tag{A.16}$$

Case 2: $k = i$; q in $\Re$ and w in δ

For this case S_{kqw} is in a region assumed occupied, so only the requirement that all cells between S_{kqw} and the sensor be empty is needed. The probability of this is

$$(1-b)^{k-1}, \tag{A.17}$$

and the resulting derivative is

$$\frac{\partial}{\partial b}[(1-b)^{k-1}] = -(1-b)^{k-2}(k-1). \tag{A.18}$$

Case 3: $k = i$; q not in $\Re$ or w not in δ

This follows the same argument as case 1 and yields the equations (A.15) and (A.16).

Case 4: $k > i$; q in $\Re$ and w in δ

In this case S_{kqw} in occluded by the cells that are assumed occupied, and cannot affect the cell r_{rnv}.

$$C_{kqw}(S_{imu} = occ) = 0, \tag{A.19}$$

and

$$\frac{\partial}{\partial b}(0) = 0. \tag{A.20}$$

Case 5: $k > i$; q not in $\Re$ or w not in δ

This follows the same logic as case 1 and results in equations (A.15) and (A.16).

Now we have to find $C_{kqw}(S_{imu} = emp)$. Again there are five cases.

Case 1: $k < i$; all q; all w

This follows the same reasoning as case 1 of $C_{kqw}(S_{imu} = occ)$ and yields equations (A.15) and (A.16).

Case 2: $k = i$; q in $\Re$ and w in δ

In this case, S_{kqw} is in a region assumed to be empty. This follows the reasoning of case 4 of $C_{kqw}(S_{imu} = occ)$ and yields equations (A.19) and (A.20).

Case 3: $k = i$; q not in $\Re$ and w not in δ

This is the same type of situation as case 1 and yields the equations (A.15) and (A.16).

Case 4: $k > i$; q in $\Re$ and w in δ

In this case S_{kqw} could have caused r_{jnv} if S_{kqw} is occupied and all the cells between it and the sensor are empty. There are $k - 1$ cells between S_{kqw} and the sensor, but one

of them is in the region assumed to be empty. So the probability of all these cells being empty is

$$(1-b)^{k-2}. \tag{A.21}$$

The total probability is

$$b(1-b)^{k-2}, \tag{A.22}$$

and the required derivative is

$$\frac{\partial}{\partial b}[b(1-b)^{k-2}] = -(1-b)^{k-3}bk + 2(1-b)^{k-3}b + (1-b)^{k-2}. \tag{A.23}$$

Case 5: $k > i$; q not in $\Re$ or w not in δ

This follows the same logic as case 1 and results in equations (A.15) and (A.16).

Note that equations (A.15),(A.17),(A.19) and (A.22) all reduce to the values in table (A.1) if $b = 1/2$. Now equations (A.16),(A.18), (A.20) and (A.23) can be used to evaluate equation (A.14), and $\frac{\partial}{\partial b}$ can be found.

Finally we have enough information to find V_1 in equation (A.1).

APPENDIX B

Confidence Metric and Higher-Order Uncertainty

The confidence metric that we proposed in chapter 3 is based on the concept of higher-order uncertainty. Higher-order uncertainty is a probability of a probability. That is, it is an assessment of the our certainty that a probability value is correct. This is exactly what our confidence metric is; it is our confidence that the probability of occupancy contained in the occupancy grid is correct. Hence, higher-order uncertainty seems ideal for our application. However, researchers in the field of probability disagree about the validity of higher-order uncertainty [48, 49, 50]. This appendix give a brief overview of the debate and explains why we believe higher-order uncertainty if appropriate for our application.

There are two fundamental theories of probability. They are known as the objective view and the subjective view. The following statements taken from [48] will be useful in showing the differences between the two viewpoints:

S1 - "The probability that the coin will land heads is .5"

S2 - "The probability that there is life on Venus is less than .0001."

S3 - "The probability that it will rain in two weeks is .2."

S4 - "The probability that there is life in other galaxies is .5."

The objective view is based on the concept of relative frequency which is defined by

$$f(E) = \lim_{n \to \infty} \frac{S_n(E)}{n}, \tag{B.1}$$

where E is one possible outcome of an experiment that can be performed an infinite number of times. $S_n(E)$ is the number of times E occurs in the first n trials. Hence the relative frequency, $f(E)$, is the percentage of the experiments that result in E as the number of trials approach infinity [50]. Objectivists argue that it is only valid to define probabilities for events that can (in theory) be repeated an infinite number of times and a relative frequency determined. Statement S1 above satisfies this requirement. With this definition, probabilities reflect an objective reality. In S1, for example, the probability is measuring a property of the coin.

The subjective view, often called Bayesianism [48], finds little value in the relative frequency concept. Subjectivists wish to characterize reality to facilitate decision making. They think of probability as a subjective degree of belief which must obey the probability calculus. By this definition all the statements (S1–S4) are valid. However, unlike objectivists, they feel that probabilities are a property of the person who develops them. In this case, S1 does not reflect a property of the coin, but a property of the person whose opinion it is [49]. Since beliefs are properties of individuals and not object properties of nature, Bayesianism does not require that each individual have the same probability for every event. However, the probability assessments of each individual has to be consistent with the information he/she has. Individual differences in beliefs result from different states of knowledge and differing ability to interpret the information.

One important difference between the subjective and objective views is which elements are required to obey the probability calculus. In the objective view, only statements such as S1 that reflect objective properties of reality are required to obey the probability axioms. Statements of opinion or subjective degrees of belief such as S2 and S4 need not conform, and the term probability should not be used for them. In objectivist theory, methods for updating existing probabilities over time need not conform to a coherent probability calculus. Subjectivist theory makes no distinction between statements of objective reality

and statements of subjective degrees of belief. Both are required to obey the probability calculus. Furthermore, methods for changing belief are required to be coherent. Updating is done with Bayes rule of conditioning [48].

Now we come to the issue of higher-order uncertainty. Objectivists view probability as a property of reality. This allows no room for uncertainty. Each observer will always arrive at the same probability value using the relative frequency approach. The subjective view, however, is not as clear cut. Consider again statements S1 – S4. All of these statements are consistent with the subjective viewpoint, however some seem more certain than others. S1 is a classic example in probability. There is little doubt that this statement is true for a fair coin. S2 is a statement of subjective probability. We do not know of a near infinite number of Venus-like planets that we can use to derive this statement from a relative frequency approach. Indeed, this statement reflects the uncertainty of our knowledge of the conditions on Venus and the types of life that may develop, not our estimate of the possibility of life evolving under the conditions we believe are on Venus. However, this statement seems relatively certain. We have enough information about the conditions on Venus and the requirements of life to be reasonably sure that there is no life on Venus.

Statement S3 could be thought of as an objective or a subjective probability. It is objective if the .2 represents the probability of rain on any given day at a given location. It is subjective if it is based on meteorological information. If the objective view is taken, S3 is as certain as S1. In the subjective case, however, it seems rather uncertain. Weather forecasting is notoriously unreliable, especially weeks in advance. S4 is a subjective statement. The .5 value is used to indicate that we have no idea whether life exists in other galaxies. A statement based on such little information seems to be very uncertain. Note that this differs from the subjective view of S3. S3 may be based on mounds of meteorological information; we simply do not have a good enough weather model to use it effectively.

Statements such as S3 and S4 seem to give credence to the argument for higher-order uncertainty in the subjective view of probability. However there are three major arguments that have been made against higher-order uncertainty [49]. They are:

1. Higher-order uncertainty violates the subjectivist theory.
2. Higher-order uncertainty would lead to an infinite hierarchy of probabilities.
3. Higher-order uncertainty can be reduced to first-order uncertainty by restating the problem.

We will consider each of these individually.

B.1 Violation of the Subjective Theory

Subjectivists believe that probabilities are functions of the individual who develops them, not a property of the real world. According to the subjective theory, each individual can determine his beliefs to any accuracy required. These belief values may differ from those of other people for the same event. This is perfectly acceptable as long as each individual's beliefs are consistent with his/her state of knowledge. Since subjectivist think of probabilities as opinions, not events, uncertainty on them is meaningless [49]. Additionally, placing an uncertainty on a probability, would seem to imply that a probability is an estimate of a true state of nature and not a perfectly knowable function of an individual. This is a statement that many subjectivists strongly disagree with.

B.2 Infinite Hierarchy of Probabilities

Another troubling aspect about higher-order uncertainty is the possibility of an infinite hierarchy of probabilities. The argument goes like this. A first-order probability is uncertain and a second-order probability is introduced to quantify this uncertainty. However, the second-order probability is merely a probability just as the first-order probability is. If the first-order probability can be uncertain, the second-order probability can be uncertain as well. This means a third-order probability can be introduced to describe the uncertainty on the second-order probability. This argument can be extended to an infinite number of higher-order uncertainties.

B.3 Reduction to First Order Uncertainties

Some problems may seem to warrant the introduction of higher-order uncertainties, however, if the problem is restated, it can be seen that first-order uncertainties are sufficient. Neapolitan [50] gives several good examples of this.

Of these three arguments, number 1 is the only one that can destroy the validity of higher-order uncertainty. The statement that higher-order uncertainty leads to an infinite hierarchy of probabilities may be true and is undesirable. However, it does not mean that higher-order uncertainty is not valid, it merely explains that it has drawbacks. In our case, we can simply choose to ignore all but the first and second-order uncertainties. The third argument implys that higher-order uncertainty is not needed. However, saying that something is not needed does not imply that it is not valid. Furthermore one is not always free to restate the problem to avoid the use of higher-order uncertainty. The goal of this research is to create a confidence metric that complements an algorithm that has already been formulated and implemented.

Argument 1 is potentially more damaging. The occupancy grid framework is based on the subjective (Bayesian) view of probability. If it could be shown that higher-order uncertainty is meaningless in the Bayesian framework, it would be difficult to justify our confidence metric.

Mosleh and Bier [49] provide a strong argument that certain types of higher-order uncertainty are valid in the subjective view. They argue that uncertainty can be caused by two basic reasons, cognitive imprecision and uncertainty about underlying events. They claim that uncertainty of a probability is consistent with the subjectivist viewpoint if it is due to uncertainty about the underlying events on which the probability is conditioned. However, uncertainty due to cognitive imprecision is not consistent. Let us look at each of these areas in more detail.

B.4 Cognitive Imprecision

Subjectivists feel that an individual can express his/her beliefs to any degree of accuracy required. However, in reality this does not seem to be the case. There seems to be some cognitive limit to accessing beliefs just as there is a limit to the different shades of colors that can be distinguished. For example, I may be asked the following multiple choice question:

How sure are you that the Baltimore Orioles will win the World Series?

a) 1% b) 25% c) 50% d) 75% e) 99%

With this wide separation between probabilities I can quite easily select "b" as the closest to my true belief. Now I am asked to refine my answer with the following choices:

a) 15% b) 20% c) 25% d) 30% e) 35%

This will require more in depth introspection, however, I can select "b" rather easily. The next level of refinement is:

a) 18% b) 19% c) 20% d) 21% e) 22%

This becomes more difficult. I find that the values are so close together that I cannot determine my beliefs to this level of detail. This is partially due to the fact that I have not been following baseball closely lately and I can not evaluate the teams very well. A baseball analyst may be able to make this distinction, but even he/she would have trouble with the next level of refinement:

a) 19.8% b) 19.9% c) 20.0% d) 20.1% e) 20.2%

There is a limit to resolution that people are able to express their beliefs, and this extends to assessments of preferences. Consider the example given by Luce and Raiffa [58]. A person who is indifferent between a trip to Rome and a trip to Paris may also claim to be indifferent between a trip to Rome and a trip to Paris plus $20. However the same person would almost certainly prefer a trip to Paris plus $20 than just a trip to Paris. This is because the person cannot distinguish between the value of a trip to Rome and a trip to Paris to the precision of $20 [49].

Cognitive imprecision as described above, seems to be intuitively valid. However, it unquestionably contradicts the assumptions of the Bayesian framework. Namely the assumption that individuals can determine their beliefs to any precision that they require. Therefore it does not seem reasonable to attempt to model this type of uncertainty in a framework that expressly forbids it. Instead, a methodology which relaxes the infinite precision constraint would be used. This includes such methods as set-based Bayesianism [59] and the Dempster-Shafer theory of evidence [60].

B.5 Uncertainty of Underlying Conditions

Uncertainty about a probability is often caused by uncertainty about that underlying events, propositions, or hypotheses on which the probability is conditioned [49]. Consider the probability of event E given the information X, $P(E|X)$. If we are uncertain of the given information, X is a random variable. Then $P(E|X)$ is a function of a random variable and must be a random variable itself. Therefore if we are uncertain of X, we are also uncertain of $P(E|X)$.

Consider the example given in [49]. We have two coins, one that has both heads and the other has both tails. We select one of the two coins at random and flip it. We want to determine the probability that we get heads. The answer is intuitively 0.5. In this case, whether we get heads or tails is determined by which coin we select and not the result of the flip. The probability of selecting the coin with two heads is 0.5; therefore the probability of heads is 0.5.

Let X be the unknown condition of the coin. $X = HH$ if the coin has two heads. $X = TT$ if the coin has two tails. The conditional probability of getting a heads is

$$P(head|X = HH) = 1, \tag{B.2}$$

$$P(head|X = TT) = 0. \tag{B.3}$$

$P(head|X)$ is a random variable with values of either 1 or 0. The probability of each state is

$$P(P(head|X = HH) = 1) = .5. \tag{B.4}$$

$$P(P(head|X = TT) = 0) = .5. \tag{B.5}$$

Again it can be seen that the probability of getting heads is .5.

To quote Mosleh and Bier [49], "In this formulation, we agree with de Finetti [61] that it is inconsistent with the subjective theory of probability to be uncertain about the value of $P(E|X)$ when the conditions X are fully known. Similarly, it is also inconsistent to be uncertain about the expected or unconditional probability of the event in question, $P(E)$, since the unconditional probability can be found by simply taking the expectation of $P(E|X)$ over all possible values of X. However, ...there are situations where use of the unconditional (or expected) probability is incorrect, and the conditional probability is needed. Since the conditions X may not always be known, it must clearly be admissible to be uncertain about the conditional probability $P(E|X)$ in such cases."

This is the view that we adopt for the creation of the confidence metric. It seems intuitively correct that uncertainty about some of the parameters used in the occupancy grid framework will translate to uncertainty about the resulting probability values. In this formulation the confidence on a probability can be thought of as a confidence of the underlying assumptions of the method.

VITA

David Elsner was born in Ames, Iowa in 1968. He lived in central Iowa until 1981 when he moved with his parents to western Nebraska. Mr. Elsner graduated from Hemingford High School in 1987. Later that year he began work toward his BSEE at the University of Nebraska-Lincoln. He completed the degree in December 1991 with a specialization in communications. In the beginning of 1993, David began work with Johnson Engineering Corporation, a NASA contractor, in Houston, Texas. He worked as a Design Engineer, creating hardware for space station and space shuttle trainers. Mr. Elsner enrolled in the MSEE program at the University of Tennessee-Knoxville in 1996 where he is currently working as a Graduate Research Assistant in the Imaging, Robotics and Intelligent Systems Laboratory under the supervision of Dr. R. T. Whitaker and Dr. M. A. Abidi. David expects to graduate in December 1997 with a specialization in image processing and robotic vision.