INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

GEON RECOGNITION USING A MOBILE ROBOT VISION SYSTEM

Romney K.H. Ng

Department of Electrical Engineering
McGill University

October 1998

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Masters of Engineering

Abstract

Geons are simple 3-D geometric primitives proposed by Biederman for object representation. The idea is to segment objects into an arrangement of simple geometric shapes. Objects represented by geons can be identified more easily since the many-to-one mapping of object recognition is reduced to the recognition of a small number of geons. In this project, a vision system is implemented for the search and recognition of geons.

The vision system is composed of a colour camera and a laser range finder, and is mounted on a pan-tilt unit. The entire setup is mounted on a mobile robot. The task of the system can be divided into two phases: The first phase is focus of attention. Our focus of attention algorithm is based on colour. The geons are painted with a pre-determined colour, and the colour camera searches for objects with that colour. Upon detection of a geon, the mobile robot moves towards it in preparation for the next phase.

The second phase is object recognition. With the robot in the proximity of the geon, the laser range finder is used to obtain range data of the geon for recognition. Since only a small number of geons (N = 7) is considered in this project, very sparse range data is sufficient for successful recognition.

Résumé

Les géons sont de simples primitives géométriques 3-D introduites par Biederman pour la représentation d'objets. L'idée est de segmenter les objets en un ensemble de formes géométriques simples. Les objets représentés par des géons peuvent être reconnus plus facilement puisque le problème de reconnaissance qui consiste à trouver un objet particulier parmi un grand nombre d'objets se limite à reconnaître un petit nombre de géons. Dans ce projet, un système de vision est implanté pour la recherche et la reconnaissance de géons.

Le système de vision est composé d'une caméra couleur et d'un laser de profondeur et est monté sur une unité pan-tilt. L'ensemble du montage est placé sur un robot mobile. La tâche du système peut être divisée en deux phases: la première correspond à la détermination du centre d'intérêt. Notre algorithme de détermination du centre d'intérêt est basé sur la couleur. Les géons sont peints avec une couleur prédéterminée et la caméra couleur cherche des objets de cette couleur. Une fois qu'un géon est détecté, le robot mobile se dirige vers lui pour préparer la phase suivante.

La seconde phase est celle de reconnaissance d'objets. Avec le robot à proximité du géon, le laser de profondeur est utilisé pour obtenir des données de profondeur sur le géon pour la reconnaissance. Comme dans ce projet seulement un petit nombre de géon est considéré ($N = 7$), des données de profondeur très espacées sont suffisantes pour que la reconnaissance soit effectuée avec succès.

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards Professor Martin Levine for his invaluable advice and guidance. I would also like to thank Marc Bolduc and Thierry Baron of the Mobile Robotics Laboratory for their numerous technical assistance, Gilbert Soucy and Mourad El-Gamal for providing insight into my rangefinder design, Robert Sim for providing the framework of this project, and Tenzin Shomar and Sandeep Shenoy for their help on proofreading this thesis. Many thanks to Robert Sim and François Bélair for their strong effort to help integrate their robot navigation mechanism with my work. Last but not least, I would like to thank my parents for their unconditional support over the years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction and Motivation

The objective of this thesis is to present an algorithm for the search and recognition of simple 3-D shapes, called geons, based on sparse range data. A vision system was constructed to test our work. The following chapters will describe in detail the implementation of this vision system, as well as the recognition algorithm. This first chapter introduces the use of geons in computer vision, and the goals and motivation for our research.

1. Introduction

Recent advances in computer imaging technology have led to increasing research effort being placed on object recognition. In particular, three-dimensional generic object recognition turns out to be especially interesting and challenging. The term "generic object recognition" refers to the recognition of objects from a domain of a large number of models [**18**], and the distinction between two-dimensional recognition and three-dimensional recognition will be explored in the next chapter. One approach to this problem that has gained considerable interest is the volumetric approach. In this class of methods, complex objects are considered as a concatenation of many simple volumetric parts. The objects are first divided into parts, and then each part is identified. The identities of the individual parts are combined with information on how the parts are grouped to achieve recognition of the entire object. This is

analogous to speech processing, in which phonemes - small, simple units of speech - are first identified, and then their concatenation leads to speech recognition. Biederman's Recognition By Components (RBC) theory [**8, 9**] proposed the use of 36 unique geons as the volumetric primitives to be used in generic 3-D object recognition. The most important difference between geons and other geometric primitives, such as superquadrics, is that geons are described solely by qualitative attributes. Biederman noted that the human observer judges quantitative attributes poorly, and yet can efficiently recognize almost any complex real-world object. It is, therefore, probable that quantitative description is not an important factor in generic object recognition. A more thorough discussion of geons will be presented in the next chapter.

Although distinguishing a moderate number of simple 3-D shapes appears to be a fairly simple problem, it does involve some open issues. One issue of particular importance is the choice of sensor used to gather relevant information. The majority of computer vision methods use ordinary video cameras as an input device to acquire two-dimensional intensity images of the scene. This option is attractive due to the off-the-shelf availability of video cameras. Some research have been done on using line drawings to deduce geon shapes [**6, 7, 17**]. Theoretically, line drawings of objects could be obtained by extracting edge information from 2-D intensity images. However, reliable edge detection is difficult to realize. Poor and uncontrolled lighting conditions often leads to partial failure of edge extraction mechanisms. Moreover, edge information alone sometimes produce ambiguous results.

Since explicit 3-D information can be obtained from a range map, the use of laser rangefinders for object recognition has become increasingly popular. There exist several laser ranging techniques, each with their advantages and drawbacks. For reasons to be discussed later, we have adopted the triangulation technique for the implementation of our laser rangefinder.

This research explores the feasibility of geon recognition based on very sparse range data. In addition, an active vision system was implemented to search for geons in an indoor environment, acquiring a good view of the geon to facilitate recognition.

The ultimate goal is to have the vision system mounted on a mobile robot, which will navigate towards the geon to collect range data for recognition.

2. Motivations and Goals

Rather than attempting to solve the daunting problem of generic 3-D recognition, this research only considers the recognition of individual geons. Raja and Jain [**48**] tackled this problem by acquiring range data of geons and fitting them to superquadrics, then determining the identity of the geons based on superquadric parameters. Wu [**66**] used a global optimization technique to match multiple-view range data with parametric descriptions of geons. Our proposed approach differs from those mentioned above in two important respects. Firstly, it is reasonable to expect relatively little computation for the recognition of simple objects, therefore, instead of using a full range map, we use only six laser stripe profiles for recognition. This will significantly reduce the amount of data processing required. Secondly, the essence of geons is their qualitative nature, and fitting superquadric parameters to them more or less defeats that purpose. Our recognition algorithm will make decisions based on qualitative attributes rather than quantitative parameters.

Using range data to recognize geons is only one of our objectives. Our ultimate goal is to integrate the vision system with a mobile robot, so that geon recognition can become one of the robot's functionalities. This involves additional issues such as focal attention for the search of geons in the environment, and robot navigation to bring the robot near the geons. These issues will be addressed in Chapter 4.

This research has some potentially interesting applications. As an example, an automated front end loader in a warehouse could be made to distinguish simple objects like boxes (cuboids), barrels (cylinders), and cones (tapered cylinders), and manipulate them accordingly. Geons can also act as landmarks to aid mobile robot navigation.

3. Thesis Outline

The following chapters will address different aspects of geon recognition and our approach to tackling this problem. In the next chapter, we will compare 2-D and 3-D methods to object recognition, and review some recent work on the latter. Specifically, we will focus on the increasingly important role of geons in the 3-D vision research community. Chapter 3 describes the hardware, namely the vision system as well as the mobile robot. The details of our rangefinder calibration method are also presented. Chapter 4 addresses the active vision portion of our vision system - the use of colour as the cue for focus of attention, the method to achieve proper placement of laser stripes, and a brief description of the robot navigation algorithm. Chapter 5 describes the recognition module, which involves preprocessing of data and the generation of evidence to facilitate the final decision process. Chapter 6 presents the methods and results of the experiments. The experiments consist of three parts: the first part deals exclusively with simulated data. In the second part real geons are placed in the relative vicinity of the vision system so that recognition can be achieved with camera movements alone. In the third part the vision system is mounted on the mobile robot, which navigates through a controlled indoor environment to search for geons, and attempts to identify them upon detection. Finally, the last chapter summarizes the thesis and presents a critical analysis of our research.

CHAPTER 2

Background

Object recognition with mobile robots often implies more than merely identifying an object in an image. It also involves searching for the object of interest in the environment, and if necessary, approaching it and making adjustments to acquire a better view. In other words, the problem of mobile robot object recognition is composed of two parts:

(i) Active Vision/Focus of Attention (Where is it?)
(ii) Object Recognition (What is it?)

The first section of this chapter addresses the problem of active vision and focus of attention. The second section discusses object recognition, as well as the use of geons for object representation and recognition. We conclude with a brief description of our approach to the experiment.

1. Active Vision

Most research in machine vision involves passively sampled images. Recognition algorithms often operate under idealistic conditions such as controlled lighting, sharply focused images, and a priori knowledge of the objects' locations. While these assumptions may be valid for certain applications (e.g., factory automation), they hardly apply to intelligent, autonomous systems. Human perception, for instance, is

not passive, but active. Our eyes adjust to different levels of illumination and bring scenes into focus, our heads move to search for an object, and we are capable of movement towards the object to obtain a better view.

Bajcsy [**2**] defined active vision as "a study of modelling and controlling strategies applied to the data acquisition process which will depend on the current state of the data interpretation and the goal or task of the process". This is different from the concept of "active sensors", which refers to sensors that transmit signals (e.g., sonar, structured light) into the environment and use the reflected signals to deduce range information. We use the term "active" to denote controlling passive sensors in an active manner, i.e., purposefully changing the sensor's state parameters according to specific sensing strategies.

An observer is called "active" when engaged in some kind of activity whose purpose is to control the geometric parameters of the sensory apparatus [**1**]. Many active vision systems involve the alteration of sensor settings. For example, Krotkov [**35**] presented the Agile camera system which performs stereo ranging. Each of the two cameras has control of focus, zoom, and aperture. The vergence angle between the two cameras can also be controlled. Another class of active vision systems involves mobile robots. With a vision system mounted on a mobile robot, one can often dispense with the need to control sensor settings. Instead, the robot can be steered to desired locations to acquire images. For example, Gvozdjak [**28**] implemented an active object recognition system on a mobile robot, employing a hierarchical recognition algorithm based on a multi-resolution representation of objects and images. First, the robot searches for a coarse shape that matches the low resolution image of the model. If a match is found, the robot moves closer to the object to see whether its finer details also match the higher resolution image of the model. The cycle repeats until either a no-match is confirmed or the finest available details are matched. It can be shown that an active observer can solve basic vision problems in a much more efficient way than a passive one. Problems that are ill-posed, nonlinear or unstable for a passive observer can become well-posed, linear or stable for an active observer[**1**].

1.1. Focus of Attention. Since an active vision system is often required to explore the environment with little a priori information, it is important that it has a way to limit its attention to certain regions of interest. The purpose of defining focus of attention (FOA) is to locate and analyse relevant information essential to the current task and ignore irrelevant details, so that computing resources can be more efficiently allocated. The issue of FOA is not restricted solely to vision, as it is also relevant to the other senses. For instance, the human brain can establish focus of attention on audio information. This is evident by the fact that two persons engaged in conversation in a noisy surrounding can comprehend each other's speech, while background noise is not processed. In the case of computer vision, recognition systems are often confronted with massive amounts of visual information, the majority of which is irrelevant to the current task. Without any means of pruning the information, most visual tasks would be too computationally intensive to realize real-time performance. A focus of attention algorithm that can drastically reduce the amount of computations required is therefore highly desirable.

One important issue in this regard is what features should be used to attract the vision system's attention. Psychophysical studies show that features such as line endings, closure, orientation, curvature, luminance, colour and motion attract early visual attention in the human vision system [**59**]. This is is not necessarily true for machine vision systems. In this case, the choice of features should depend on the task. For example, a motion tracking system should use motion as a feature for attracting attention. Due to the breadth and the sheer amount of work done in the area of FOA, we will not attempt to give a detailed review of the literature on this subject. Instead, we will focus on research that employs colour as the cue for locating areas of interest, since colour is what we have used to deduce the locations of objects in our experiments.

With recent advances in colour CCD camera technology, colour vision is becoming much more affordable and reliable. The use of colour as a visual feature is less computationally expensive and often more reliable than geometrical features in matching,

image segmentation [**15, 42, 53, 60**], and even object recognition [**25, 26, 32, 58**]. Relatively little work has been done on using colour as the only cue for focus of attention [**3, 21, 24, 62**]. The Georgia Institute of Technology Mobile Robot Laboratory has implemented a multi-agent robot trash-collecting team which used colour to detect trash (red soda cans) and garbage bins (painted blue)[**3**]. Although experimental results were satisfactory (they won the Office Cleanup Event at the 1994 AAAI Mobile Robot Competition), some rather restrictive assumptions had to be made: All objects of interest were assumed to be painted uniformly in one colour, and only red, blue and green objects could be detected. Fujiwara et al [**24**] implemented a mobile robot vision system which performs visual search of multi-coloured objects using an extended version of Swain and Ballard's histogram backprojection algorithm [**58**]. The algorithm performed well even in cluttered scenes, and often provided enough localization for the robot to grasp the object without the need of verification by other methods. Ennesser and Medioni [**21**] introduced a focus of attention algorithm using local colour information. Unlike Swain and Ballard's histogram backprojection algorithm, which performs point-by-point processing over the entire scene, this algorithm exploits more information and prunes the search space by directly matching local sub-images with the model.

Colour-based recognition suffers from the problem of colour variation under different lighting conditions. Colour constancy algorithms have been proposed to remedy this effect [**23, 25, 26, 32**].

A focus of attention mechanism should be able to spot the region of interest in the scene, allowing the greatest part of the scene to be quickly discarded. Speed is therefore an important criterion to the quality of an FOA algorithm. Colour information can often be inferred more rapidly than geometric features, and is shown to be effective as a cue for attention in some human visual tasks. Some open issues need to be addressed, such as the problem of colour variation [**23, 25, 26, 32**], and the choice of colour space [**21, 42**]. Colour alone cannot solve all visual search problems [**24**],

but can often offer a significant reduction in search space complexity when looking for coloured objects.

2. Object Recognition

The problem of object recognition involves extracting image features from sensors (edges, regions, texture, range, etc.) and matching an object model to those features to determine the object's location, orientation, and identity[**5**]. Object recognition systems can be classified into two categories:

(i) Two-Dimensional Object Recognition
(ii) Three-Dimensional Object Recognition

Since our research primarily deals with 3-D object recognition, the discussion on the former will be relatively brief.

2.1. Two-Dimensional Object Recognition. Two-dimensional object recognition differs from its three-dimensional counterpart in one important way, in that it makes no attempt to infer any 3-D information (such as depth, object occlusion, shading, and surface orientation). Despite this restriction, many other cues can be extracted from a 2-D image, such as colour, brightness, reflections, texture, edges, shape and size. Even though we live in a three-dimensional world, 2-D recognition is sufficient to solve many problems. For example, OCR (Optical Character Recognition) systems are exclusively 2-D since writing is two-dimensional by nature. Other applications such as remote sensing (flat terrain seen from high altitudes), radiology (where the image is a "shadow" of the object), and microscopy (where the image is a cross-section of the object) can often be adequately resolved using 2-D recognition schemes [**51**]. Even objects that are three-dimensional in nature can sometimes be successfully recognized by 2-D recognition systems. For example, an industrial robot vision system can recognize parts on a conveyor belt, given conditions such as a well illuminated scene, non-overlapping objects, and objects in specific 3-D orientations.

Chin and Dyer [**16**] have classified 2-D object recognition methods into three classes based on the kinds of features employed for matching. The first type of method uses global features such as an object's size, shape, and perimeter. The second type uses local features that describe more complex properties of the object, such as edges and curve segments. The final type uses local and relational features which are organized in a graph. Nodes describe local features and arcs describe the relationship between the connected pairs of features. This is called the relational graph method.

The features mentioned above are more or less all the information available for a 2-D object recognition system. It quickly becomes apparent that this lack of higher dimensional spatial information severely limits vision systems' capabilities towards general-purpose object recognition, particularly for mobile robot vision, which inevitably deals with objects and obstacles in the 3-D world. Often little a priori knowledge can be assumed. Nonetheless, 2-D vision is a fairly broad field on its own accord, with many sound theories proposed and many practical systems having been implemented. Since this research is concerned with 3-D object recognition, a detailed review of 2-D object recognition is beyond the scope of this thesis.

2.2. Three-Dimensional Object Recognition. The term "three-dimensional object recognition" can be rather vague, since many different schemes and methods have been proposed to solve this problem. Some employ range data, some attempt to interpret multiple views of objects, and some attempt to infer 3-D features of an object from a single 2-D intensity or colour image. Therefore it is not easy to give a formal definition to 3-D object recognition. For the purpose of this thesis, we define 3-D object recognition as "the ability to infer the identity, location, and orientation of solid 3-D objects by extracting view-invariant 3-D features from the scene and matching them with features of 3-D object models" [**17**].

The definition given above indicates that 3-D object recognition does not necessarily require explicit 3-D (range) data. In fact, most early research in 3-D recognition employed intensity images due to the relatively low cost of frame grabbers and digital

cameras. The main problem faced by 3-D recognition systems that employ 2-D intensity images is that considerable information about the three-dimensional structure of the scene is lost in the projection onto the two-dimensional image [**5**]. Therefore, 3-D features must be extracted implicitly from the 2-D images. Various kinds of 3-D information can be inferred from an intensity image using attributes such as shading, texture, edges, and corners [**12**].

Recognition of 3-D objects is difficult because a 3-D object can have many distinguishable appearances, depending on its location and orientation with respect to the observer. Therefore the object model must contain a description of the object from all possible viewing angles. A straightforward exhaustive search approach is clearly impractical. More elaborate feature extraction and matching algorithms are required. For example, Bergevin and Levine [**6, 7**] introduced an object recognition system called PARVO (**P**rimal **A**ccess **R**ecognition of **V**isual **O**bjects) which takes line drawings of objects as input. PARVO extracts features such as corners, T-junctions, and symmetry. Using these features, the object is segmented into simple parts, and each segmented part is matched to a geon label (geons are specific geometric shapes which will be discussed in detail in the following section).

Another approach that has gained a lot of attention in the object recognition community is the aspect graph (or characteristic views) approach. Koenderink and Van Doorn [**34**] introduced the concept of aspect graphs for representing shapes of 3-D objects. This method stems from the observation that although an object can have an infinite number of orientations, it has a limited number of "characteristic views", or aspects, from which the object has the same qualitative appearance, i.e. the same faces and edges are visible, even though quantitative attributes such as the length of edges may be different. All the different aspects of an object comprise an aspect graph of the object. Each node of the aspect graph represents an aspect, and each arc represents a shift in viewpoint that transforms one aspect to another. Stewman and Bowyer [**19, 55, 56, 57**] proposed a "viewing sphere" approach to generating the aspect graph. A "viewing sphere" is an imaginary sphere whose centre is the

geometric model of an object. The surface of the sphere is tessellated into small cells, each of which represents the same aspect view. Using all the cells on the viewing sphere (which corresponds to all possible aspects of the object model), the object model's aspect graph can be generated. This aspect graph approach to 3-D object recognition is considered important because it provides a complete viewer-centered representation of an object [**19**] and it has been shown to be effective in the recognition of complex polyhedral objects.

The aspect graph method and PARVO, like other edge-based approaches, have the shortcoming that no explicit surface information is available. Objects represented by edges alone may lead to ambiguities like the one shown in Figure 2.1. Although the inference of surface information from intensity images is possible (e.g., through shading and texture), it is often difficult to accomplish and is error-prone. Explicit 3-D information of the scene is therefore highly beneficial to the efficiency of 3-D object recognition.

The past few years has seen a significant increase in research efforts into range-based object recognition [**27, 29, 36, 39, 43, 47, 48, 54, 66**]. This is partly due to the off-the-shelf availability of commercial laser rangefinders. Rangefinders can now be implemented at a relatively modest cost and effort, and the quality of range data has been steadily improving.

In 3-D object recognition, range data is usually presented in the form of range images (or depth maps). A range image is a two-dimensional array, with each pixel indicating the distance measurement from a known reference coordinate system to a particular point on the object surface in the scene [**39**]. The main advantage of using a range image for object recognition is that it gives a description of the physical structure of the object, rather than just the intensity or colour of light reflected off the object. Range data is relatively uncorrupted by reflectance variations, ambient lighting and shadows [**5**], and quantitative geometric properties of the scene can be inferred accurately. Therefore, the recognition of objects by shape is much easier than in intensity or colour images. Of course, information regarding the brightness

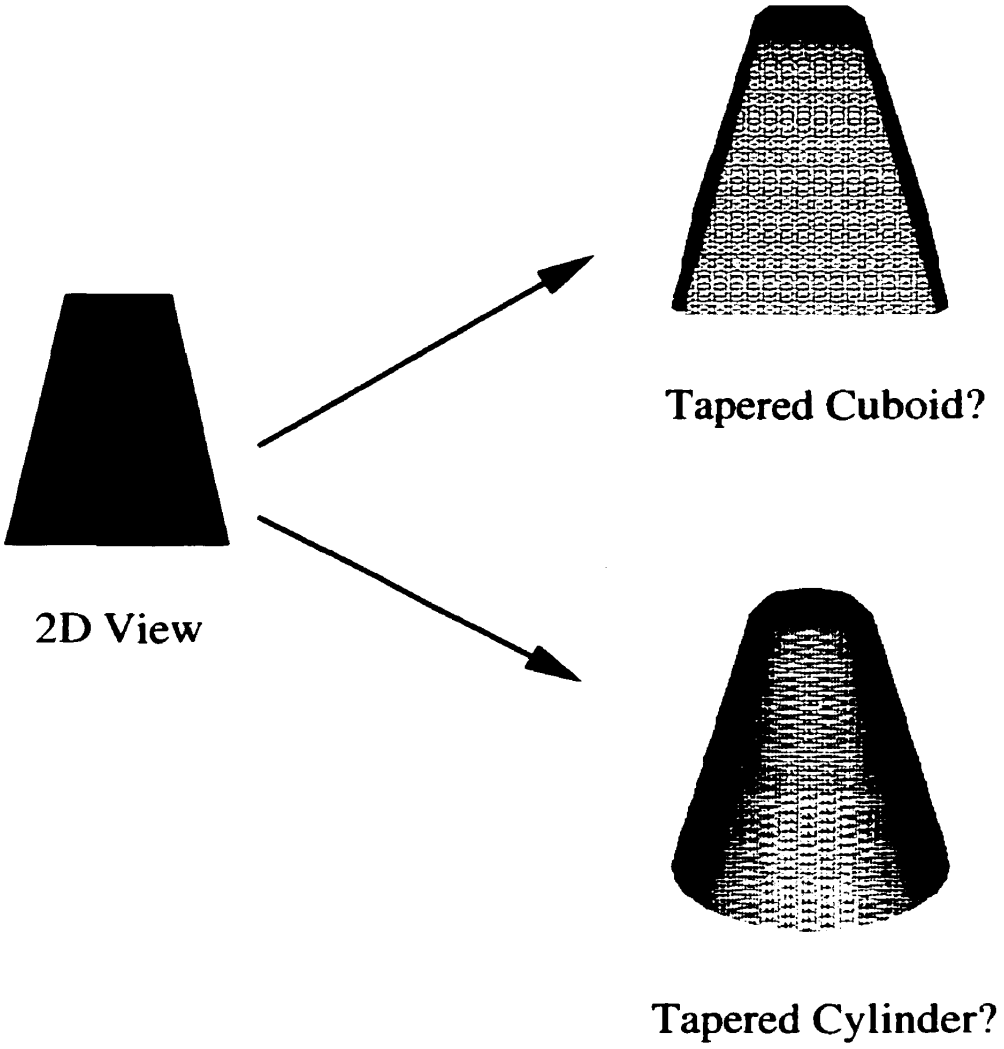


FIGURE 2.1. Example of ambiguity in 2-D vision.

or colour of the scene will be lost in a range image, and recognition methods that require such information will not be applicable. Some research has been done on fusing intensity/colour images and range images, utilizing the strength of both to achieve effective object recognition [**4, 20, 61**]. Those methods have shown limited success and tend to require very heavy computations.

Range data are fundamentally different from intensity and colour images, and the use of range data for object recognition has led to a whole new genre of data processing

and feature extraction methodologies. Nonetheless, the fundamental problems of object recognition remain the same: What features should be extracted? How should the objects be represented? The following section addresses different representational schemes for object recognition using range data.

3. Object Representation

An appropriate way of describing objects is of crucial importance to the efficiency and robustness of the object recognition system. Representation schemes for 3-D objects are usually either edge-based, surface-based, or volume-based [**13, 39**]. The edge-based approach is easy to implement because edges in a range image are simply locations where range values change significantly. Moreover, edges in range images can contain more information than those in intensity images: various types of edges such as convex, concave, limb and occluding can be differentiated in a range image [**39**]. However, an edge-based representation does not provide any explicit surface information which may be important for distinguishing surfaces having identical edge descriptions from a certain view, and ambiguities like the one depicted in Figure 2.1 may still arise.

The surface-based approach represents objects by characterizing surfaces bounded by edges. With explicit 3-D coordinates of each point made available by range images, an object's surface properties can be deduced using differential geometry. This approach tends to be more computationally intensive then the edge-based approach due to the calculations required for obtaining surface attributes such as surface normals. Also, surface segmentation can be a difficult problem.

The volume-based approach depends on the observation that most man-made objects can be loosely described as a composition of several (less than 10) geometric parts such as boxes, discs, and cylinders. This is analogous to the way each spoken word is regarded as a composition of a few phonemes in speech recognition [**13**]. The advantage of this approach is that the many-to-one mapping problem of object recognition is broken down into smaller problems of recognition at the level of simple

geometric parts. Once the recognition of parts (which is much easier than the recognition of complex objects) is accomplished, spatial relations between parts are used to differentiate objects with similar parts but different arrangements.

The next question is *what* geometric parts should be used to describe complex objects. Pentland [**45, 46**] first proposed the use of superquadrics as part primitives for object recognition. Superquadrics are a family of parametric shapes described by an equation that contains 5 parameters, and the parameter values determine the superquadric's shape (e.g., ellipsoids, cylinders, parallelpipeds, pyramids, cones, and some other round-edged shapes in between these standard shapes) [**48**]. Hanson [**31**] later proposed hyperquadrics for computer graphics applications, and Han et al. [**29**] used hyperquadric models for shape recovery from range data. Hyperquadrics are generalizations of superquadrics, their equation containing more parameters so that it can describe a larger set of geometric shapes. The research done on using superquadrics and hyperquadrics for shape recovery from range data [**11, 31, 48**] adopts more or less the same approach: an error-of-fit function is defined, and the parameters of the quadric shape are estimated by attempting to minimize the error-of-fit function using nonlinear optimization methods. This approach is problematic because the nonlinear optimization process tends to be very time-consuming, may fail to reach a global minimum, and sometimes require an initial guess. It is also doubtful that the human vision system recognizes objects by attempting to quantify shape parameters.

A new class of geometric primitives, called geons, has been proposed for 3-D object representation[**8, 9**]. As described below, geons fundamentally differ from parametric shapes such as superquadrics and hyperquadrics, and may lead to more robust 3-D object recognition systems.

3.1. What are Geons? Biederman [**8, 9**] proposed a theory known as Recognition-by-Components (RBC). The theory postulates that human perceptual recognition of objects is a process in which objects are segmented at regions of deep

concavity into an arrangement of simple geometric components, as shown in Figure 2.2. The fundamental assumption behind RBC is that a modest set of generalized-cone components ($N \leq 36$) is sufficient to represent all known objects for low-level perception. The set of generalized-cones is termed "geons" (short for "geometric ions"). The 36 geons are derived from three attributes of a geon's cross-section: edge (straight or curved), symmetry (symmetrical or assymmetrical), size (expanding, constant, or expanding and contracting), and one attribute of its axis (straight or curved). The most important difference between geons and other parametric shape primitives such as superquadrics is that a geon's attribute values need only be dichotomous or trichotomous rather than quantitative, so that a human being's limited capacities for absolute judgement are not taxed [8]. Moreover, slight quantitative variations in the attributes usually have little effect on the recognition of objects, as illustrated in Figure 2.3.

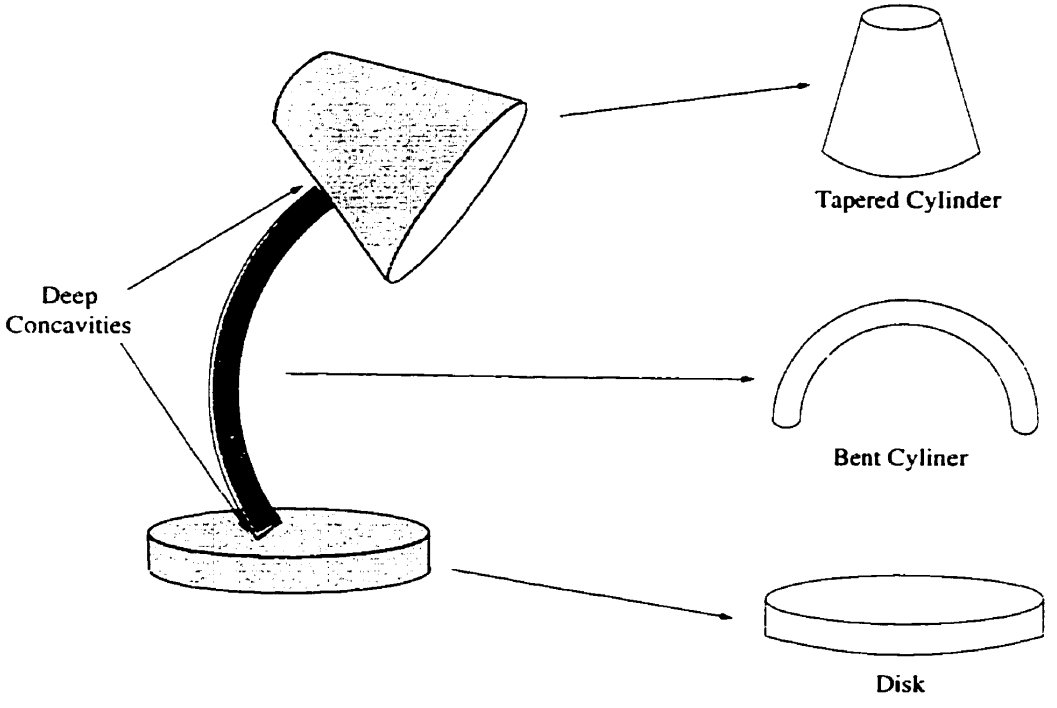


FIGURE 2.2. Example of an object composed of simple geometric primitives.

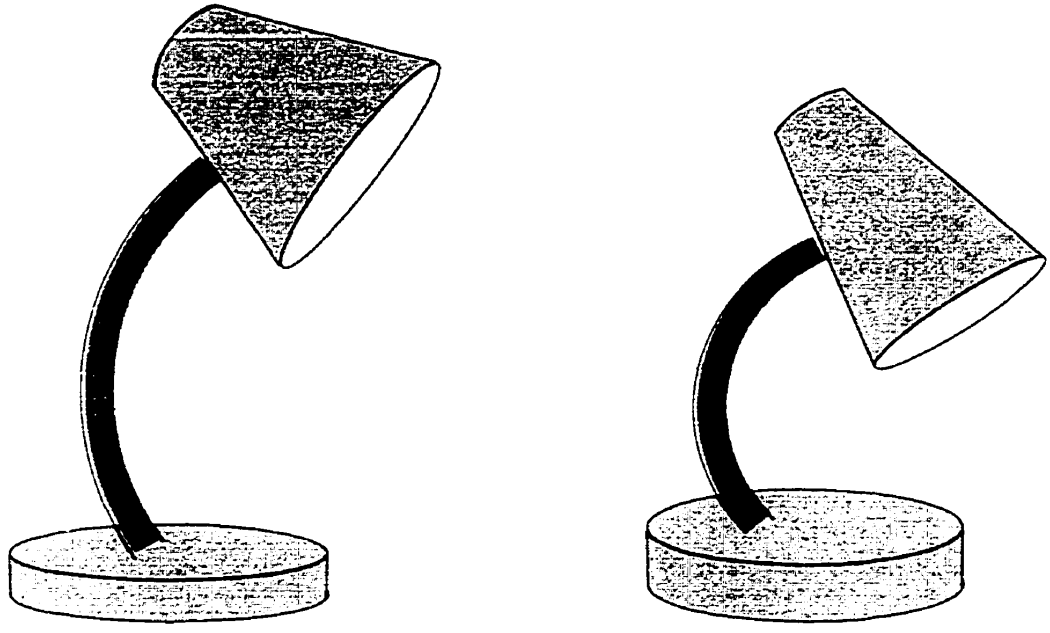


FIGURE 2.3. Two objects composed of the same geometric primitives with slight quantitative variations.

Brown et al [**14**] raised the question of whether geons are indeed geometric *primitives* in the traditional sense that they "pop out" in search tasks. Features are said to "pop out" if they are detected automatically and preattentively. Experiments were performed on human subjects, who were shown images of objects, and their reaction time (RT) needed to detect the presence or absence of a target geon was measured. Experimental results indicated that geons are not processed preattentively by the human vision system[**14**]. Nonetheless, geons can be identified and distinguished from each other very quickly. In addition, Biederman [**8**] has shown that objects represented by geons can be recognized with relative ease, even under sub-optimal viewing conditions such as an object being partially occluded, lacking some components, or having its contour partially deleted. These attributes make geons an attractive candidate for object representation.

The RBC theory has motivated further studies of geons in the computer vision community. PARVO [**6, 7**] was the first significant effort to build a geon-based vision system, and it is also one of the few vision systems whose design respects and makes

explicit the main assumptions of the RBC theory. Although PARVO was created to take only line drawings of objects as input, it is possible to extend the system to handle objects found in range images. Hummel and Biederman [**9**] have implemented a neural net model for geon-based recognition which constitutes a framework for general object recognition based on RBC, but no experimental results have been made available.

A relatively small research effort has also been addressed to the recognition of geons from range data [**48, 66**]. Raja and Jain [**48**] have performed geon recognition by relating a subset of 12 geons to superquadrics fitted to range data. Experiments were done using real range images of geons. Results show that estimation of superquadrics parameters is extremely sensitive to noise and "rough" object surfaces, but under normal circumstances, qualitative shape attributes can be inferred from superquadric parameters quite accurately. Wu and Levine [**66**] used multiview range images to approximate shapes by a set of 7 geons. Experimental results show that multiview analysis leads to better performance in shape approximation than single-view, particularly inder the influence of noise, missing data, and slight variations in shape.

The set of 7 geons used by Wu and Levine [**65, 66**] is the same set of geons we attempt to recognize in this research (see Figure 2.4). These specific geon types were chosen because they are regular, simple, and symmetrical shapes. Most of them can be described without ambiguity by a simple geometric term, such as ellipsoid, cuboid, cylinder, cone and pyramid. These shapes are also consistent with the basic forms used by more traditional methods of 3-D object representation such as sculpture [**65**].

Although geons have emerged as the most popular shape representation for *generic* object recognition, the above-mentioned research related to geons is still being applied to very constrained domains. In some cases synthetic data were used, or segmentation was performed manually, or the objects are made unrealistically simple. Much work remains to be done before a generic geon-based recognition system can be realized.

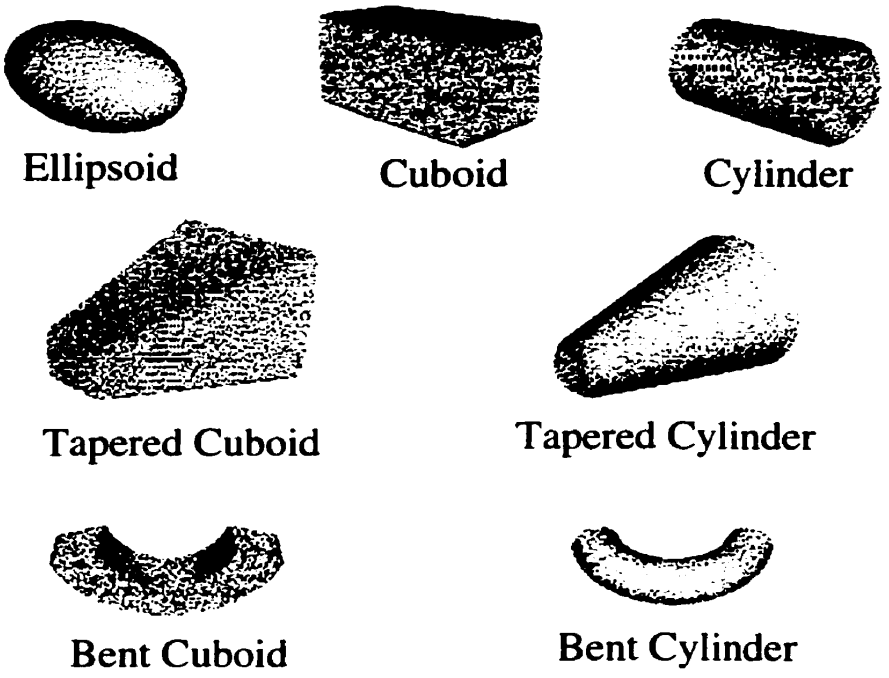


FIGURE 2.4. The 7 geons used in this research.

4. Proposed Approach

Speed is the most crucial criterion in determining our method of establishing focus of attention. In general, 3-D feature extraction methods are more reliable than 2-D methods, but they are also more computationally expensive [**24**]. Colour can be used as a 2-D global feature which requires less computation in matching than structural features such as shape, corners or edges. Colour detection is also an inexpensive method due to the relatively low cost of colour CCD cameras in recent years. For the above reasons, colour was chosen as the cue for focus of attention in our experiments.

Upon detection of the object of interest, a robot navigation strategy must be employed to steer the robot to the vicinity of the object without colliding with obstacles. Zelek [**67**] proposed and implemented a mobile robot control architecture called SPOTT (A **S**ystem which integrates **P**otential fields for planning **O**n-line with **TR**+

program control in order to successfully execute a general suite of Task commands). SPOTT integrates data from laser rangefinders, sonar, infrared sensors and tactile sensors to navigate the robot in an unknown or partially known environment. Local path-planning is based on a potential field method. Despite SPOTT's success in navigating a mobile robot through an indoor environment, it is prohibitively slow due to the massive amount of sensor data it is required to manipulate, and the intense amount of computation needed to calculate the potential field. A much simpler navigation algorithm was therefore adopted in this thesis. The Mobile Robotics team at McGill University recently developed a path planning program in preparation for the 1997 AAAI mobile robot competition [1]. In this algorithm, only sonar sensors and a colour camera are used to perform path planning, and it is shown to be very fast and robust. With slight modification this path planning module was incorporated into our application.

Speed is also of critical importance to our recognition process. Most research done on 3-D shape recognition has required complete range maps of objects of interest [**13, 48**]. The problem is that the acquisition time for range maps is often quite long (on the order of tens of seconds). Since we are only attempting to differentiate between a set of 7 geons, we submit that only a few strategically placed laser-stripes will suffice for our purpose. Very little work has been done on 3-D object recognition using such sparse data. Qiang et al. [**47**] proposed a method for recognizing and locating polyhedral objects using as little as one laser stripe. In their experiments, they use a model for each stable position for every object. In our case, however, we do not have precise models available since geons have qualitative attributes. In our experiments we will show that three horizontal and three vertical laser stripes will be sufficient for distinguishing our set of 7 geons.

Many recognition systems begin with quantitative information derived from the image and perform recognition by matching the information with object model

[1]The team came first in the event *Find Life on Mars (Single Agent Nonmanipulator Category)*. Team members: Francois Belair, Eric Bourque, Deep Jugessur, Rob Sim. Supervisor: Professor Gregory Dudek.

databases. Even previous work on geon recognition [**48, 66**], mentioned in the last section, involves estimation of certain quantitative parameters. To comply with the essence of geons and the underlying concepts of the RBC theory, we attempted to recognize geons through the inference of qualitative features from range data. An evidence-based recognition approach [**33**] was adopted for our system.

The idea behind evidence-based recognition is that rather than using all the information provided by a representation, it is possible to use only "discriminating" information, or evidence, to cue certain objects. In our case, evidence such as tapering, bending, curved/straight cross-section, and absence/presence of corners will be useful for geon recognition.

With this qualitative approach, we hope to accomplish fast and accurate geon recognition without the need of complex parameters estimation.

CHAPTER 3

Hardware Implementation, Calibration, and Specifications

The main objective of this research is to recognize geons from range data. Therefore the laser rangefinder is the most central piece of hardware to our experiments. The first section of this chapter gives a general overview of laser ranging technology, as well as the implementation of our rangefinder. The second section describes our calibration method. The following sections will give brief descriptions and specifications of the other hardware used: The colour camera for establishing focus of attention, the pan-tilt unit for camera movement, and the mobile robot for navigation and exploration.

1. The Laser Rangefinder

Many tasks call for the need to quantify range, in particular, mobile robot navigation. A mobile robot's task is often to gather information from its surroundings and make intelligent movements accordingly. Range information is especially useful, if not crucial, to successful path-planning in mobile robots. That is why most mobile robots are equipped with at least one set of range sensors. The use of range data for *object recognition* did not become common until fairly recently. This is because range data must be highly accurate to be deemed practical for object recognition purposes, and early range-measuring techniques lacked such accuracy.

There is a number of distance measuring techniques. The more commonly used ones are[**22**]:

- Laser ranging
- Stereo disparity
- Sonar
- Interferometry

Laser ranging is by far the most popular candidate for range image acquisition, since other techniques suffer serious drawbacks: Stereo vision exhibits the correspondance problem; sonar is inaccurate and therefore rarely used for object recognition; interferometry is costly and limited to relative distance measurement. Also, lasers produce a narrow and collimated beam with minimal divergence. This property allows the source to be highly directional and spatially selective[**37**], which makes it very suitable for long-distance measurement and ranging. Lasers generally transmit light of a fixed wavelength, so a narrow band optical filter can be placed in front of the image sensor to reject ambient light and detect nothing else but the laser beam.

There are three different methods of laser ranging:

- Triangulation
- Time of flight (TOF) measurement
- Phase-shift measurement

The concept of time of flight laser ranging is similar to that of sonar range sensing: short bursts of energy are projected from the source, and the time it takes them to be reflected back to the sensor is measured. Using elementary physics, the range between the sensor and the surface that reflected the signal is $d = ct/2$, where d is the range, c is the speed of light, and t is the round-trip time taken by the signal. Object recognition often calls for range data with accuracy in the millimeter range, and to achieve that kind of accuracy, the TOF laser rangefinder must be able to measure time t with subnanosecond precision. Since timing circuitry with that kind

of resolution is very expensive, TOF laser rangefinders are rarely used for object recognition.

Phase-shift rangefinders deduce distance by measuring the phase difference between the outgoing signal and the reflected signal. The range d can be derived from the equation [**22**]:

$$d = \frac{\phi c}{4\pi f} \tag{3.1}$$

where ϕ is the phase-shift, c is the speed of light, and f is the frequency of the laser. One problem with this method is that phase-shift measurement can lead to ambiguous results: the laser's sinusoids repeat every 360^o, so equation (3.1) should really be:

$$d = \frac{(\phi + 2\pi n)c}{4\pi f} \tag{3.2}$$

where $n = 0,1,2,...$

n cannot be known without using other means of range measurement. This problem is often countered by restricting the operating range of the rangefinder within an interval where no ambiguities may arise. Another problem with phase-shift measurement is that sophisticated and expensive phase measurement electronics are needed.

In a triangulation-based rangefinder, the laser projector is usually tilted at a fixed angle with respect to the camera's optical axis (see Figure 3.3). Δh, which depicts the location of the reflected laser light's projection on the image plane, varies according to the distance z. Therefore range information can be deduced from Δh. One drawback of triangulation ranging is that its accuracy diminishes with range. It will be shown later in this chapter that Δh is roughly proportional to the reciprocal of distance z. As the distance grows larger, Δh must be measured with higher precision to give the same resolution in z values. Another drawback of triangulation is the missing part problem: since the laser projector is not aligned with the camera's optical axis, there

may be situations in which the laser hits an object part that is not visible to the camera. This situation is illustrated in Figure 3.1.

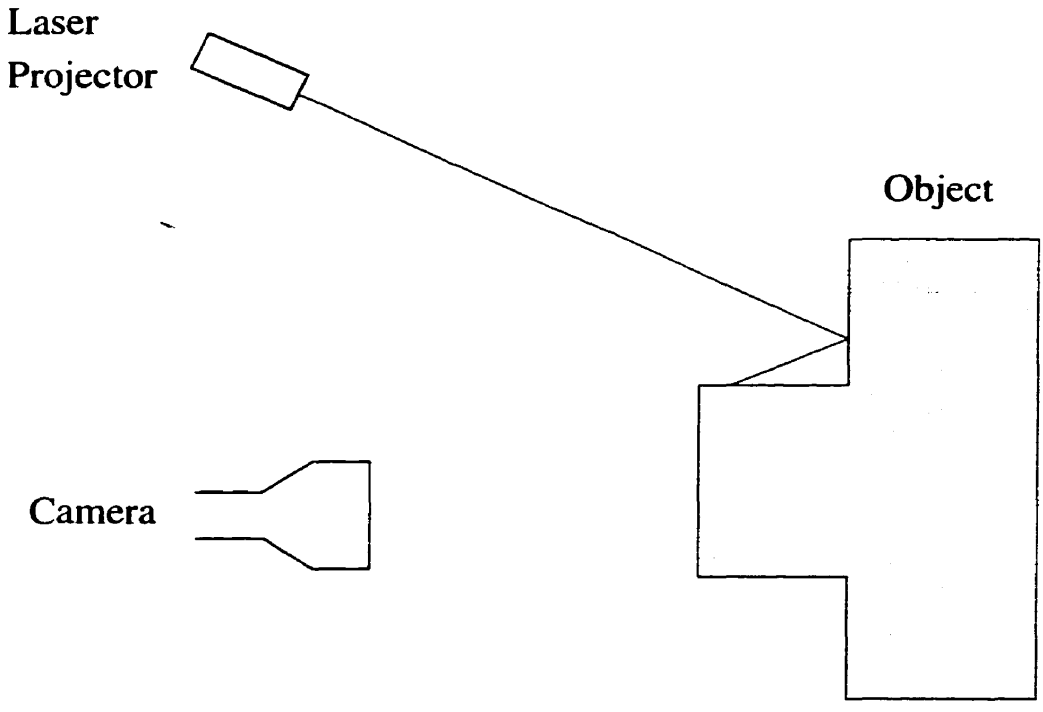


FIGURE 3.1. Illustration of the "missing part" problem in triangulation-based laser rangefinders.

Although triangulation-based rangefinding has its share of disadvantages, it can be implemented inexpensively and tends to give good accuracy over a range of several meters. Since our experiments only require range data up to the range of about 2 meters, the triangulation approach was adopted for the implementation of our rangefinder.

1.1. Setup. Figure 3.2 shows our laser rangefinder. The laser projector on top of the black-and-white camera projects a horizontal laser line while the one beside the camera projects a vertical line. The lasers used in this setup have a rated power of 20 milliwatts each and a wavelength of 670nm. A bandpass optical filter is placed in front of the camera to remove ambient light.

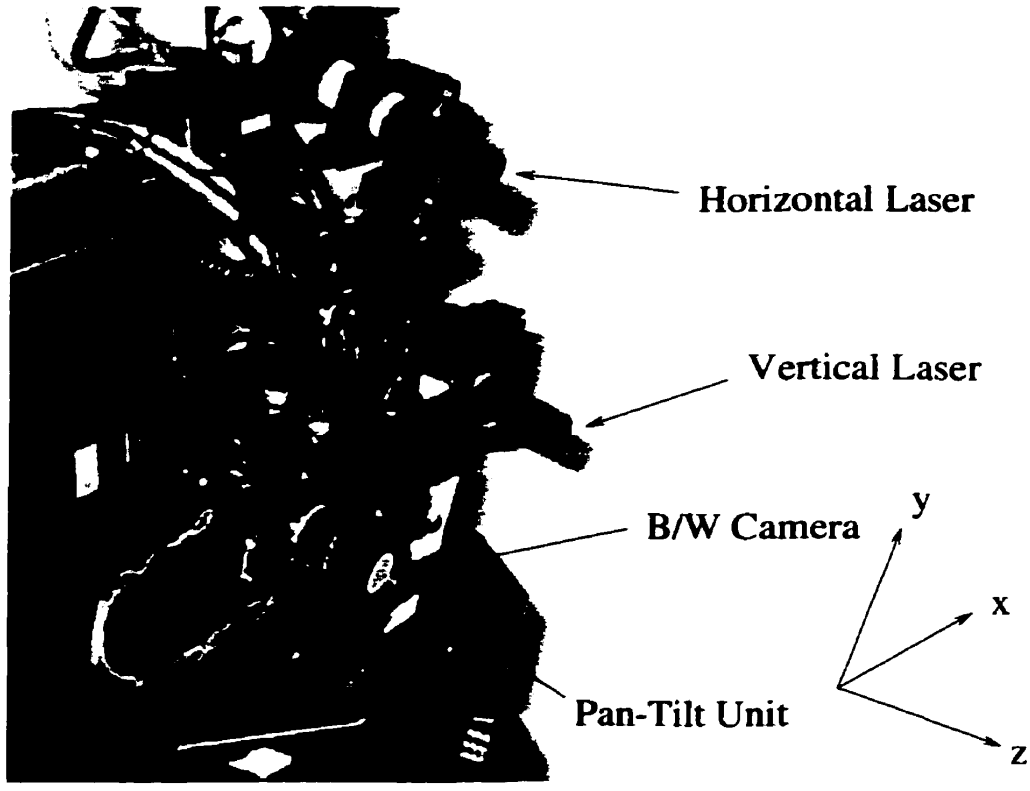


FIGURE 3.2. The laser rangefinder.

Figure 3.3 shows the rangefinder configuration viewed from one side. The laser projector and the image plane (CCD array) are separated by a known baseline distance A. The laser projector is tilted at a known angle ρ with respect to the z-axis (which is the same as the optical axis). Δh is the vertical displacement of the laser-stripe image on the image plane with respect to the centre of the image plane. The basic principle of triangulation is that given two angles and one side of a triangle, all the other angles and sides can be derived. In this case, ρ and A are known, and β can be derived as shown in equation 3.3:

$$\beta = \frac{\pi}{2} + \theta \tag{3.3}$$

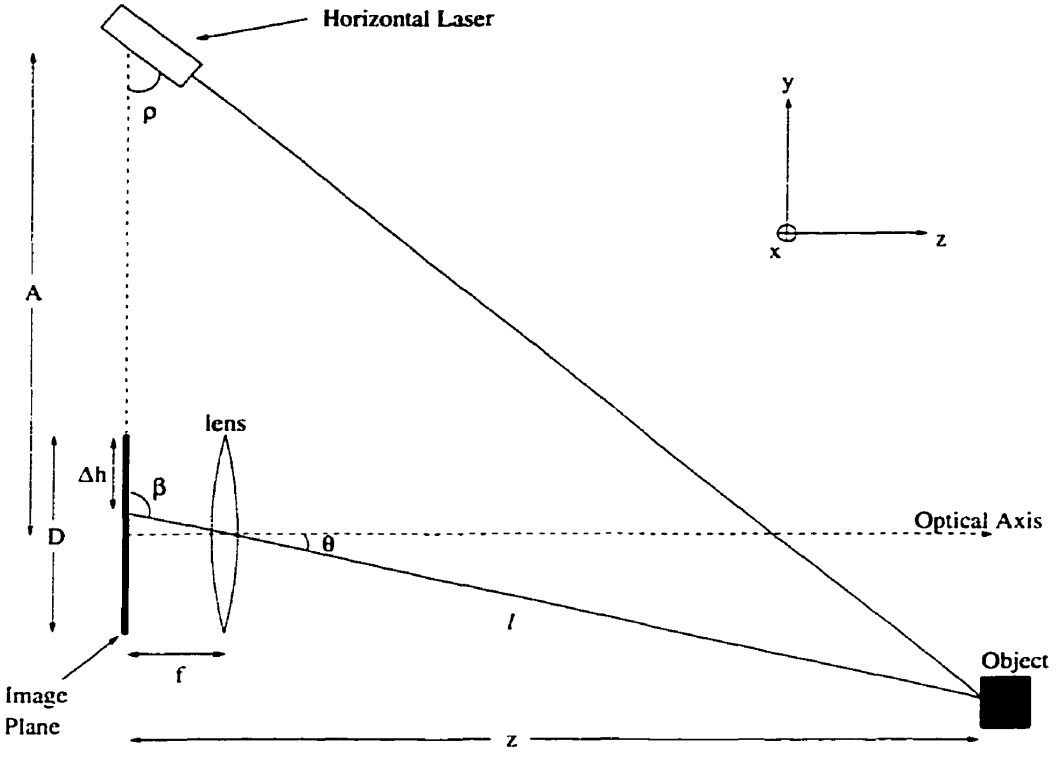


FIGURE 3.3. Horizontal configuration of the laser rangefinder.

where

$$\theta = arctan(\frac{D/2 - \Delta h}{f}) \tag{3.4}$$

With the values of ρ, β, and A known, the length of the side l can be calculated. Subsequently, the distance z can be easily calculated as:

$$z = l\cos\theta \tag{3.5}$$

In theory, the distance z between the object and the image plane can be derived using equations 3.3 and 3.5. In practice, however, it is extremely difficult to measure the values of physical dimensions such as baseline distance A and angle ρ with sufficient accuracy. Most laser rangefinders therefore require calibration. The calibration of rangefinders will be discussed in the next section.

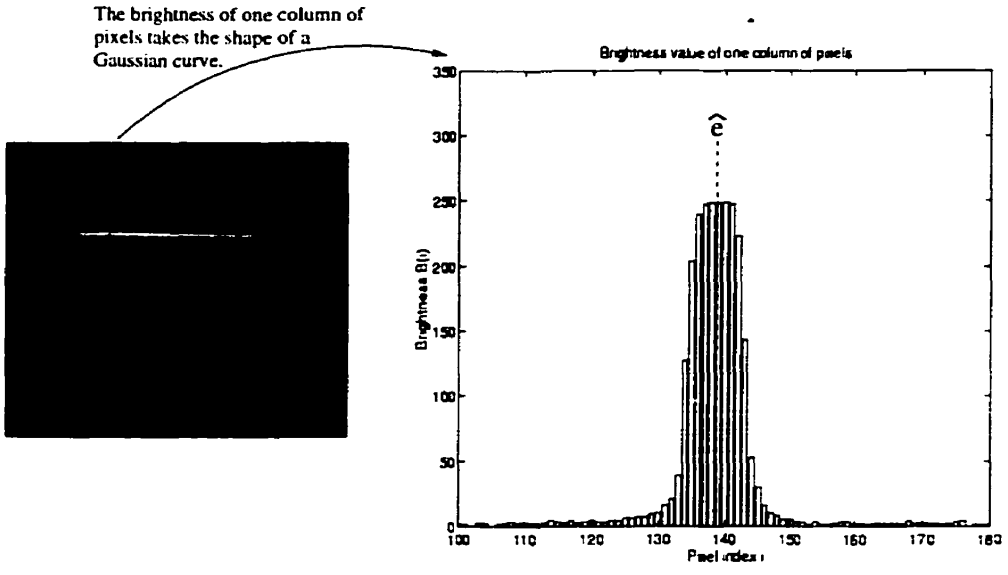


FIGURE 3.4. Brightness of a column of pixels.

In order to obtain an accurate reading of distance z, the location of the laser stripe image on the image plane (Δh) must be measured with sub-pixel precision. In general, the brightness profile of the laser stripe on the image plane takes the shape of a Gaussian curve, as shown in Figure 3.4. Treating this profile as a Gaussian curve, the mean of the Gaussian can be calculated and taken as the "location" of the laser. Since the profile is obtained from a digital image, it is discrete in nature and therefore the mean $\hat{e}$ can be calculated using the weighted sum-of-index method:

$$\hat{e} = \frac{\sum_{i=0}^{n-1} i \cdot B(i)}{\sum_{i=0}^{n-1} B(i)} \tag{3.6}$$

where n is the number of rows of pixels in the image.

The setup of our rangefinder is quite typical of triangulation-based rangefinders. Although variations in attributes such as baseline distance A and angle (ρ) can effect performance slightly, the accuracy of a rangefinder is mainly dependent on its calibration, which we will discuss next.

2. Calibration

There are several calibration techniques for laser rangefinders in the literature [**50, 64**]. The most commonly used one involves solving for the camera model's parameters such as focal length, lens distortion coefficients, and dimensions of CCD elements. Although these parameters are provided by the camera manufacturers, they are often not accurate enough for 3-D vision applications. The calibration procedure involves placing an object - usually a precisely marked grid - at a known location and orientation with respect to the camera. An image of the grid is acquired, and features such as points and edges are extracted. Using these features, a set of geometric equations can be derived. If the number of equations is greater than or equal to the number of unknown parameters, the parameters can be obtained using nonlinear optimization. The drawback of this technique is that it tends to be very computer-intensive. The nonlinear optimization often fails to converge to a solution, due to slight variations in physical parameters, e.g. decentration of lenses [**50, 64**], imperfections in CCD elements, etc.

A very simple and direct approach was used for calibrating our laser rangefinder. A flat calibration board was placed at a known distance in front of the camera, and the rangefinder was moved away from the board by small known increments, each time taking an image of the laser-stripe (In this experiment, 10 images were taken and averaged in order to minimize the effect of random noise). The location of the laser-stripe on the image (Δh) corresponding to each known distance was recorded in a look-up table. Figure 3.5 shows a plot of Δh values versus z.

The next step is to interpolate the points to obtain a curve. This step is needed in order to handle range values that lie between the sampled points we obtained from calibration. The following relationship between z and Δh can be derived by observing Figure 3.3:

$$\tan\theta = \frac{D/2 - \Delta h}{f} = \frac{z/\tan\rho - A}{z} \tag{3.7}$$

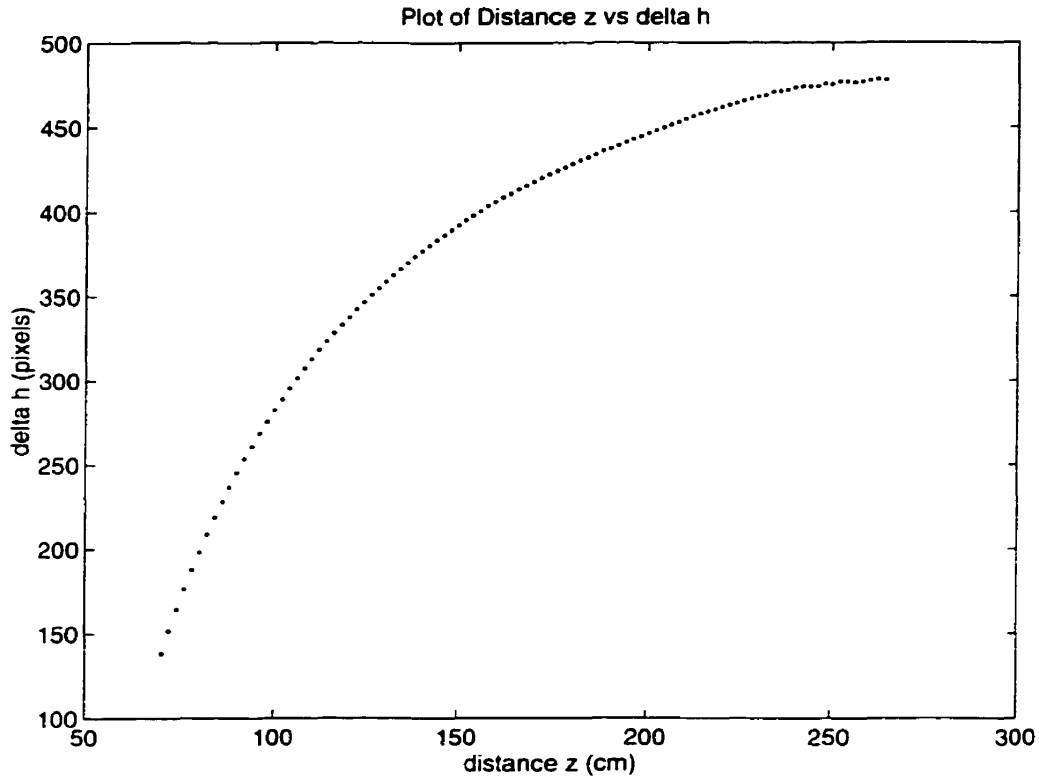


FIGURE 3.5. Plot of z versus Δh.

Solving this equation leads to:

$$\Delta h = \frac{A}{z} - \left(\frac{1}{\tan \rho} - \frac{D}{2}\right) \tag{3.8}$$

or simply

$$\Delta h = \frac{A}{z} - k \tag{3.9}$$

where k is a constant.

Theoretically, the values of A, D, and ρ in equation 3.8 can be measured. However, it is extremely difficult to measure those values with sufficient accuracy for our purpose. It is also possible to fit a curve to the points by solving for A and k in equation 3.9 using nonlinear optimization. The disadvantage is that triangulation-based

rangefinders suffer from diminishing accuracy with increasing range, so the points at long range are inaccurate compared to the ones at close range. Fitting a curve to all of the points will therefore compromise the accuracy of the measurements taken at close range. To alleviate this problem, another approach was used. Linear interpolation was done on each pair of consecutive points. In other words, the space between two adjacent points was simply joined with a straight line, as shown in Figure 3.6. Error is introduced by the discrepency between the straight line and the actual data. This error can be minimized by sampling images at small intervals (e.g., 5mm).

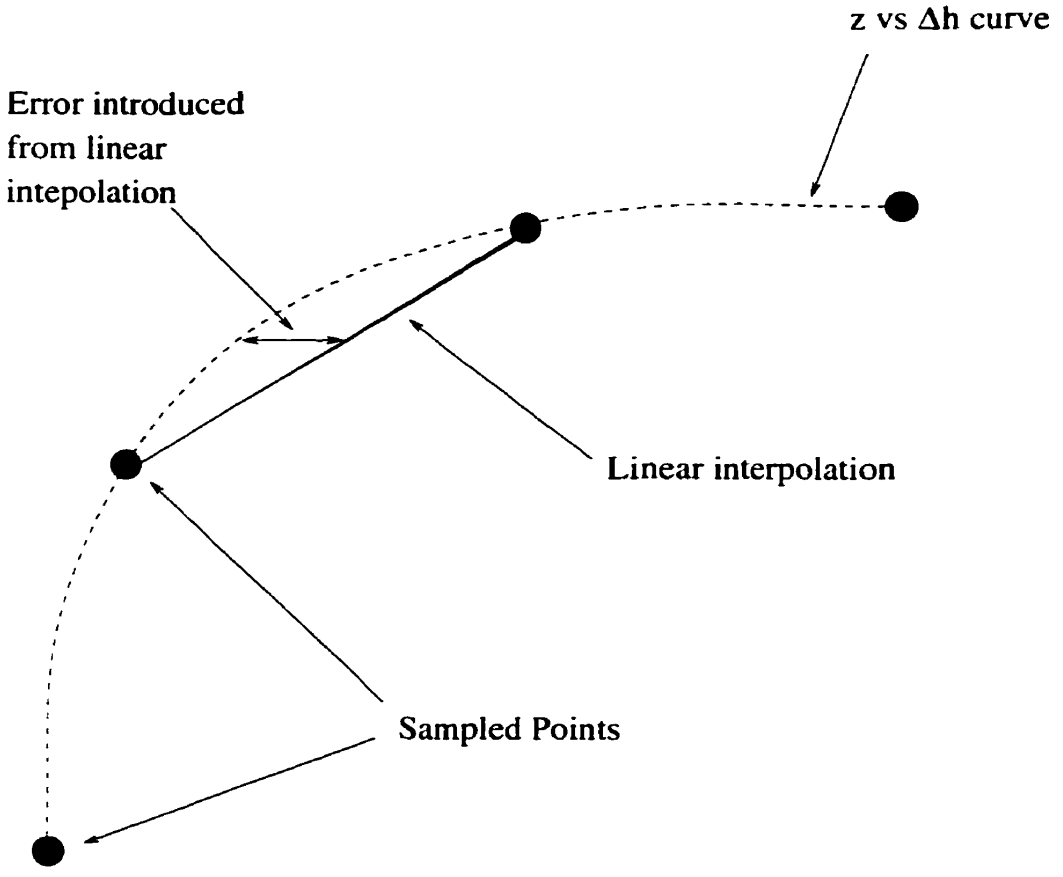


FIGURE 3.6. Linear interpolation of z versus Δh curve.

Ideally, the horizontal laser stripe should be projected onto the CCD array horizontally. In practice, it is difficult to align the laser projector and the CCD camera perfectly, and spherical lens distortion causes the stripe to bend slightly, as shown

in Figure 3.7. To take this into account, Δh is calculated independently for every column in the image.

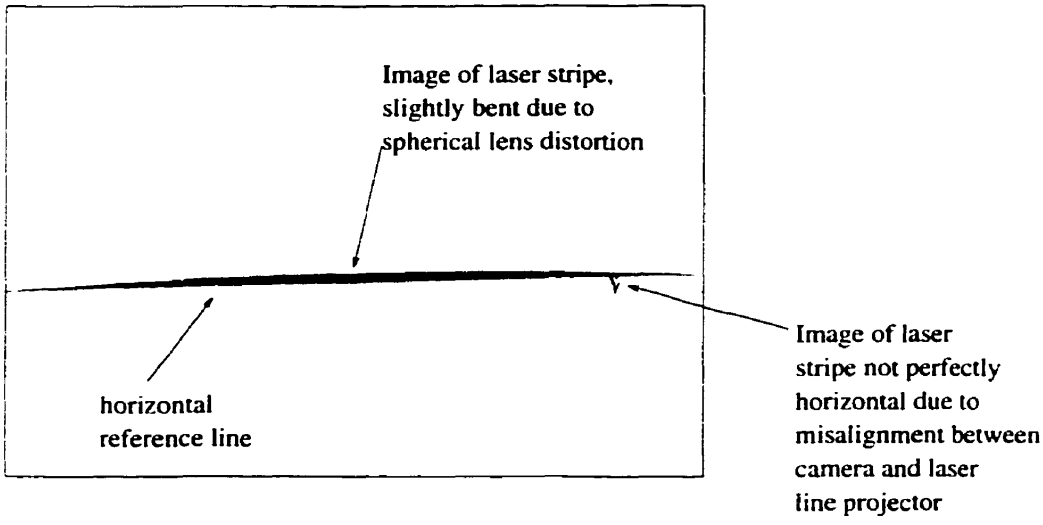


FIGURE 3.7. Typical image of laser stripe.

Every point on the horizontal laser stripe represents an (x, z) coordinate in space. The procedure mentioned above only solves for the value of z. There remains the problem of solving for x. Assuming that spherical lens distortion is negligible (this assumption is valid in our case since a 16mm lens, which has minimal spherical distortion, is used), x can be solved using simple geometry. Figure 3.8 shows the top view representation of the rangefinder. To simplify calculations, we define the frame of reference such that the optical axis intersects with the x-axis at $x = 0$. The point $p_1(x_1, y_1, z_1)$ on the laser-stripe is projected onto the point $p'_1(x'_1, y'_1)$ on the CCD image plane. z_1 can be found by calculating the value of Δh on the column y'_1, as explained earlier. Once z_1 is known, x_1 can be calculated using equation 3.10:

$$\frac{x'_1}{f} = tan\theta = \frac{x_1}{z_1 - f} \Rightarrow x_1 = \frac{x'_1(z_1 - f)}{f} \tag{3.10}$$

where f is the focal length of the lens, and x'_1 is the horizontal distance between p'_1 and the center of the image plane, which can be easily derived if the dimensions of the CCD array is known.

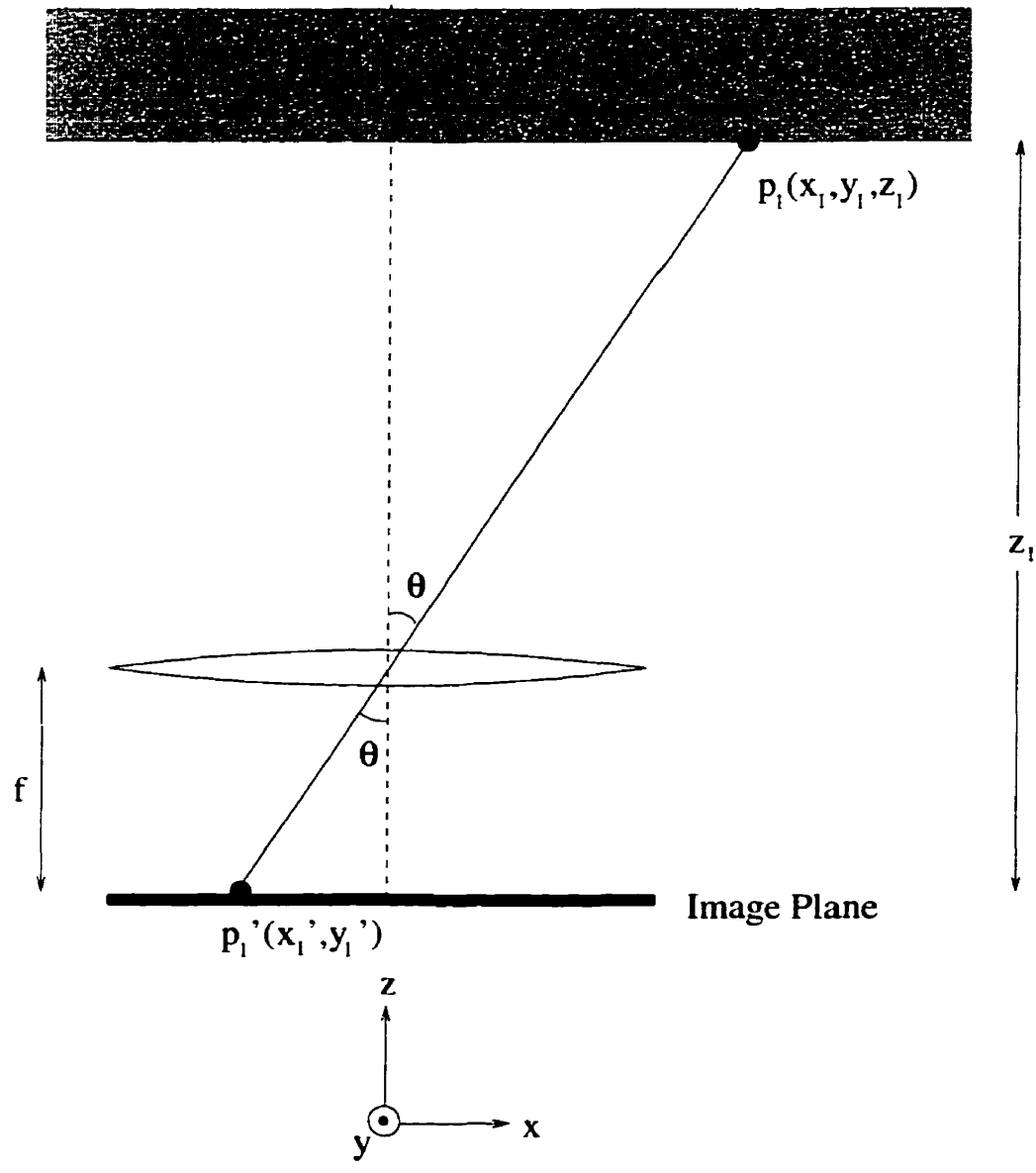


FIGURE 3.8. Topview of the laser rangefinder configuration.

The setup and calibration of the vertical laser are very similar to that of the horizontal laser, and therefore will not be discussed in detail.

2.1. Performance. The accuracy of our rangefinder was evaluated by placing the calibration board about 1 meter from it and obtaining the range profile. Figure 3.9 shows the distribution of the corresponding z values. The error at a range of 1 meter is about 1mm (0.1% error). This performance is comparable to most other triangulation-based rangefinders reported in the literature[**10, 38, 44, 54**].

Although our calibration method resulted in accurate readings, it involves a tedious and manual process in which a human operator must move the rangefinder setup by small, precise increments repeatedly. A more elegant calibration scheme should be implemented. This will be left as possible future work.

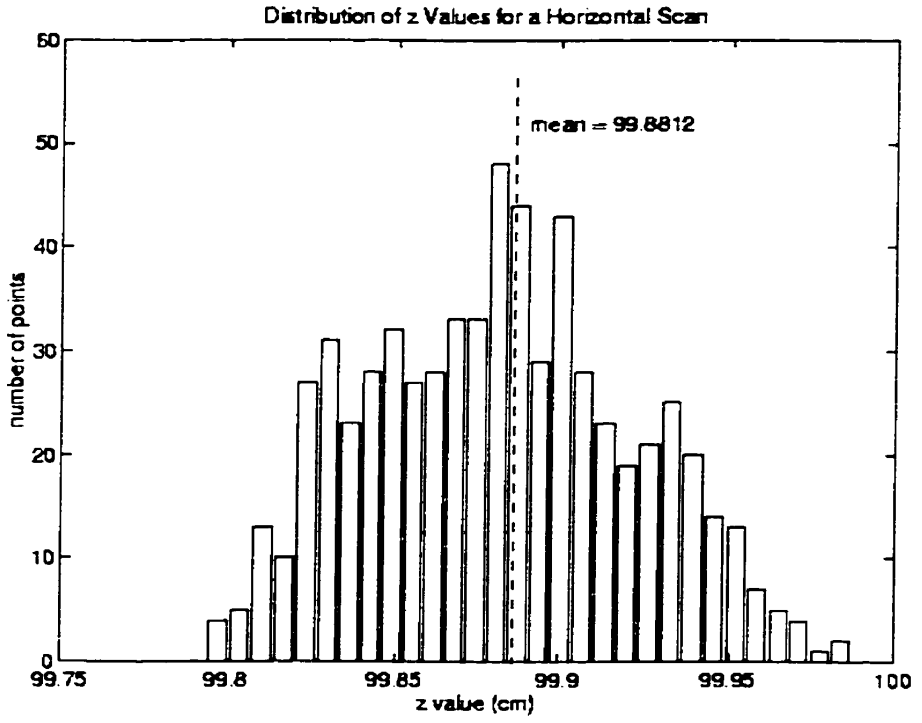


FIGURE 3.9. Distribution of z values for a scan at 1 meter.

3. The Pan-Tilt Unit and the Colour Camera

The laser rangefinder is mounted on a Pan-Tilt Unit[1](PTU) to give it two degrees of freedom in rotation(see Figure 3.2). The PTU has a maximum angular velocity of 300°/second, and a resolution of 0.05°. It also comes with a Pan-Tilt controller, which accepts commands via an RS-232 connection from a host computer (in this case, the robot's on-board computer), and a C programmers interface, which enables the user to interact with the PTU through software.

The colour camera on the rangefinder(see Figure 3.2) is used to establish focus of attention, which will be discussed in the next chapter. It is mounted directly above the black and white camera, and the two cameras have the same geometric properties, i.e., their lenses have the same focal length and their CCD arrays have the same dimensions. This is to ensure a correspondance between what the color camera sees and what the black and white camera sees.

4. The Mobile Robot

Figure 3.10 shows the Nomad 200[2] mobile robot used in this experiment. The robot is a three servo, three wheel synchronous drive non-holonomic system. It has two degrees of freedom, translation and rotation, so it can move on the x-y plane by first rotating to a desired bearing, and then moving forward. The turret of the robot can also rotate independently. The Nomad 200 has a maximum translational speed of 50cm per second and a maximum rotational speed of 60° per second. It consists of 3 sensory systems: a sonar ranging system, a tactile system, and an infrared proximity system. (The vision system shown in Figure 3.10 was implemented and added onto the robot for this research).

The sonar ranging system is a time of flight ranging sensor composed of a ring of 16 transducers. It can give range information from 44cm to 650cm with 1% accuracy

[1]The Pan-Tilt Unit is manufactured by Directed Perception, Inc. 1451 Capuchino Ave., Burlingame, California. 94010 (415)342-9399

[2]The Nomad 200 is manufactured by Nomadic Technologies Inc., 2133 Leghorn Street, Mountain View, CA 94043-1605, tel. (415) 988-7200, e-mail nomad@robots.com

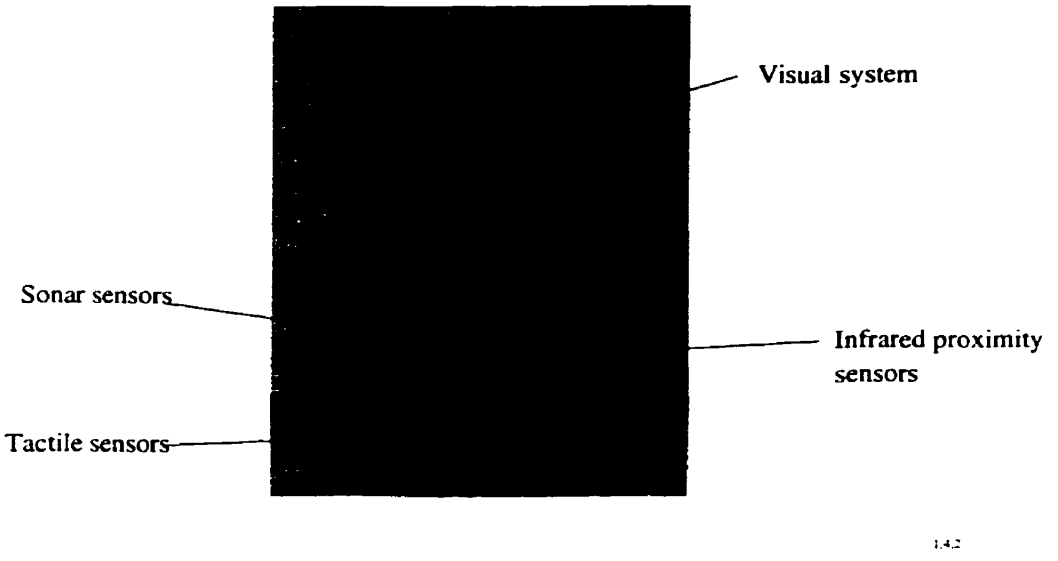


FIGURE 3.10. The mobile robot.

over the entire range. The tactile system consists of 20 independent pressure sensitive sensors, placed uniformly around the robot's bumper. Each sensor has sensitivity of about 8 ounces. The tactile system is intended to protect the Nomad 200 from damage in accidental collisions with obstacles. The infrared proximity system has 16 sensors which determine range by measuring the intensity of reflected infrared signals. The system gives range information of up to 60cm under the proper conditions. Due to its relatively short range of operation, the infrared proximity sensor system is intended mainly to warn the robot of imminent collisions with obstacles.

The Nomad 200 robot, like most other mobile robots, is equipped with different types of sensors because each type of sensor can compensate for the inadequacies of others. Ideally, the robot control architecture should make use of all available sensor readings. SPOTT [**67**] is such an architecture. However, due to the speed requirements, we have chosen a simpler, faster navigation algorithm which only makes use of sonar data and colour video signals.

5. Summary

This chapter has presented the basic concept of triangulation-based range sensing, the implementation and calibration of our rangefinder, as well as other hardware components used in this research (i.e., the colour camera, the Pan-Tilt Unit, and the mobile robot). The rangefinder is composed of a black and white camera, two laser projectors, an optical filter and positioning tools, all of which are off-the-shelf commercial products. The colour camera is mounted on the laser rangefinder. It's function is to establish focus of attention, the details of which will be given in the following chapter. The rangefinder is in turn mounted onto the PTU, which makes camera movement possible.

All information from the hardware components are first sent to the robot's on-board computer. Such information include video imagery from the cameras, current position of the PTU, and the switching of the laser projectors. Since the processing power of the robot's computer is limited, the information is sent via wireless transmission to a Silicon Graphics Indy computer, where all the computations are done. Figure 3.11 illustrates the interconnections between the hardware components.

The assembly and calibration of the vision system are part of this research. The development and maintainance of the mobile robot, on the other hand, is an on-going effort of the Mobile Robotics Laboratory in the Centre for Intelligent Machines, McGill University.

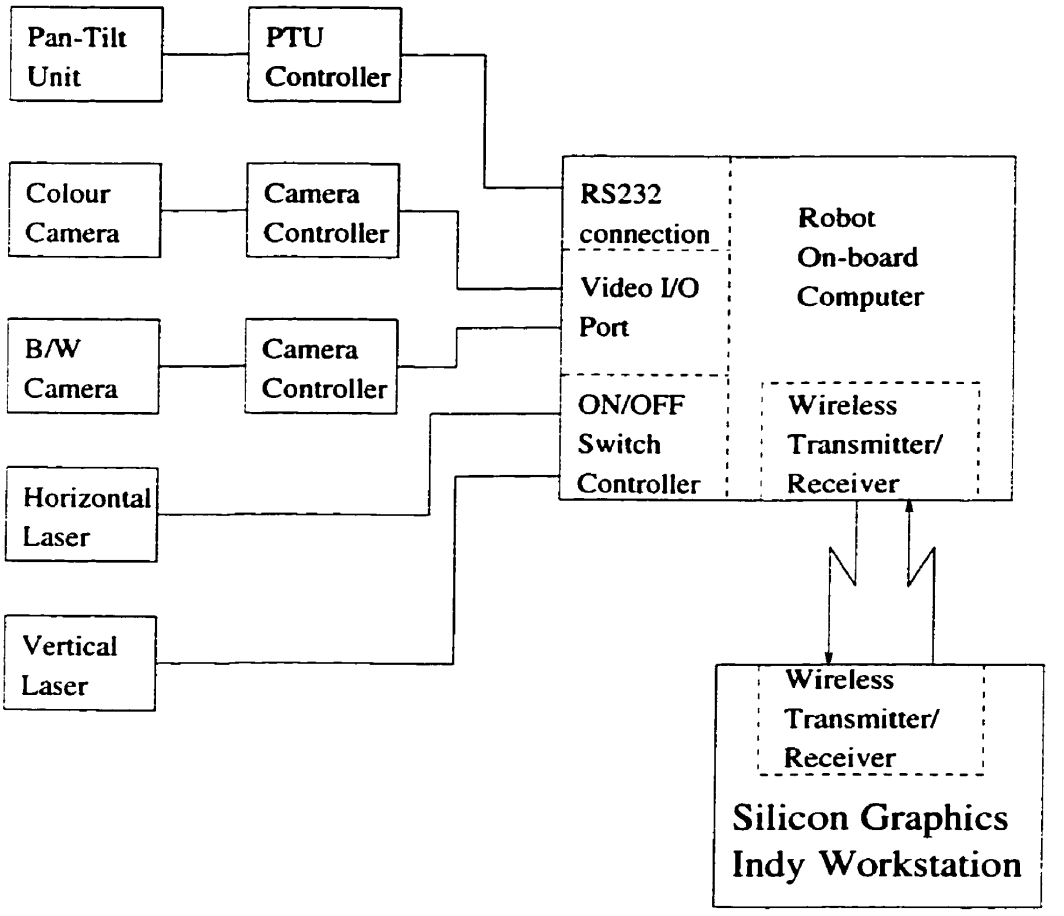


FIGURE 3.11. Interconnections between the hardware components.

CHAPTER 4

Implementation of the Active Vision Module

As mentioned in Chapter 2, the implementation of our vision system can be divided into two main modules: the active vision module which involves controlling strategies for acquiring a good view of the objects of interest, and the object recognition module which involves the use of range data to identify geons. This chapter deals with the former. The first section of this chapter describes the focus of attention mechanism, namely the colour image processing algorithm used to extract the geons from their surroundings. Section 2 describes how the geons' locations are inferred, and how the proper placement of laser stripes is achieved. Finally, the mobile robot navigation algorithm is presented in Section 3.

1. Focus of Attention

The objective of the focus of attention is to locate the geons in the robot's surroundings. Colour is chosen as the feature for cuing visual attention. All the geons are painted uniformly in a predetermined colour, and the goal is to extract pixels belonging to that colour from the colour image. In essence, this is a problem of colour segmentation [**15, 42, 53, 60**].

1.1. Colour Space. The performance of a colour segmentation algorithm is strongly dependent on the choice of the colour space in which segmentation is performed. Each colour space has its pros and cons, depending on the application. For instance, the HSI (Hue, Saturation, and Intensity) space is convenient for representing human colour perception, while the YIQ (Luminance, In-phase chrominance, and Quadrature chrominance) space is efficient for encoding colour information in TV signals [**42**].

The majority of colour image capture devices represent colour images by assigning each pixel a R (red), G (green), and B (Blue) value. The RGB representation of an image can then be transformed into other colour spaces for further processing. The choice of colour space therefore implies the choice of transformation from the RGB space to the desired colour space. There are two types of colour space transformation: linear and nonlinear. A linear transformation takes the form of:

$$I_1 = c_{11}R + c_{12}B + c_{13}G \tag{4.1}$$

$$I_2 = c_{21}R + c_{22}B + c_{23}G \tag{4.2}$$

$$I_3 = c_{31}R + c_{32}B + c_{33}G \tag{4.3}$$

where c_{ij} are constants.

For example, the RGB-to-YIQ transformation is [**52**]:

$$Y = 0.299R + 0.114B + 0.587G \tag{4.4}$$

$$I = 0.596R - 0.322B - 0.274G \tag{4.5}$$

$$Q = 0.211R + 0.312B - 0.523G \tag{4.6}$$

An example of a nonlinear transformation is the RGB-to-HSI transformation shown below:

$$H = arctan2(\sqrt{3}(G-B),(2R-G-B)) \tag{4.7}$$

$$S = 1 - 3min(\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}) \tag{4.8}$$

$$I = \frac{R+G+B}{3} \tag{4.9}$$

Ohta et al. [**42**] have made a comparative study of more than 100 different colour spaces by testing a recursive thresholding segmentation algorithm in each colour space. Ideally, a similar study should be done to determine the optimal colour space to perform our colour segmentation. However, the speed of the overall process would be seriously taxed by the computational requirements for a colour space transformation. For instance, the RGB-HSI transformation described in equations(4.7 - 4.9) calls for the calculation of an arctangent and a square root, both of which involve relatively intensive computations. To do the tranformation on a 640 × 480 colour image, these operations must be carried out 307,200 times. An analysis shows that the RGB-HSI transformation on a 640 × 480 image would take 3.39 seconds for a Silicon Graphics Indy workstation. Even a linear transformation, like the RGB-YIQ transformation, requires a considerable amount of "number-crunching". The RGB-YIQ transformation of a 640 × 480 image takes 1.15 seconds on the same workstation. The whole process of searching for a geon and navigating towards it would likely require the analysis of tens of colour images, and the time required to do the colour space transformation would add up quickly. Also, experiments show that segmentation in the RGB space gives acceptable performance. Therefore, for the sake of speed and simplicity, we have chosen the RGB colour space to conduct colour image analysis and segmentation.

1.2. Colour Segmentation. The purpose of colour segmentation is to partition an image into meaningful regions based on the colour characteristics of the scene. There are two variations to the problem of colour segmentation [**53**]. The first applies to situations where some particular colour space characteristics are known a

priori and the goal is to detect regions which satisfy those characteristics. The second variation arises when there is no a priori knowledge regarding colour, and the goal is to segment the image into regions with similar colour space characteristics. The latter often calls for sophisticated thresholding, clustering or region growing techniques since little or no human interaction is permissible. Since we have a priori knowledge of the geons' colour, our problem belongs to the first category. Our task is further simplified by the fact that we only need to segment the geons' colour from the rest of the scene, so essentially we are dealing with the problem of binarizing an image in colour space.

Binarizing a 1D histogram of a *grayscale* image involves determining the lower threshold value t_L and the upper threshold value t_H (see Figure 4.1(a)). All pixels with grayscale value between t_L and t_H are considered foreground, and the rest are considered background. This can be extended to a 2-D histogram in which the foreground is defined by a rectangular box, whose sides are bounded by $t_{1L}, t_{1H}, t_{2L}, t_{2H}$, as shown in Figure 4.1(b). By the same reasoning, a 3-D histogram (colour histogram) can be thresholded by defining a lower and upper threshold value on each of the R, G, and B axes. The resulting thresholded region will take the shape of a rectangular parallelpiped. It may not be appropriate to segment multi-dimensional data by thresholding each 1-D axis independently. Vector quantization or Voronoi tessellation is better suited for segmentation because the multi-dimensionality of data is taken into account. Figure 4.2 illustrates how vector quantization differs from simple thresholding in the 2-D case.

The mobile robotics team at the Georgia Institute of Technology has used a very simple method to distinguish red, green, and blue objects [**3**]. Given an RGB image, the "super-components" of red, green, and blue are extracted. The super-component of a colour is computed by subtracting the values of the other two components at each pixel, e.g., super-red is red - (blue + green), super-green is green - (blue + red). Once the super-components of each pixel have been calculated, the detection of a colour can be done by simply thresholding the corresponding super-component. For instance, to

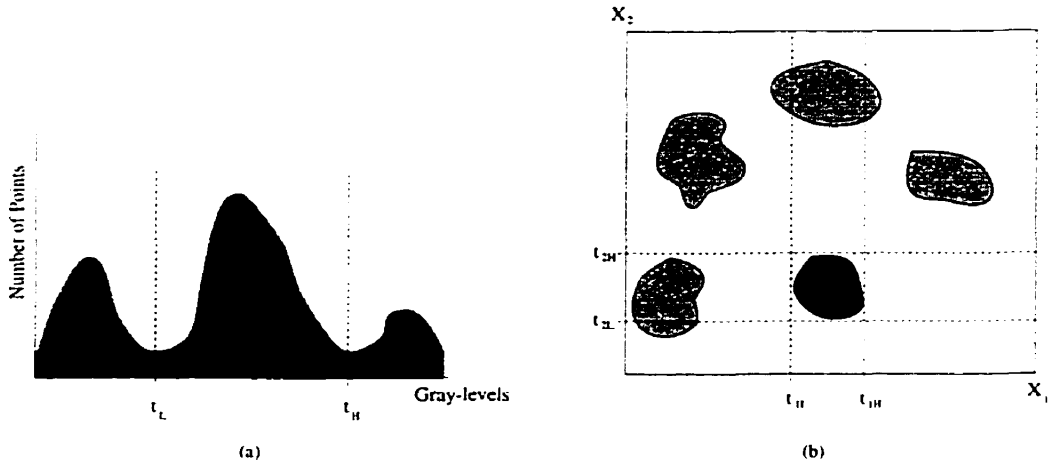


FIGURE 4.1. (a)Segmentation of a 1-D histogram. (b)Segmentation of a 2-D histogram

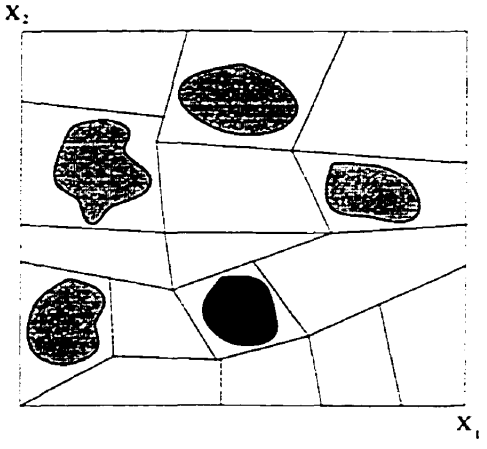


FIGURE 4.2. Segmentation using vector quantization

extract red objects from a scene, the super-red value for each pixel is calculated. If a pixel's super-red value is greater than a threshold value t_r, i.e. $r - (g + b) > t_r$, then the pixel is considered red. This method is simpler than the multi-dimensional thresholding method because only one threshold value needs to be defined, compared to the six threshold values needed for 3-D thresholding (a lower threshold t_L and a higher threshold t_H for each of the R, G, and B axis). However, the super-component approach has a serious drawback: only the three primary colours (red, green and blue) can be extracted. A super-yellow component, for example, cannot be extracted unless a colour space transformation is done in advance.

(a) (b)

FIGURE 4.3. (a) Geon placed against a black background. (b) Thresholded image of (a)

In our research, a simple 3-D thresholding method was used to perform colour segmentation. A geon is placed against a black background, as shown in Figure 4.3(a) and its image is captured. The image is then segmented into foreground (the geon) and background using the super-red method mentioned above (see Figure 4.3(b)). The threshold t_r is found by trial-and-error. In this case, segmentation was done using the threshold $r \geq b + g + 20$, where $r, g, b \in [0, 255]$. Although this threshold is successful in segmenting the image in Figure 4.3(a), it is not restrictive enough and thus may not work for other backgrounds that are more rich in colour, i.e., it may mistake many other coloured objects in the background as part of the foreground. To further refine the thresholded region, a colour histogram of the geon is plotted (see Figure 4.4(a)). The cluster of points in the colour histogram represents the colour of each pixel in the image shown in Figure 4.3(a). By observing the 2-D projections of the colour histogram (Figure 4.4(b)-(d)), thresholds can be selected such that the thresholded region contains most of the points. In this case, the thresholds are:

$$b < 0.467r - 7 \tag{4.10}$$

$$b < 75 \tag{4.11}$$

$$r > 50 \tag{4.12}$$

$$b < 1.333g + 10 \tag{4.13}$$

$$b > 1.333g - 86.7 \tag{4.14}$$

$$g < 115 \tag{4.15}$$

$$g < 0.639r - 12.78 \tag{4.16}$$

$$g > 0.381r - 45.7 \tag{4.17}$$

Coloured objects are subject to slight variations in colour under different lighting conditions [**25, 26**]. Figure 4.5(a) shows the colour histogram of the same scene taken simultaneously under both fluorescent light and sunlight. The colour variation is evident when one compares the 2-D projections of the histogram shown in Figure 4.5(b)-(d) with those in Figure 4.4(b)-(d). To take into account this colour shift, the thresholded region must be enlarged to include most of the points under both lighting conditions. (The case in which only sunlight is present is not considered since we assume that we have control over the illumination). In order to accomodate both scenarios, the threshold region must be made larger so that most of the points in the histogram shown in Figure 4.4 and the one in Figure 4.5 are included. The new thresholds are shown below:

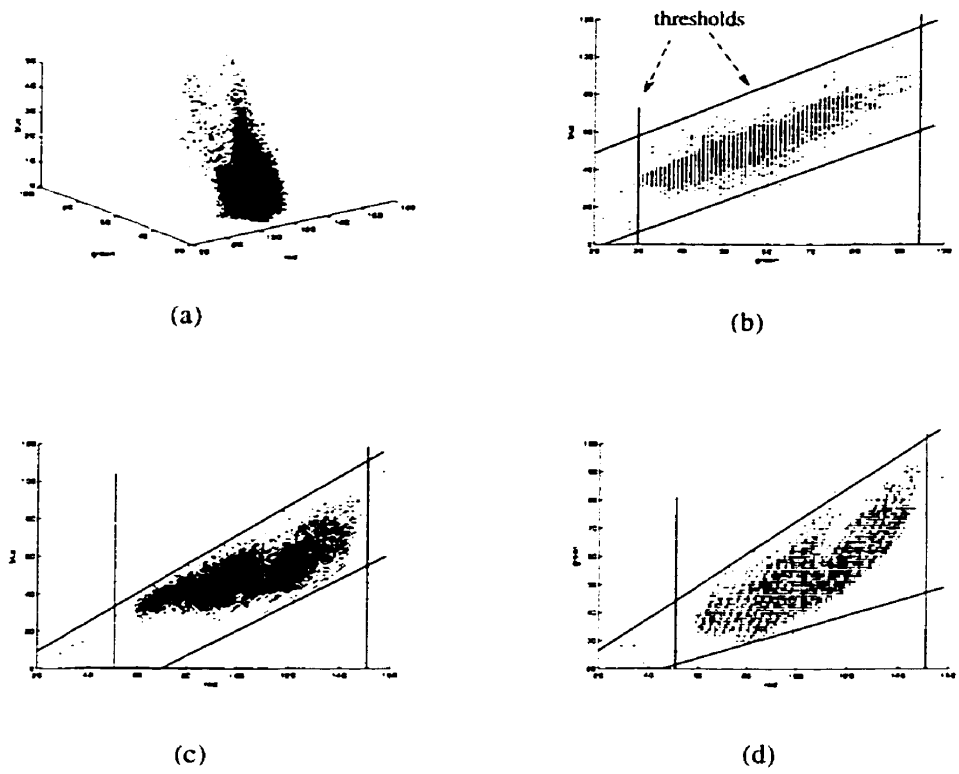


FIGURE 4.4. (a)The 3-D colour histogram of the geon. (b)Projection of the 3-D histogram on the blue-green plane. (c)Projection of the 3-D histogram on the red-green plane. (d)Projection of the 3-D histogram on the red-blue plane.

$$b < 0.75r - 5 \tag{4.18}$$

$$b < 80 \tag{4.19}$$

$$r > 50 \tag{4.20}$$

$$b < 1.333g + 10 \tag{4.21}$$

$$b > 0.67g - 40 \tag{4.22}$$

$$g < 100 \tag{4.23}$$

$$b < 0.44r - 50.6 \tag{4.24}$$

$$g > 30 \tag{4.25}$$

$$g < 0.58r + 25.2 \tag{4.26}$$

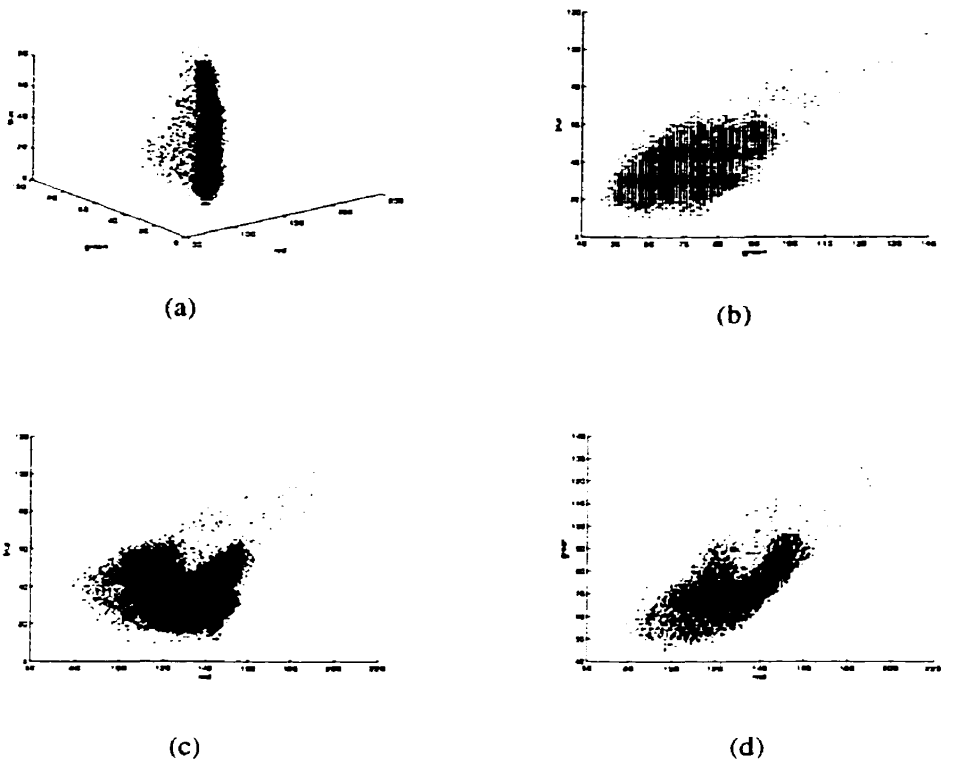


FIGURE 4.5. (a)The 3-D colour histogram of the geon under sunlight and fluorescent light. (b)Projection of the 3-D histogram on the blue-green plane. (c)Projection of the 3-D histogram on the red-green plane. (d)Projection of the 3-D histogram on the red-blue plane.

Even the best colour segmentation algorithms sometimes fail to separate target objects from their background. The most common causes of failure are variations in lighting conditions, specular reflections, and shadows. Many colour constancy algorithms have been proposed [**25, 26, 32, 58**], but the problem of colour constancy remains largely unsolved. Most schemes proposed in the field of colour segmentation assume target objects rich in colour information, which is not the case in this research. Segmentation in a different colour space may improve performance, but we chose not to do any colour space transformation in order to limit the computational time.

Clustering and region growing methods can also be employed to achieve more accurate colour segmentation [**15, 42, 53, 60**]. Those methods were not considered, again due to the issue of speed.

2. Locating the Geons

The previous section described how a geon is extracted from a cluttered scene, which is essentially how the focus of attention problem is solved. This section addresses the issue of how the location of the geon, in particular its bearing and distance with respect to the robot, can be deduced from the information obtained from the last section.

2.1. Data Preprocessing. The binary image obtained after the colour segmentation is likely to contain noise. The noise may be intrinsic to the scene, due to random variations in the RGB values of each pixel (see Figure 4.6). This kind of noise is usually only a few pixels in size. To alleviate the effect of this noise, a morphological erosion [**52**] operation is performed on the image. In a morphological erosion operation, any foreground pixel that has one or more background pixel connected to it is rejected and labelled as a background pixel. As a result, any blobs that are only a few pixels in size, or any lines less than 3 pixels wide will be filtered out.

Another scenario is when some other object in the background has the same colour as the geon by coincidence (Figure 4.7). Blob analysis is used to solve this problem. A blob is a group of pixels that are connected. When more than one blob is present in the image, the largest blob is taken as the geon and all other blobs are discarded. We assume that all background objects with the same colour as the geon are smaller than the geon (or at least appear smaller in the image).

The measures mentioned above may still fail to separate the geon from noise or other background objects with similar colour. Two scenarios may arise: an object with the same colour as the geon may appear to be overlapping the geon from the camera's point of view; thus the geon and the object will be considered as one single blob. Another case is when there is an object of the same colour as the geon and the

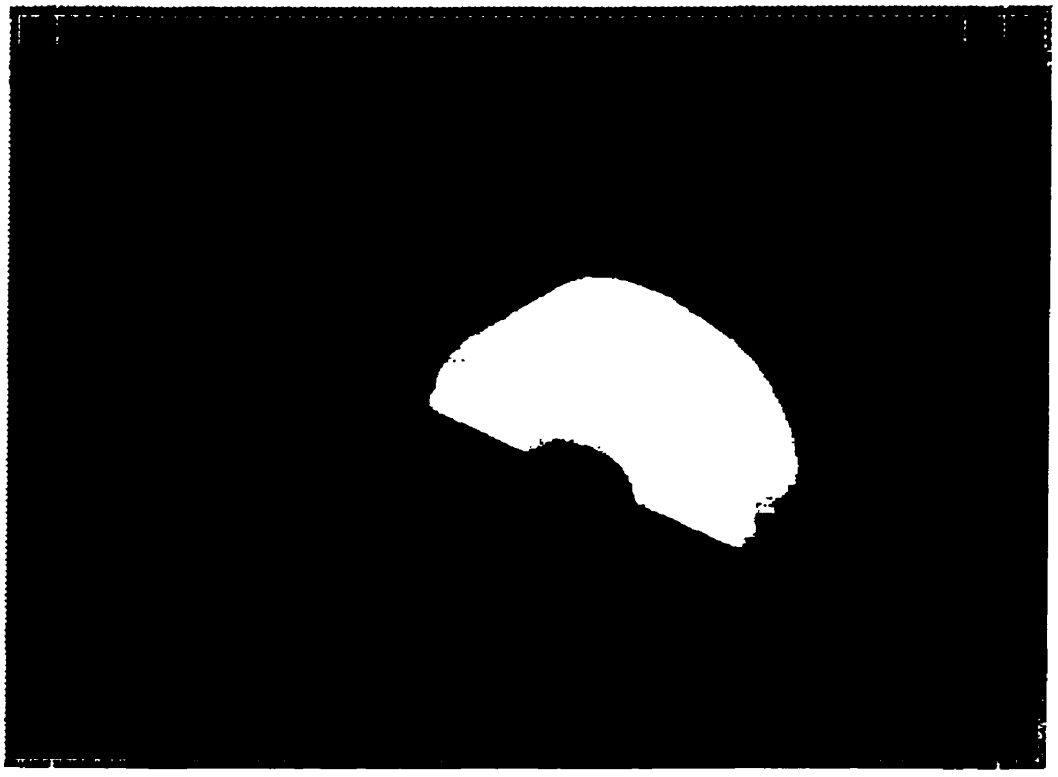
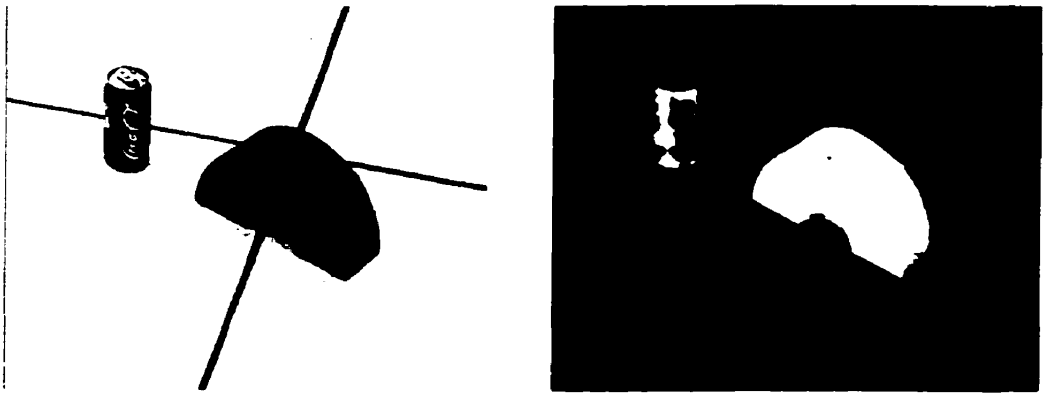


FIGURE 4.6. The presence of random noise in the thresholded image



(a) (b)

FIGURE 4.7. (a) Geon placed near a background object with similar colour. (b) Binarized image of (a).

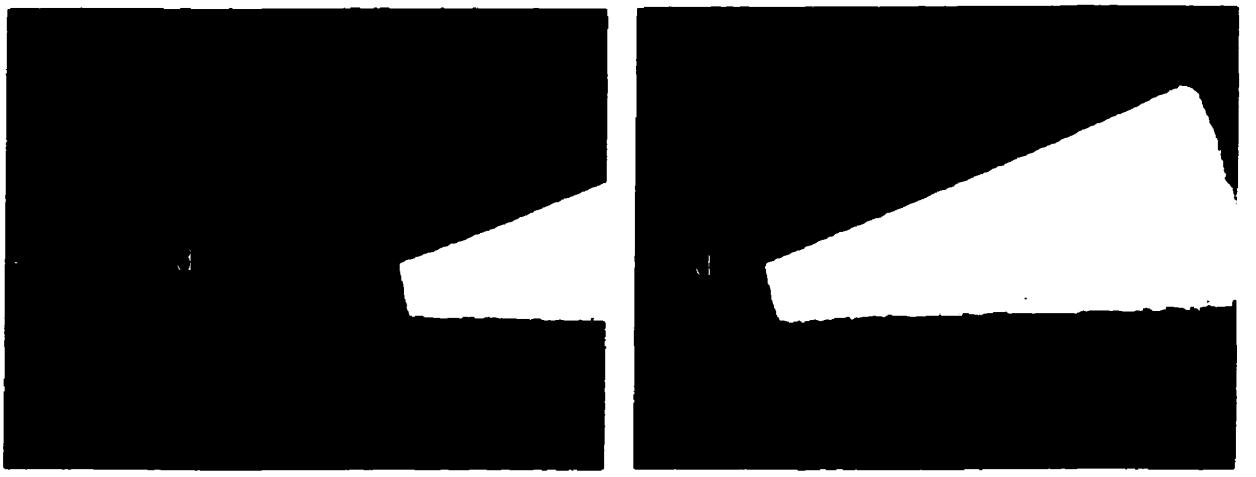
object appears larger than the geon. In this case the object will be considered as the geon, while the real geon will be discarded as background. There appears to be no easy way around these two problems. We chose to ignore these special cases because

the chance of them occuring under the circumstances of our experiment was quite slim.

2.2. Visual Search. As will be addressed in the next section, the robot navigation process will presumably bring the robot within a distance of about 1 meter from the geon, and the vision system will also be roughly facing the geon. Once this point has been attained, two tasks remain for the active vision module: to bring the geon within the camera's field of view, and to deduce the location of the geon.

Although we can assume that the vision system will be pointing towards the direction of the geon, it may still lie outside the camera's field of view, especially since a narrow angle lens is used for our camera in order to minimize spherical lens distortion. Some sort of camera movement is therefore needed to ensure detection of the geon. A simple search is performed by moving the camera with the Pan-Tilt Unit (PTU) to cover a large area in front of the vision system. After every PTU movement the colour camera captures an image and performs colour segmentation. Once the segmented image shows a blob large enough to be considered a geon, the PTU movement stops. At this point, the geon may be only partially inside the camera's field of view, as shown in Figure 4.8(a). If this is the case, further camera movements will be required. For example, in the case shown in Figure 4.8(a), where the blob occupies the rightmost columns in the image, the camera must pan clockwise in order to bring the entire geon within field of view. Likewise, if the blob lies on the topmost edge of the image, the camera must tilt up, and vice versa. The question is by how much the camera should pan or tilt.

Referring again to Figure 4.8(a), the horizontal distance d (in pixels) between the leftmost edge of the image and the leftmost pixel of the blob can be used to determine the desired camera pan angle to bring the geon within view. If we can assume that spherical lens distortion is negligible (which, as mentioned earlier, is a reasonable assumption in our case), then θ_d, the angle subtended by d (See Figure 4.8(a)) can be determined by the following equation:



(a) (b)

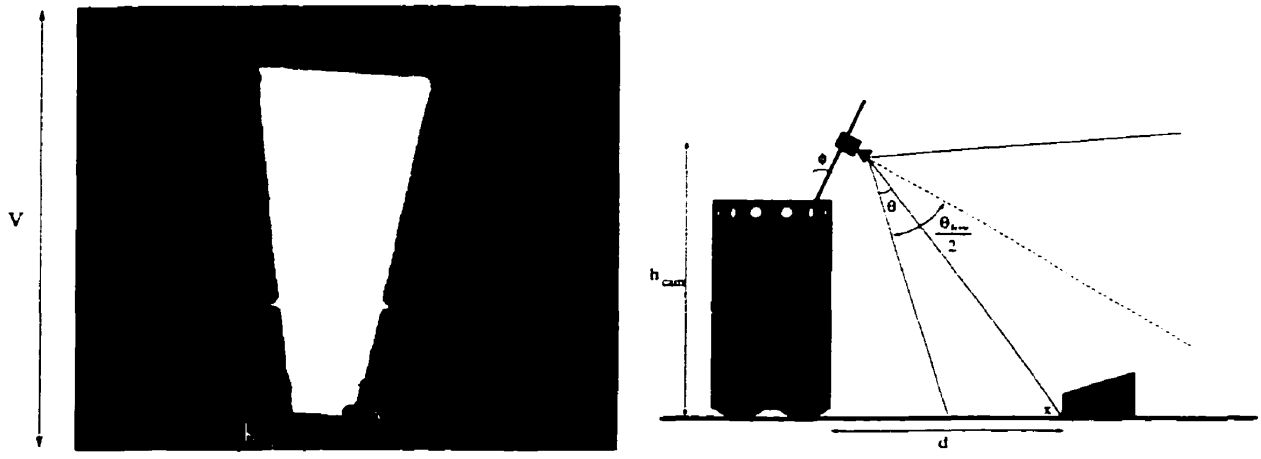
FIGURE 4.8. (a)Geon partially within camera's field of view. (b)Image of the same geon after one iteration of adjustment.

$$\theta_d = \frac{d}{h} \times \theta_{fov} \tag{4.27}$$

where θ_d is the angle subtended by d, θ_{fov} is the angle subtended by the camera's field of view, and h is the total number of columns of pixels in the image.

Once θ_d is obtained, we know that the camera can be panned clockwise by any angle smaller than θ_d in order to bring a larger portion, if not the entirety of the geon into view. We chose to pan the camera by $(0.75 \times \theta_d)^\circ$ each time. Figure 4.8(b) shows the image of the geon after the camera adjustment. If the geon is still partially out of view, which appears to be the case here, the new θ_d will be calculated and the above-mentioned procedure will be reiterated. If the object remains partially out of view after several iterations, or if the object occupies both the leftmost and rightmost edges of the image, then the object is too large to fit into the camera's field of view, and the robot will be required to move backwards a little to reattempt the visual search.

The case in which the blob is lying on the top or bottom edge of the image is handled in very much the same way and therefore will not be elaborated here.



(a) (b)

FIGURE 4.9. Distance between robot and geon

2.3. Determining Distance. With the entire geon within view, the horizontal distance between the robot and the geon can be determined quite easily. As illustrated in Figure 4.9, the bottommost point of the blob usually corresponds to the point on the geon closest to the base of the robot, and that point is usually in contact with the floor. That point, designated as point x, will be used as a reference point. The horizontal distance d between point x and the base of the robot is:

$$d = h_{cam} \times tan((\frac{\pi}{2} - \phi) - \frac{\theta_{fov}}{2} + \theta) \tag{4.28}$$

where h_{cam} is the vertical height of the colour camera with respect to the floor, ϕ is the tilt angle of the PTU, θ_{fov} is the angle subtended by the camera's vertical field of view, and θ is the angular difference between the bottom of the image and point x. As pointed out before, θ can be calculated using the equation (refer to Figure 4.9(a) for variable definitions):

$$\theta = \frac{k}{v} \times \theta_{fov} \tag{4.29}$$

The PTU's tilt angle ϕ can be easily obtained by calling a PTU library function.

Although simple and robust, this method of distance estimation can only offer limited accuracy. Experiments show that it gives an error of about $\pm 5\%$ at a distance of 100cm. Several sources may contribute to this error: firstly, the tilt angle of the PTU, given by a PTU software library function, is of limited precision; secondly, the calculation of θ in equation 4.29 ignores the effect of spherical lens distortion, albeit a small one; thirdly, the binarized image that shows the geon as a blob, like the one in Figure 4.10(a), may not portray the geon accurately. Specular reflections and shadowing often render the geon partially undetected, as illustrated in Figure 4.10(b). Nevertheless, this method was adopted because the recognition process does not require a high degree of accuracy in the measurement of the distance. The distance is used to ensure that the laser rangefinder is within its operating range (the rangefinder was calibrated up to a range of 2 meters), and is also used to determine the placement of the laser scans, which will be discussed next.

(a) (b)

FIGURE 4.10. (a)Geon under the influence of specular reflections. (b) Binarized image of (a).

2.4. Positioning of Laser Stripes. The recognition algorithm proposed in this thesis relies on strategic placement of laser stripes on the geons. The 3 horizontal laser stripes are placed so that the geon is divided vertically into four equal portions, and the 3 vertical stripes are placed such that the geon is divided equally into four horizontal portions. This is illustrated in Figure 4.11.

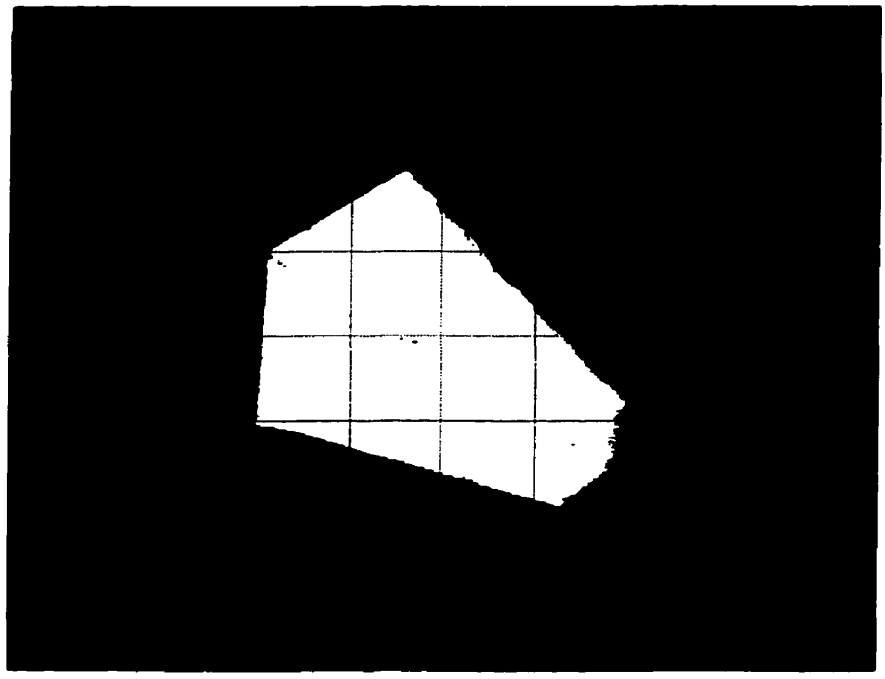


FIGURE 4.11. Desired locations of laser stripes on the geon.

The difficulty stems from the fact that the laser projectors are not aligned with the optical axis of the camera. Therefore some geometric calculations are required to determine by how much the PTU should pan or tilt to accomplish proper laser stripe placement.

The first step is to establish a "bounding box" around the geon in the binarized image (see Figure 4.12(a)). The horizontal laser stripes h_1, h_2, and h_3 should divide y into four equal parts. Figure 4.12(b) shows the side view of the system. If θ_y is the angle subtended by y, and θ_{h1}, θ_{h2}, θ_{h3} is the angular distance between point p and h_1, h_2, and h_3, respectively, then:

$$\theta_{h1} = 0.75\theta_y \tag{4.30}$$

$$\theta_{h2} = 0.5\theta_y \tag{4.31}$$

$$\theta_{h3} = 0.25\theta_y \tag{4.32}$$

$$\theta_y = \frac{y}{d_y} \times \theta_{v_{fov}} \tag{4.33}$$

where $\theta_{v_{fov}}$ is the angle subtended by the camera's vertical field of view.

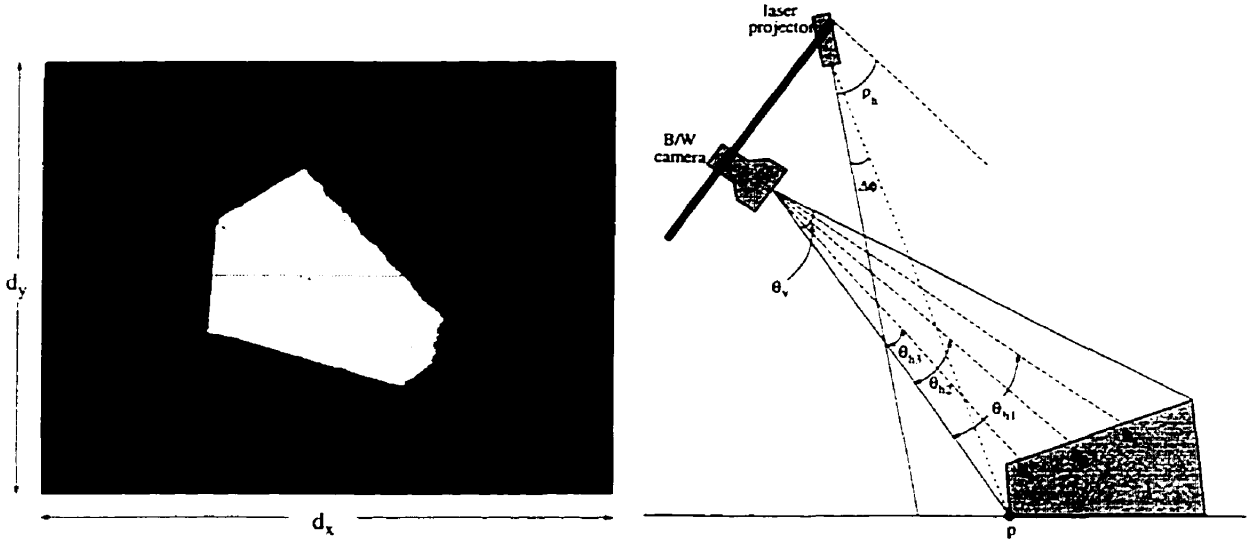


FIGURE 4.12. (a)Desired locations of the horizontal laser scans. (b)Side view of the vision system and the geon.

The next step is to calculate $\Delta\phi$ (see Figure 4.13), which is the angular difference between point p (the bottom of the geon) and the current projection of the horizontal

laser. Using simple trigonometry, β_1 and β_2 can be determined by equations 4.34 and 4.35:

$$\beta_1 = arctan(\frac{h_{PTU} + h_{las}sin\phi}{d - h_{las}cos\phi}) \tag{4.34}$$

$$\beta_2 = \frac{\pi}{2} - (\frac{\pi}{2} - \phi - \rho_h) = \phi + \rho_h \tag{4.35}$$

Subsequently, $\Delta\phi$ can be determined by the following equation:

$$\Delta\phi = \pi - \beta_1 - (\pi - \beta_2) = -\beta_1 + \beta_2 \tag{4.36}$$

The desired tilt angles for h_1, h_2, and h_3 are simply $\Delta\phi + \theta_{h1}$, $\Delta\phi + \theta_{h2}$, and $\Delta\phi + \theta_{h3}$, respectively.

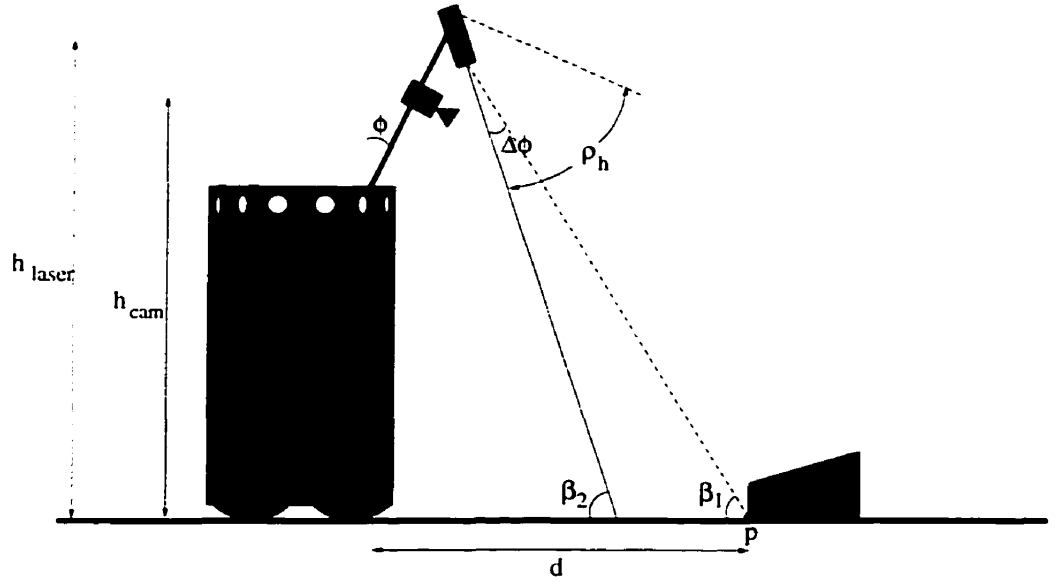


FIGURE 4.13. Illustration of how the proper placement of horizontal laser stripes can be inferred.

The placement of the vertical laser stripes also relies on geometric calculations. Figure 4.14(a) shows the top view of the vertical rangefinder's layout. The angle

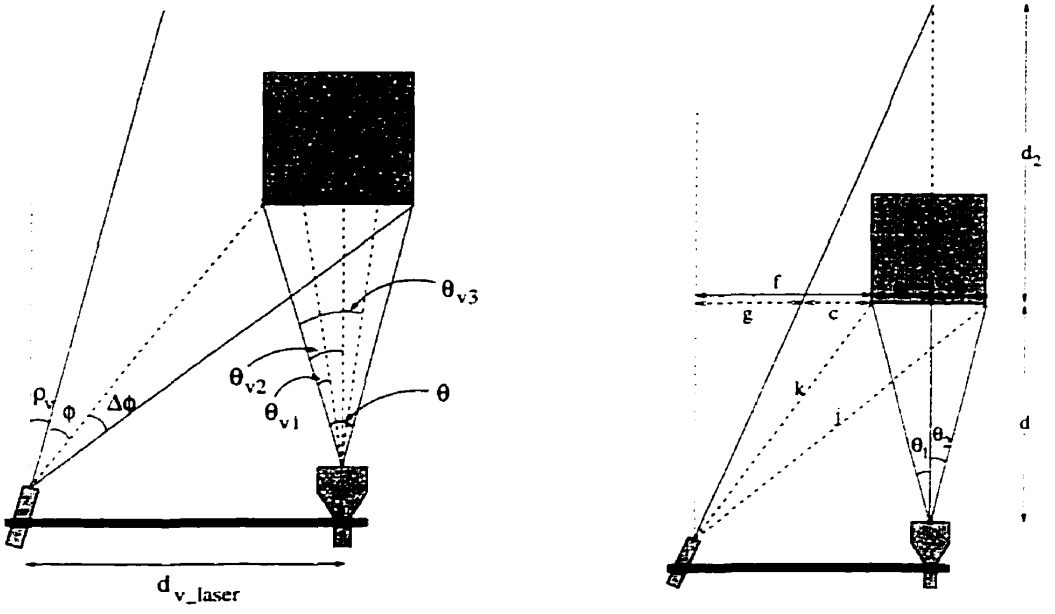


FIGURE 4.14. Definitions of geometric variables for the vertical laser scans.

spanned by the geon relative to the camera is θ, which can be calculated using the equation below:

$$\theta = \frac{a+b}{d_x} \times \theta_{fov} \tag{4.37}$$

where d_x is the width of the image (in pixels), and θ_{fov} is the angle subtended by the camera's horizontal field of view. d, the distance between the geon and the camera, can be derived using the method described in Section 2.3. ϕ is the angular difference between the laser projector's current direction and the leftmost point of the geon. $\Delta\phi$ is the angle spanned by the geon relative to the vertical laser projector. The following equations are used to calculate ϕ and $\Delta\phi$ (Refer to Figure 4.14(b) for variable definitions):

$$d_2 = \frac{d_v}{tan(\rho_v)} - d \tag{4.38}$$

$$c = d_2 tan(\rho_v) - dtan(\theta_1) \tag{4.39}$$

$$g = dtan(\rho_v) \tag{4.40}$$

$$f = g + c \tag{4.41}$$

$$\phi = arctan(\frac{f}{d}) - \rho_v \tag{4.42}$$

$$\psi = arctan(\frac{d}{d_v}) \tag{4.43}$$

$$\Delta\phi = \frac{\pi}{2} - \rho_v - \phi - \psi \tag{4.44}$$

The desired pan-angles are simply:

$$\phi_{v1} = \phi + 0.25\Delta\phi \tag{4.45}$$

$$\phi_{v2} = \phi + 0.5\Delta\phi \tag{4.46}$$

$$\phi_{v3} = \phi + 0.75\Delta\phi \tag{4.47}$$

To summarize, this section has presented the method for determining the three tilt angles θ_{h1}, θ_{h2}, θ_{h3}, and the pan angles ϕ_{v1}, ϕ_{v2}, ϕ_{v3}, which will place the six laser stripes onto the desired locations of the geon.

3. Robot Navigation

The ability to navigate in an unknown environment is a fundamental requirement for a mobile robot. Without a priori knowledge of the environment, the robot must be able to perceive its surroundings using its sensors, and execute actions that are appropriate for that environment and the goals of the robotic system. Mobile robots often need to obtain information from a combination of several sensor systems. For example, SPOTT [**67**] combines information from sonar, laser rangefinders, infrared sensors, and tactile (bumper) sensors to perform navigation. The rationale behind using multiple sensor types is that each type of sensor has its strengths and weaknesses, and the strength of one sensor type can remedy the weakness of another.

Autonomous robot navigation is a very extensive field in its own right. Since this research is not directly related to robot navigation, a detailed analysis of this topic will not be presented in this thesis. Instead, we will give a description of the navigation algorithm used in the experiments.

The role of robot navigation in this research is to bring the robot to the vicinity of a geon, if one exists in the surroundings. Once this is accomplished, the vision system will take over. First, it will perform a visual search of the geon. Then, upon detection of the geon, it will attempt to infer the geon's identity based on range data.

The navigation system[1] employs the ring of sonar sensors on the robot (see Figure 3.10), and a colour camera mounted on the robot such that it points towards the floor at an oblique angle (this is a different camera from the one on our vision system). The camera is equipped with a wide angle lens to give it a broad field of view. The

[1]The navigation system used in this research was implemented by François Bélair, Deep Jugessur, and Robert Sim of the Mobile Robotics Laboratory at the Centre for Intelligent Machines, McGill University.

robot perceives its environment through colour images, and range information from the sonar. Three assumptions are made in this navigation scheme:

(i) The colour of the floor is uniform and is known a priori.

(ii) The colour of the target (geon) is known a priori.

(iii) Any object whose colour is not the same as the floor's colour is an obstacle. (Note: A target is also considered as and obstacle.)

Once the colour camera detects a colour different from that of the floor, it will be registered as an obstacle, and its location determined through a geometric transformation. Note that in this case, any target is also an obstacle, since the target is a geon, and collision between the robot and the geon is undesirable. Sonar is also used for obstacle detection. The advantage of using sonar is that it can acquire range information from all directions quickly and inexpensively. The drawback is that sonar suffers from certain deficiencies such as lack of accuracy and multiple reflections [**22**]. To minimize their effect, multiple sonar readings may be taken at each step.

The next step is path planning. The goal of path planning is to find the most efficient route that brings the robot from its current position to the target position without colliding with obstacles. A *safe polygon* approach to path planning was adopted. Figure 4.15(a) illustrates the concept of safe polygon. For each obstacle in the robot's vicinity, a line, perpendicular to the shortest distance between the robot and the obstacle, is drawn. The distance between the obstacle and the line, d, is set to a value greater than the robot's radius. The line divides the plane into two regions: an admissible region, and an inadmissible region. The intersection of all the admissible regions corresponding to each obstacle produces a polygon (in this case a pentagon), as shown in Figure 4.15(b). This polygon is referred to as the *safe polygon* since the robot can move safely within it. In the case where no obstacle is detected near the robot (Figure 4.16(a)), a default bounding box of pre-defined size is applied as the safe polygon. If there exist some obstacles, but they do not form a safe polygon

(see Figure 4.16(b)), then the safe polygon is defined as the intersection between the default bounding box and the admissible regions corresponding to the obstacles.

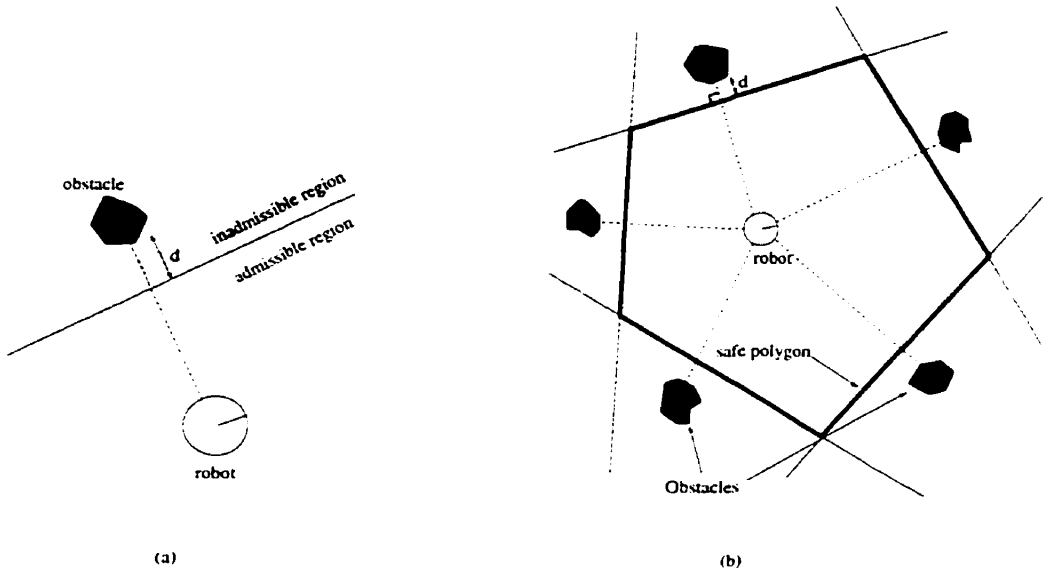


FIGURE 4.15. Illustration of the safe polygon concept.

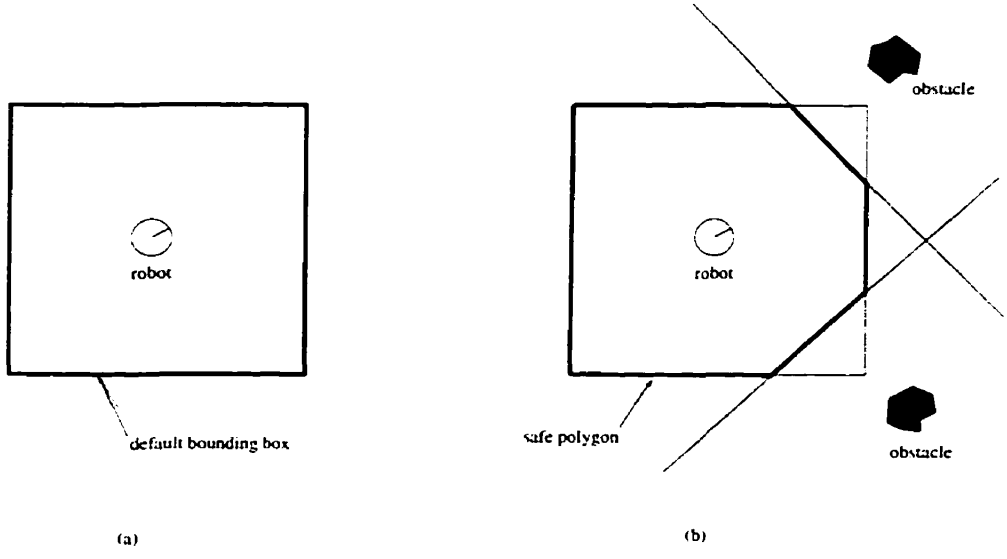


FIGURE 4.16. Default bounding box.

To navigate towards the target, the robot moves towards the vertex on the safe polygon that is closest to the target, as shown in Figure 4.17. The safe polygon will

change shape every time the robot makes a move, so the shape of the polygon must be constantly undated. By moving to the vertex closest to the target every time, the robot will eventually be brought close enough to the target.

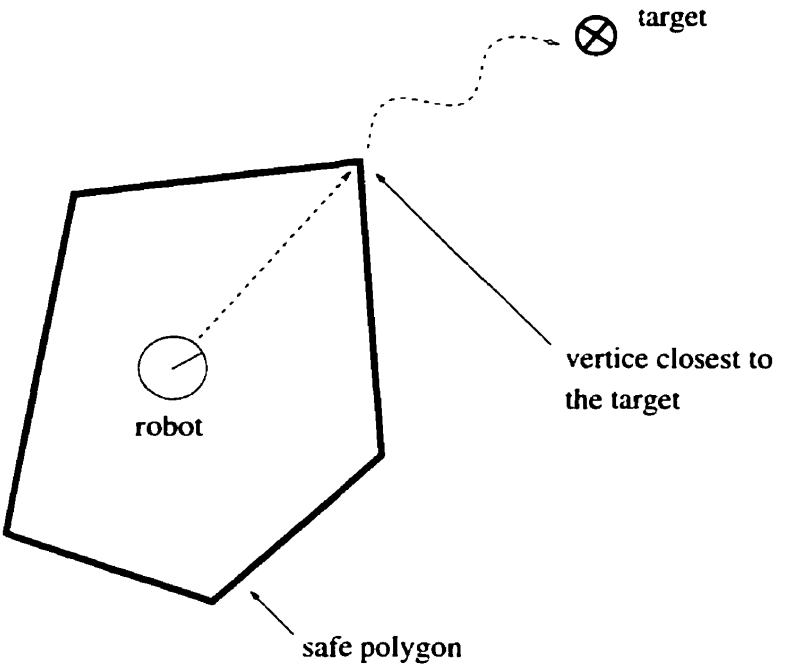


FIGURE 4.17. Choice of vertice.

In summary, we employ a simple and robust navigation scheme to bring the robot close to the geon for its recognition. The robot wanders in the environment until the geon is detected. Upon detection, the geon is registered as the target and path planning is executed. A colour camera and sonar are used to perceive the environment, and a novel safe polygon approach is used for path planning. The goal of this navigation process is to bring the robot within a certain distance (approx. 90cm) from the geon. Once that goal is attained, our active vision process can proceed to bring the geon within the rangefinder's field of view using colour segmentation and PTU movements, as described earlier in this chapter.

4. Summary

This chapter has presented the active vision module, which is responsible for locating geons in an indoor environment, navigating towards a geon, and strategically placing laser stripes onto the geon. Under the assumption that the geons are painted in a specific colour, a colour histogram segmentation method is used to extract them from the scene. A novel safe polygon path planning technique is used to bring the robot to the vicinity of the geon. Proper laser stripe placement is accomplished by geometric reasoning.

The next task, which is to identify the geon based on the acquired range data, will be addressed in the following chapter.

CHAPTER 5

The Geon Recognition Module

This chapter presents a detailed description of the geon recognition module of our vision system. Our recognition module is composed of several stages, as shown in Figure 5.1. The first section of this chapter addresses the first two stages, namely the data acquisition stage and the data preprocessing stage. Section 2 introduces the concept of evidence-based recognition. The next three sections outline the three types of evidence used in our recognition scheme, and how they are inferred. Finally, Section 6 describes the decision-making process based on available evidence.

1. Data Acquisition and Preprocessing

Section 4.2 explained how the desired PTU's pan and tilt angles for proper laser stripe placement can be calculated based on camera geometry. Therefore, to obtain the range data, the rangefinder simply needs to pan or tilt the PTU to the precalculated angles, and grab a frame each time. Range data can then be obtained from each frame using the method described in Chapter 3.

Two issues must be addressed before the range data can be used for recognition. First, smoothing must be performed to alleviate noise. There are many methods for smoothing one dimensional data, and Gaussian smoothing is probably the most widely used. One drawback to the Gaussian smoothing kernel is that in the case of spurious noise, where one or two points of the one-dimensional profile have values

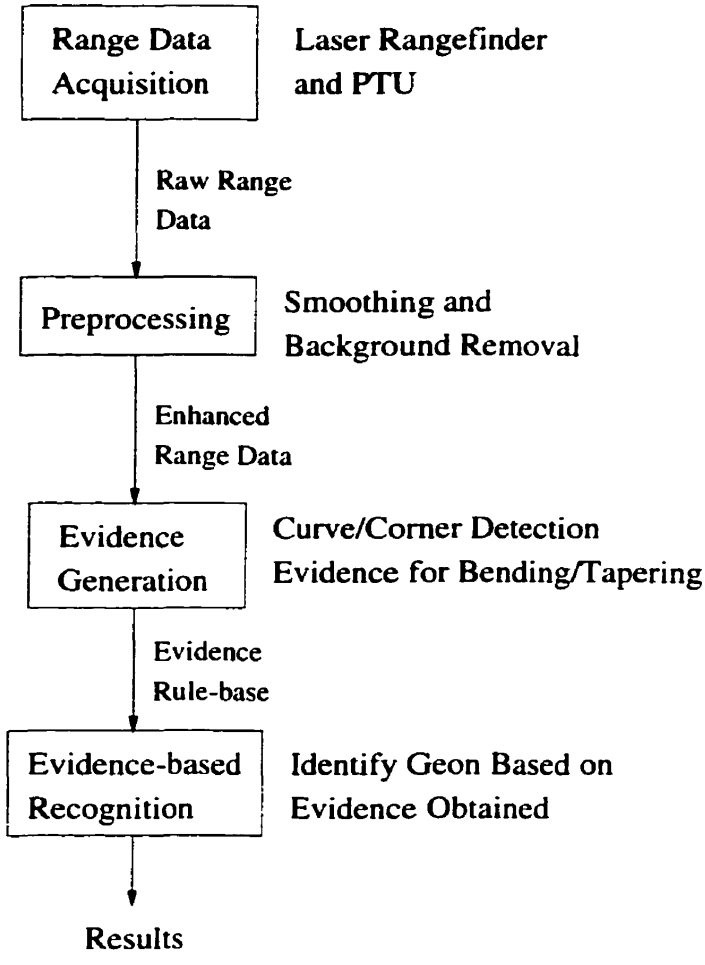


FIGURE 5.1. Stages of the recognition process

significantly different from their neighbouring points, it exhibits the undesirable effect of spreading the error to its neighbouring points, as illustrated in Figure 5.2(a)-(b). Noise of this nature is common in range data, since laser rangefinders are often prone to the effect of specular reflections. Therefore, the Gaussian kernel is not used for smoothing range data in our experiments. Instead, the median smoothing kernel is used since it does not have the undesirable effect of spreading error to neighbouring points (see Figure 5.2(c)).

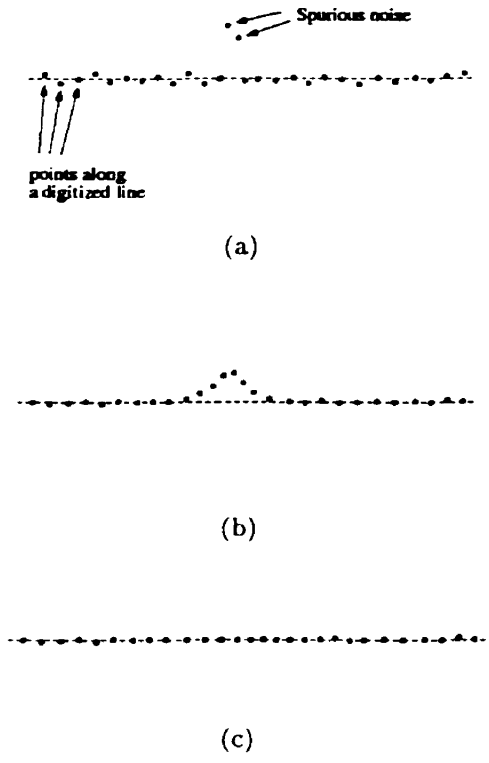


FIGURE 5.2. (a)Points along a digitized line. (b)The digitized line in (a) after Gaussian filtering. (c)The digitized line in (a) after median filtering.

The other issue is the removal of background. The rangefinder sees parts of the laser stripe reflected off the floor (or the table), as well as those reflected off the geon, making it necessary to distinguish one from the other.

Our background segmentation method is based on two sets of information. The first set of information comes from the binarized image of the geon, which is obtained by thresholding the colour histogram of the image. A Vertical Region Of Interest (VROI) can be established in the binarized image, as shown in Figure 5.3. A margin of several pixels is adopted on each side to take into account the slight discrepency between the image obtained by the colour camera and the image acquired by the B/W camera, and the effect of the morphological erosion performed on the binarized image. All points lying outside the VROI are quickly rejected as background. Note that we did not attempt to establish a Horizontal Region Of Interest (HROI), because

the colour camera is mounted above the black and white camera. Thus the colour image's HROI does not correspond to that of the black and white image.

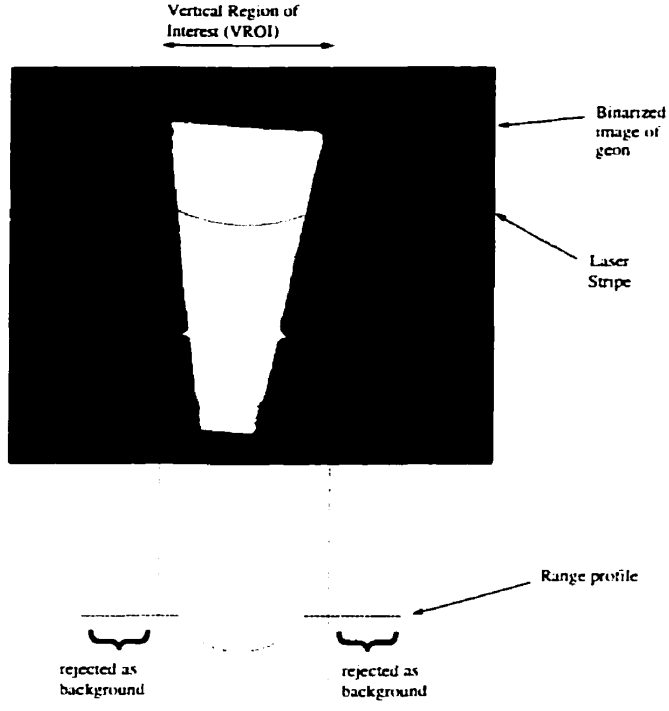


FIGURE 5.3. Background removal based on VROI

The second set of information is derived from the geometry of the robot and the vision system. Referring to Figure 5.4, the distance between the image plane and the laser projection on the object, d, can be used to infer the vertical distance between the laser projection and the image plane, h_{laser}. (d is obtained directly from the rangefinder). If the laser is projected onto the floor, then h_{laser} will be roughly equal to the height between the camera and the floor, h_{cam}. The values of h_{laser} corresponding to every point on the range profile are calculated, and if $h_{laser} \geq h_{cam}$, then the point is rejected as background. h_{laser} and h_{cam} are calculated through equation 5.1 and equation 5.2, respectively:

$$h_{laser} = d \times cos(\frac{\pi}{2} - \phi_{PTU} - \theta) \tag{5.1}$$

$$h_{cam} = h_{PTU} + h_{cam\_to\_PTU} cos(\phi_{PTU}) \tag{5.2}$$

(Refer to Figure 5.4 for definition of variables.)

The value of θ in equation 5.1 is calculated based on the location of the laser stripe on the image, and the known value of the angle subtended by the camera's field of view, θ_{fov}. The details are discussed in Subsection 4.2.2.

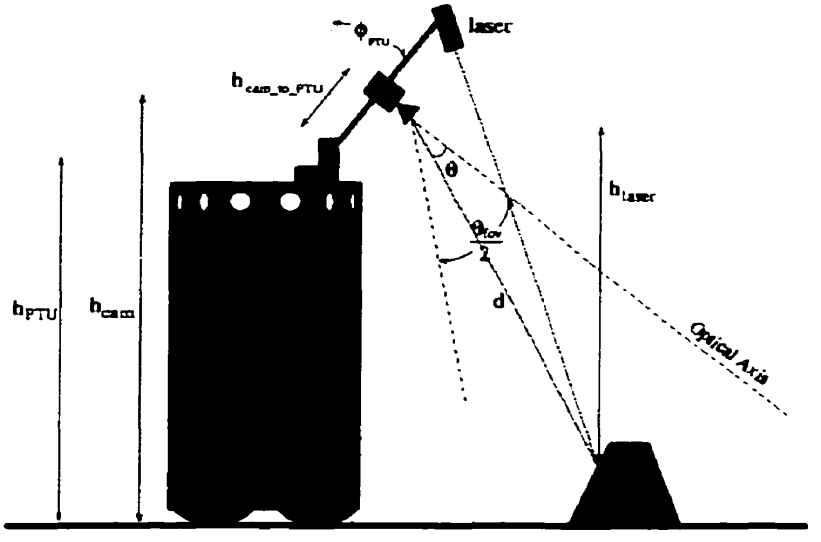


FIGURE 5.4. Background removal based on the vision system's geometry

In the experiments, a 1cm margin is given to h_{laser} to account for inaccuracies in range data and angle measurements. Thus, a point is rejected as background only if $h_{laser} \geq h_{cam} - 1$. This margin may cause some "good" points to be filtered out. We assume that this discrepancy will not seriously affect the performance of the recognition process.

Another method for background range data removal was also considered. Given a range profile, like the one shown in Figure 5.3, the first and last corners of the profile correspond to the background limits. Therefore, once those two corners are detected, the points to the left of the first corner and those to the right of the last corner can be considered as background. This method was rejected because we could not assume that the quality of laser reflected off the background will be as good as that reflected off the painted geons. In the experiments, parts of the laser stripe often hit dark patches on the floor and gaps between floor tiles, which resulted in noisy range data poorly suited for corner detection.

After these data preprocessing steps, the resulting range data are ready to be used for recognition. The following sections describe our recognition algorithm.

2. Evidence-based Recognition

As mentioned in Chapter 2, the classical approach of model-based matching is not suitable for our task due to the qualitative nature of geons. Instead, an evidence-based recognition technique was chosen.

Evidence-based recognition[**33**] identifies 3-D objects by looking for notable features of objects. Traditional object recognition techniques use quantitative information derived from the image to perform recognition by mapping all information into model representations. Such an approach involves graph-matching which has exponential time-complexity. Rather than using all such information, the use of only "remarkable" information, or evidence, which strongly cues certain objects, may drastically speed up the process. For instance, Jain and Hoffman[**33**] used three types of evidence conditions based on morphological, path, and boundary information to generate an evidence rulebase to identify 31 different objects from range data.

Since our recognition scheme is based on sparse range data of objects, it is important that we can extract enough evidence - from the little data available - to identify

the objects with a certain level of accuracy. We used three types of evidence features to identify the geons: evidence of tapering, evidence of bending, and presence of corners, curve segments, and straight line segments in the range profiles.

The next three sections will describe how the three evidence features mentioned above are determined based on the range profile.

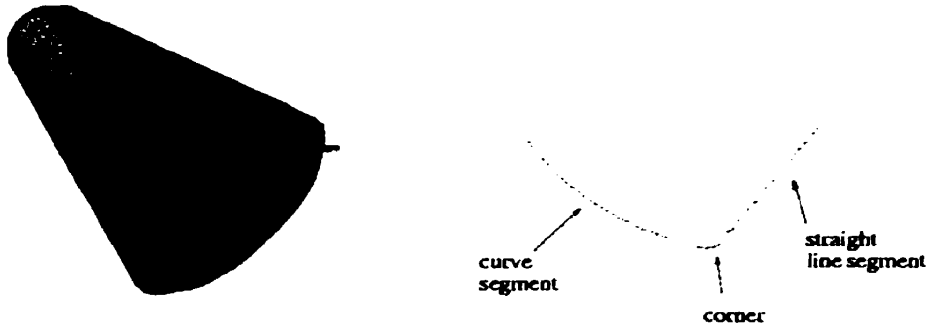
3. Curvature, Line, and Corner Detection

Considerable research effort has been placed upon curvature estimation and line fitting [**41, 49, 63**]. Since digitized curves often occur as contours of objects and regions, estimation of such curves can facilitate feature extraction for object representation and recognition [**63**]. A reliable curvature estimation algorithm is crucial to our task since the recognition of the geons is dependent on information such as the presence or absence of curves, straight lines, and corners.

A range profile of any of our geons can contain a combination of curve segments, line segments, and straight lines (see Figure 5.5). Our curvature estimation method must be able to identify those components within a digitized curve. The task is made simpler by the fact that no quantitative information (amount of curvature, angular size of corner, length, etc.) is required. On the other hand, the task is made more difficult by the fact that our digitized curves are derived from range data which are typically quite noisy, and most curvature estimation algorithms tend to be sensitive to noise.

Three different curvature estimation methods were implemented for our experiments and their performance compared: Ramer's method[**49**], Wall's method[**63**], and O'Gorman's method[**41**]. The first two methods[1], although quite fast and robust, were eventually dropped. The main reason is that these methods model curves as piecewise linear segments, as shown in Figure 5.6. Therefore, there is no evidence to show if the profile is made up of just a curve, as shown in Figure 5.6(a), or a curve and a straight line, as in Figure 5.6(b). O'Gorman's method, on the other hand,

[1]Ramer's method and Wall's method were implemented by Robert Sim, currently a graduate student in the Centre for Intelligent Machines, McGill University.



(a) (b)

FIGURE 5.5. (a)A laser scan on a tapered cylinder. (b)The range profile corresponding to the scan in (a).

explicitly differentiates and locates curve segments, line segments, and corners on a digitized curve, as shown in Figure 5.6(c).

3.1. O'Gorman's Curvilinear Feature Detection Method. The basic principle of O'Gorman's curvilinear feature detection method is to first estimate the local curvature along the digitized curve, and then analyse the plot of local curvatures to determine the presence and location of curves, lines, and corners. For this method, the Difference of Slope (DOS) on every point along the curve is calculated. Figure 5.7 illustrates how this DOS is obtained. The point of interest, $\mathbf{x}$, sits in the middle of a curve segment, $\mathbf{M}$, which separates two curve segments, both of length $\mathbf{W}$. (Lengths $\mathbf{M}$ and $\mathbf{W}$ are measured in number of pixels). Two straight lines are then fitted onto the two curve segments of length $\mathbf{W}$ using the Linear Regression method[**30**]. The angular difference between the two straight lines, θ, is then computed. The value of θ corresponding to each point along the curve is calculated, and a θ-plot is generated. In essence, the θ-plot describes the local curvature along the digitized curve: segments of the θ-plot where $\theta \approx 0$ correspond the straight line segments, while sharp, narrow peaks correspond to corners. Low, wide peaks depict curve segments.

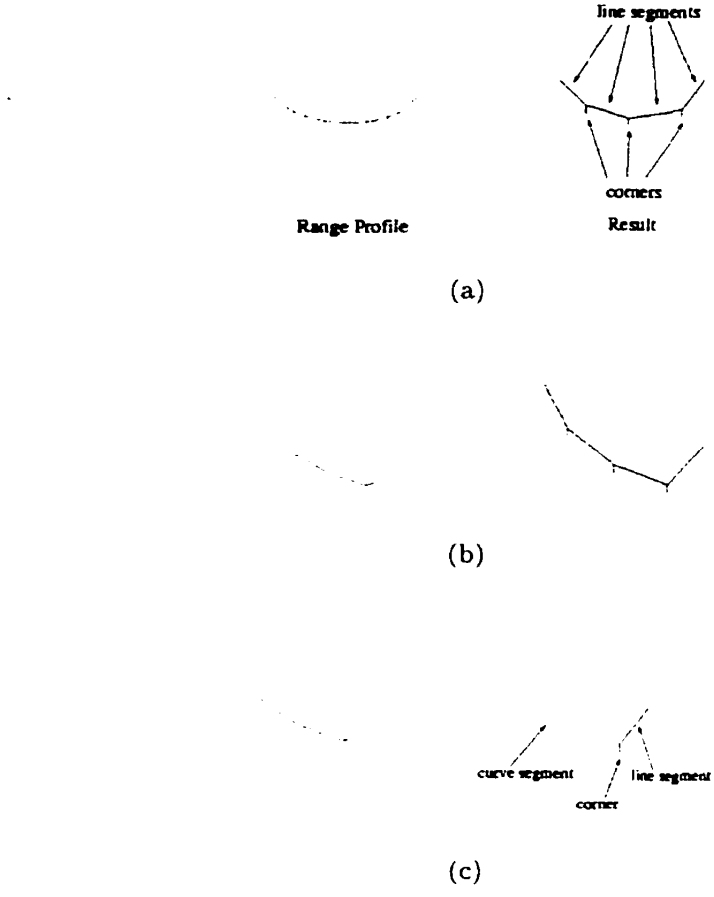


FIGURE 5.6. (a)Curve fitting using Ramer's method. (b)Curve fitting of a curve segment and a line segment using Ramer's method. (c)Curve fitting of a curve segment and a line segment using O'Gorman's method.

For this algorithm to function well, the values of parameters **W** and **M** must be determined with care. The arc length of the curve segment, **L** (where $\mathbf{L} = 2\mathbf{W} + \mathbf{M}$) determines the feature resolution of the method. The shorter **L** is, the less interference between neighbouring features. On the other hand, it is desirable to have a large value for **W**, since a long **W** has the effect of smoothing out random noise in the line-fitting process. The arc length of the gap, **M**, should be positive and equal to or greater than the maximum arc length of a corner[**40**]. The values of **W** and **M** chosen in these experiments were 10 and 3, respectively.

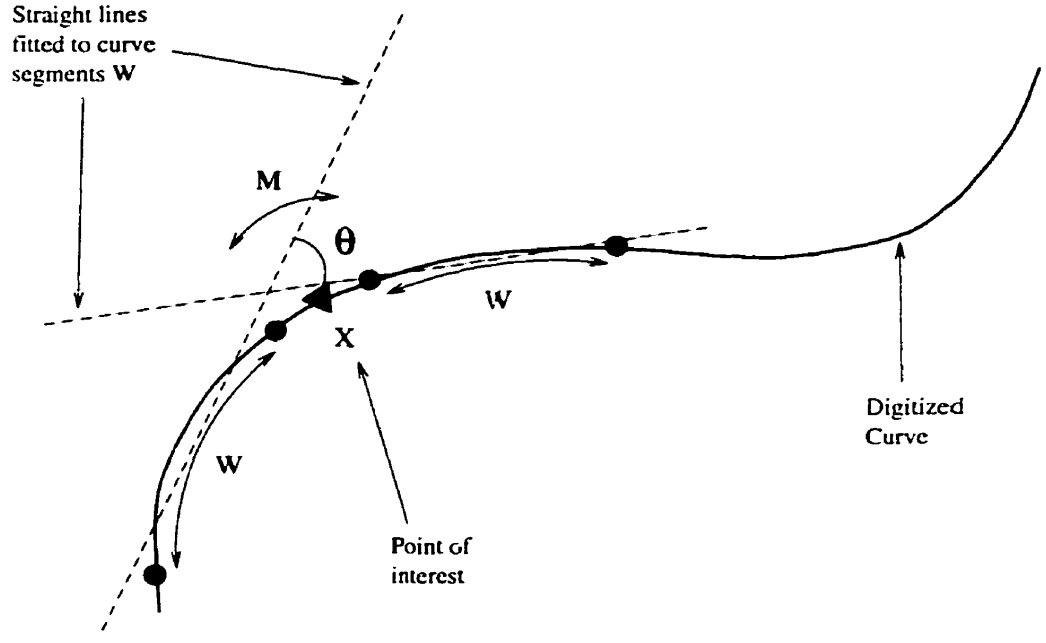


FIGURE 5.7. Illustration of the DOS method. (Adopted from [**41**])

The next step involves feature determination based on the θ-plot generated. The θ-plot of a curve segment resembles a wide, low hill, while a corner resembles a narrow, sharp peak. Theoretically, the θ-plot of a straight line should be zero at all points. However, random noise on the digitized line will result in small "jitters" on its θ-plot. Therefore, a non-zero threshold, $\theta_{S_{max}}$, must be defined such that regions on the θ-plot with values within $\pm\theta_{S_{max}}$ are considered straight line segments, and the *zero-range* is defined as:

zero-range: $-\theta_{S_{max}} \leq \theta \leq \theta_{S_{max}}$

Crossings into and out of the zero-range is designated as *zero-range crossings*, s_z:

zero-range crossing: $\theta(s_z) = \pm\theta_{S_{max}}$

A corner or a curve will result in a peak on the θ-plot with θ exceeding $\theta_{S_{max}}$. The peak consists of two zero-crossings: s_{z_j}, at the beginning of the peak, crosses out of the zero-range, and $s_{z_{j+1}}$, at the end of the peak which crosses back into the zero-range. Figure 5.8 illustrates a typical θ-plot corresponding to a corner. The width of the peak is the length between the two zero-range crossings, s_{zz}:

peak width: $s_{zz_j} = s_{z_{j+1}} - s_{z_j}$

Since corners have narrow peaks and curves have wide peaks, the width of the peak, s_{zz}, can be used to make the distinction between the two. The value of a parameter, s_{curve}, needs to be defined such that peaks of width greater than s_{curve} are considered curves. Ideally, s_{curve} is sufficient to differentiate between curves and corners. However, real range data tend to be noisy and may render this method unreliable. The peak height is therefore used in conjunction with the peak width to make the decision. Another parameter, θ_{corner}, is selected such that if the maximum height of the peak, θ_{peak} is greater than θ_{corner}, and the peak width is smaller than s_{curve}, then it is a corner. A peak with $s_{zz} > s_{curve}$ and $\theta_{peak} < \theta_{corner}$ is considered to be a curve.

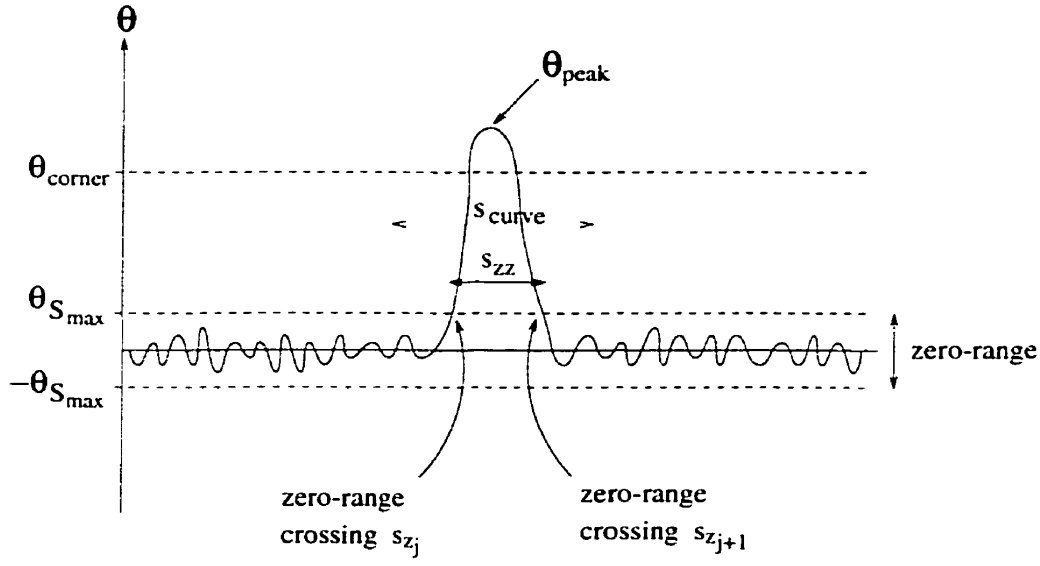
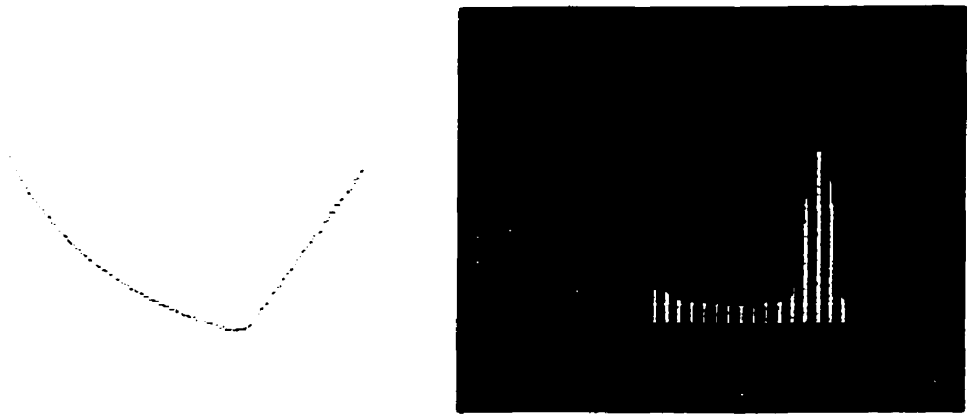


FIGURE 5.8. Typical $\theta - plot$ corresponding to a corner

Two special cases may arise: Firstly, a peak may have $s_{zz} < s_{curve}$ and $\theta_{peak} < \theta_{corner}$. Secondly, a peak may also have $s_{zz} > s_{curve}$ and $\theta_{peak} > \theta_{corner}$. The first case is considered an anomaly due to noise and therefore regarded as a straight line segment. The latter is considered as a corner immediately next to a curve, as shown in Figure 5.9.



(a) (b)

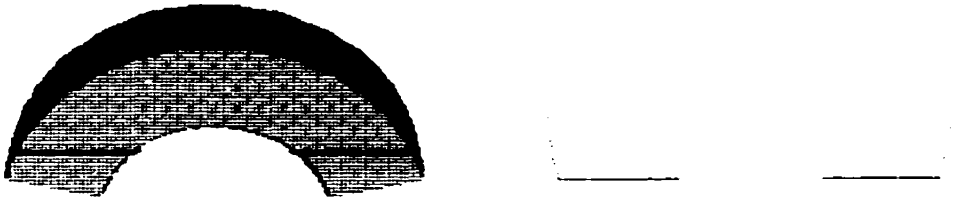
FIGURE 5.9. (a)The range profile of a corner immediately next to a curve. (b)The θ-plot corresponding to the profile in (a).

In summary, O'Gorman's curvature detection algorithm is an appropriate one for the purpose of our experiments because of its ability to explicitly distinguish between curve segments, corners, and line segments. It is also relatively insensitive to moderate levels of noise in range data.

4. Evidence of Bending

The seven geons considered in this research can be classified into three categories: bent objects (bent cylinders and bent cuboids), tapered objects (tapered cylinders and tapered cuboids), and the rest which are neither bent nor tapered (cuboids, cylinders, and ellipsoids). If any evidence of bending can be found in the data, then the search space can be narrowed down from the seven possible geons to the two bent geons.

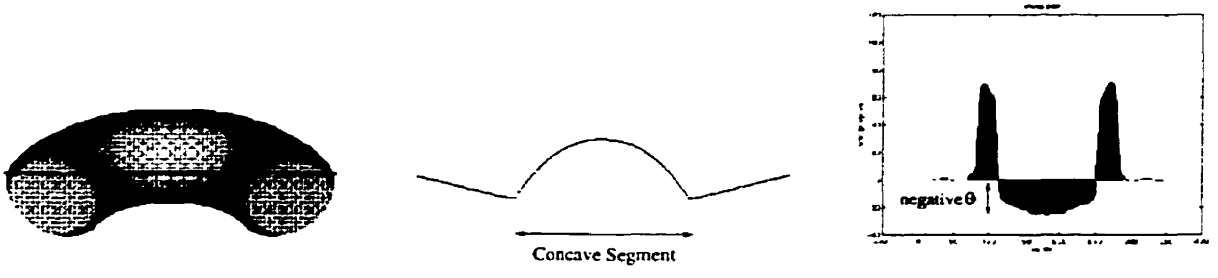
One indication of bending is the presence of a range profile consisting of two separate segments, as illustrated in Figure 5.10. Since this is a characteristic unique to bent objects, it constitutes an evidence of bending. In other words, if any one or more of the six range profiles on the geon consists of two separate segments, then the object is either a bent cuboid or a bent cylinder.



(a) (b)

FIGURE 5.10. (a)A bent cuboid (b)A range profile on the bent cuboid in (a) which consists of 2 segments.

Another sign of bending is the presence of concavities in the range profile. Out of the seven geons, only the bent ones have concave surfaces. The projection of laser stripes on those concave surfaces will result in concave segments in the range profile, as demonstrated in Figure 5.11. To detect concave segments on the range profile, we simply made use of the θ-plot generated by O'Gorman's curvature detection mechanism (refer to Section 2 of this chapter). Any concave segments on the range profile will result in negative θ values on the θ-plot, as shown in Figure 5.11(c).



(a) (b) (c)

FIGURE 5.11. (a)A bent cylinder (b)A range profile on the bent cuboid in (a) which has a concave segment. (c) θ-plot of the profile in (b).

In the event that none of the range profiles show evidence of multi-segments or concavities, it is still possible to identify bent objects by observing the coordinates

of the scans' endpoints. Figure 5.12 shows a bent cuboid and its six laser scans, none of which consists of two segments or is projected on a concave surface. The use of the range profile endpoints is similar to the use of edge information, since all endpoints invariably lie on one of the geon's edges. If complete edge information on the geon were available, evidence of bending could be easily deduced from the presence of the geon's concave edge indicated in Figure 5.12(a). However, in the absence of explicit edge information, the endpoints of each range profile must be used to infer the presence of such concave edges.

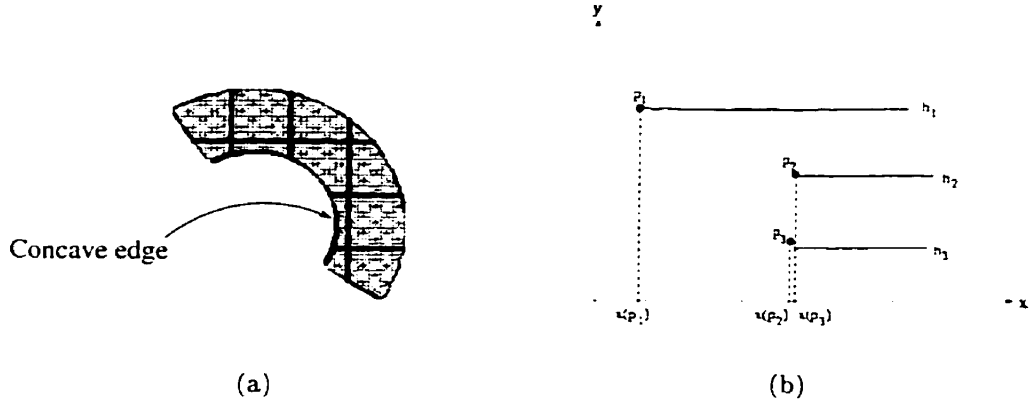


FIGURE 5.12. (a)A bent cuboid (b)The range profiles on the bent cuboid in (a) with the left endpoints illustrated.

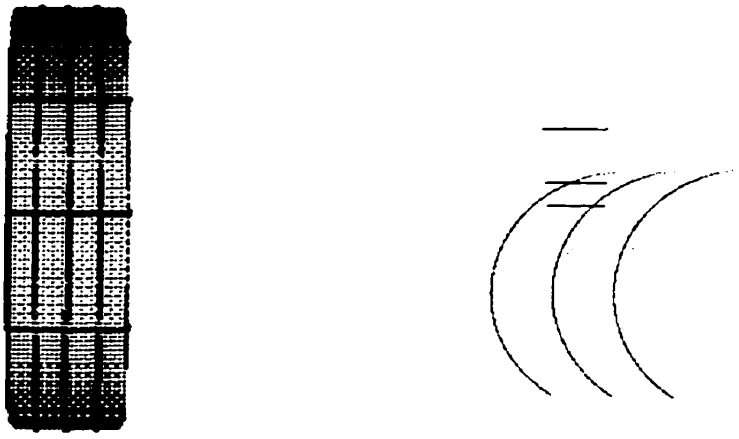
Figure 5.12(b) shows the horizontal profiles of the bent cuboid, with each of the left endpoints denoted p_1, p_2, and p_3, and their x-coordinates denoted $x(p_1)$, $x(p_2)$, and $x(p_3)$. One can determine by observation that the endpoints p_1, p_2, and p_3 lie on a concave edge. If a straight line is drawn between p_1 and p_3, and if the point p_2 is to the right of the straight line, then the points p_1, p_2, and p_3 suggest the presence of a concave edge. For convex and straight edges, p_2 would fall to the left of the line and on the line, respectively. To state this mathematically, a concave edge is present if the following equation is satisfied:

$$x(p_2) - \frac{x(p_1) + x(p_3)}{2} > k_{tol} \tag{5.3}$$

where k_{tol} is a predetermined value of tolerance to account for inaccuracies.

The same reasoning is used on the right endpoints of the horizontal profiles, as well as the top and bottom endpoints of the vertical profiles to detect the presence of concave edges.

There remain some "accidental views" from which the six range profiles of the geons will not provide any evidence of bending. Figure 5.13 shows one such case. These accidental views account for most of the erroneous results.



(a) (b)

FIGURE 5.13. (a)A bent cuboid (b)The range profiles on the bent cuboid in (a), showing no evidence of bending

5. Evidence of Tapering

Unlike the bent geons, tapered geons do not produce range profiles that contain obvious clues such as multiple-segments or concavities. As a result, evidence of tapering must be inferred from the endpoints of the range profiles.

A tapered geon's cross-sections are either consistently expanding or contracting. Therefore, range profiles with consistently increasing or decreasing lengths are strong evidence of tapering. This is illustrated in Figure 5.14. The horizontal lengths of scans h_1,h_2, and h_3 are increasing consistently, which implies the object's expanding cross-sections. The same logic applies to the vertical scans. In other words, the geon is considered tapered if any one of the following conditions is satisfied:

$$L_{h1} > L_{h2} > L_{h3} \tag{5.4}$$

$$L_{h1} < L_{h2} < L_{h3} \tag{5.5}$$

$$L_{v1} > L_{v2} > L_{v3} \tag{5.6}$$

$$L_{v1} < L_{v2} < L_{v3} \tag{5.7}$$

Of course, there are many viewpoints from which the range profiles of a tapered geon satisfy none of the above conditions. Figure 5.15(a) and (b) shows one such view. To take care of these cases, our recognition system relies on another clue based on the range profile endpoints. Figure 5.15(a)-(d) show a tapered cuboid, a cuboid, and their corresponding horizontal range profiles. Horizontal distances between the endpoints are used to determine whether or not the geon is tapered. The definitions of lengths L_{1-2}, L_{2-3}, L'_{1-2}, L'_{2-3} are indicated in Figure 5.15(d). Observing the profiles of the cuboid, one can see that the difference between L_{1-2} and L_{2-3} is approximately the same as the difference between L'_{1-2} and L'_{2-3}. This is true for all cuboids, cylinders and ellipsoids in any stable position, but not true for the tapered geons. As can be seen in Figure 5.15(b), the profiles of the tapered cuboid lack the kind of symmetry

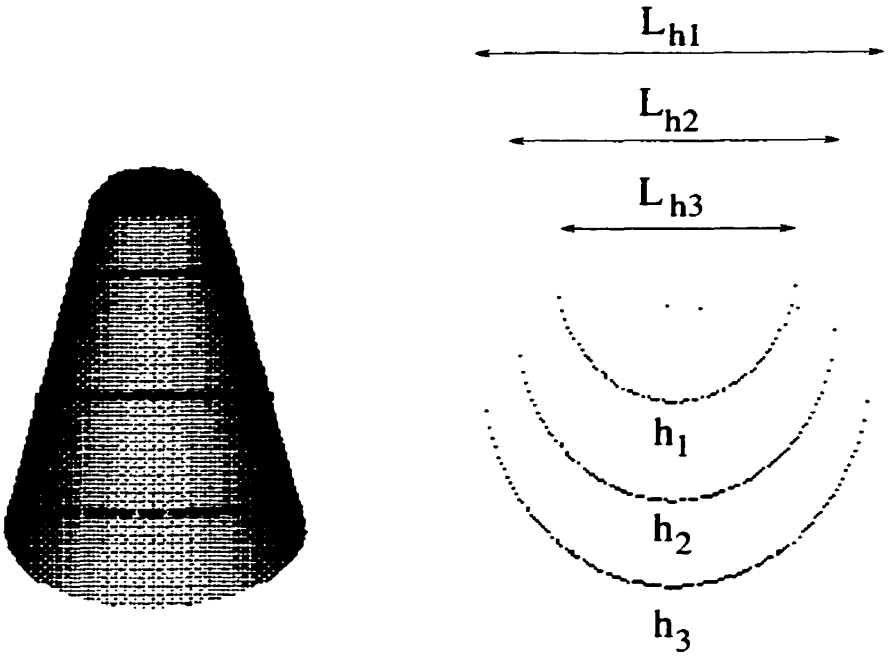


FIGURE 5.14. (a)A tapered cylinder (b)The horizontal range profiles on the tapered cylinder in (a), with consistently increasing lengths.

exhibited by the cuboid's profiles. Therefore, the tapered geons satisfy the following condition:

$$||L_{1-2} - L_{2-3}| - |L'_{1-2} - L'_{2-3}|| > k_{tol} \tag{5.8}$$

where k_{tol} is a predetermined tolerance to account for variations and noise.

Note that this condition alone does not guarantee that the geon is tapered, since bent geons also satisfy this condition. The object must conform to this equation AND show no evidence of bending to be considered tapered.

Again, there exist accidental views which will lead to erroneous recognition, as illustrated in Figure 5.16. For example, the range profiles of the tapered cuboid, shown in Figure 5.16(b), do not show any evidence of tapering mentioned in this section.

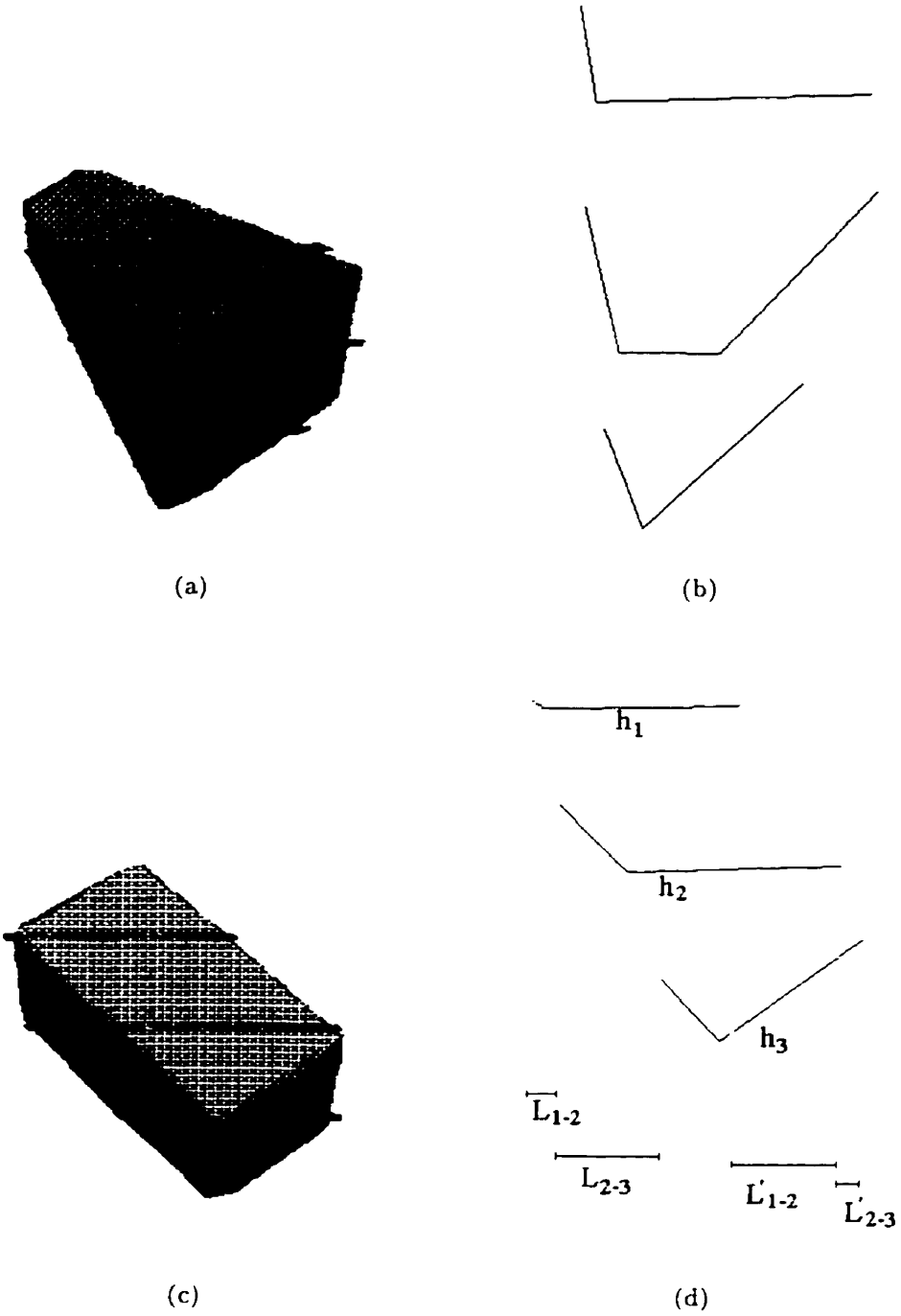
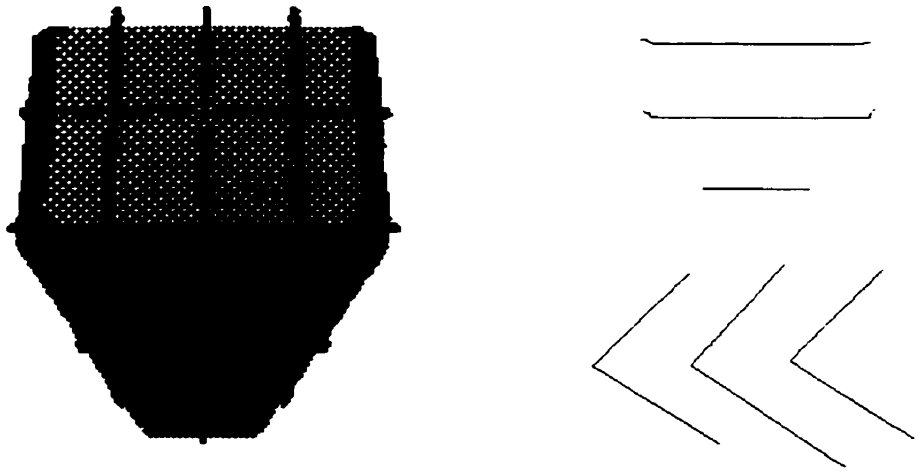


FIGURE 5.15. (a)A tapered cuboid. (b)The range profiles on the tapered cuboid in (a).(c)A cuboid. (d)The range profiles on the cuboid in (c).



(a) (b)

FIGURE 5.16. (a)A tapered cuboid (b)The range profiles on the tapered cuboid in (a), showing no evidence of tapering

In summary, this section presents two methods of detecting evidence of tapering. The first one determines whether the geon's cross-sections are expanding or contracting by checking whether the length of the range profiles is consistently increasing or decreasing. The second one checks whether the profiles exhibit signs of symmetry, a property shared by cuboids, cylinders, and ellipsoids. Both methods use information derived only from the profile endpoints, since tapered objects lack unique surface characteristics (unlike the bent objects, which have concave surfaces). This makes the recognition of tapered geons somewhat less reliable, as indicated by the experimental results presented in the next chapter. Note that in both methods the geon should be centered in the field of view so that all endpoints are within the view of the rangefinder.

6. The Decision Tree

A vast number of expert systems utilize decision trees to determine what actions to execute or to classify objects. The main advantage of using decision trees is that they are simple and thus easy to understand intuitively, and their simplicity also leads to higher speed. One class of decision trees, known as binary trees, can only have two branches for every node. Each node essentially contains a YES/NO question, depending on the answer, one of the two branches is traversed to reach the next node. A binary tree can therefore be easily constructed using a hierarchical structure of "IF...THEN" rules. This section describes the binary tree we have implemented for the purpose of identifying a geon based on its six range profiles. The questions in each node of the tree are answered by the evidence obtained using methods described in previous sections of this chapter. Our decision tree is shown in Figure 5.17.

The initial step is to look for evidence of bending. This is done first because, as mentioned in the last section, evidence of tapering cannot be established unless it is known that the geon does not show any sign of bending. If evidence of bending is present, then the remaining possibilities are bent cylinder and bent cuboid. The decision is based upon information on the curvature characteristics of the range profiles. This is difficult because both geons have curved surfaces, and thus will have range profiles with curved segments. However, the proportion of curved surfaces on a bent cylinder is far larger than that on a bent cuboid, thus one can deduce that in general, bent cylinders will result in more curve segments in the range profiles than the bent cuboids. The decision tree considers the geon a bent cylinder if more than three of the range profiles contain curve segments, otherwise it is considered a bent cuboid. The number three was chosen since it is found to minimize the number of erroneous results based on trial and error.

If no evidence of bending is found, we proceed to check for evidence of tapering. If the geon is found to be tapered, then the next step is rather straightforward: since only tapered cylinders have curved surfaces, the geon can be considered a tapered cylinder as long as more than one of the range profiles contain curve segments. A

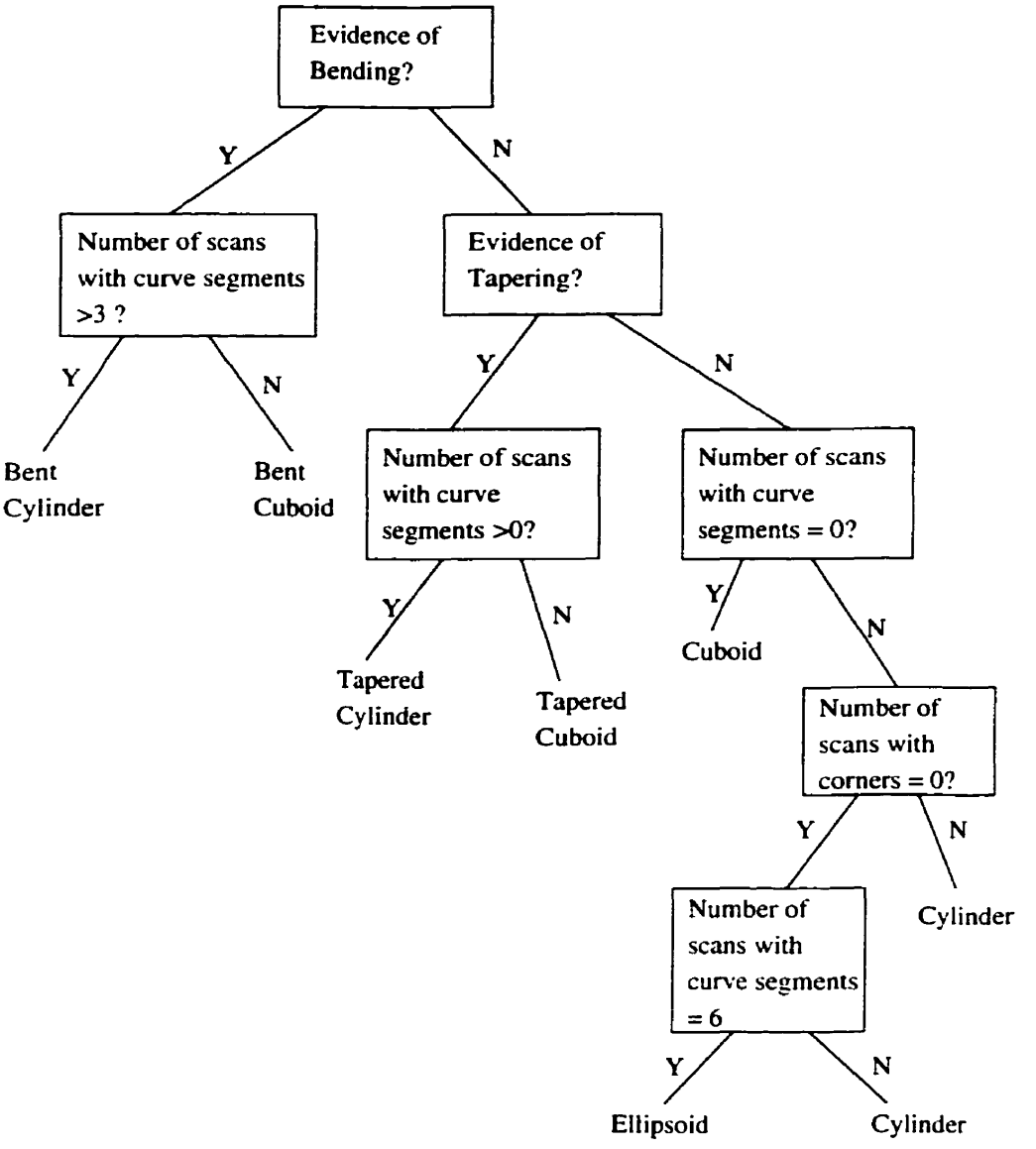
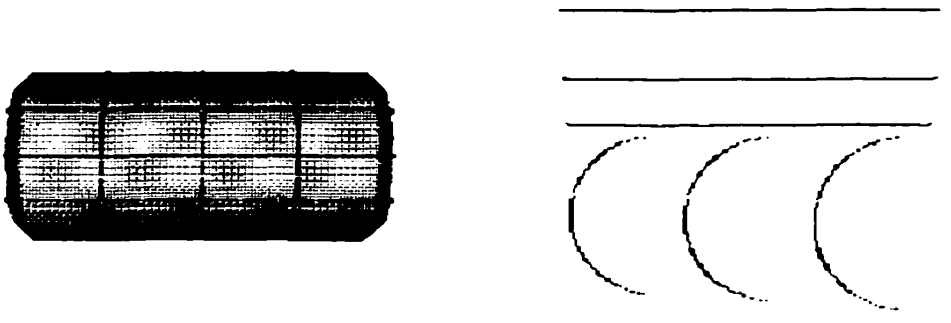


FIGURE 5.17. The decision tree

tapered cuboid, on the other hand, only has flat surfaces and therefore its profiles should not have any curve segments.

Without any evidence of tapering and bending, the geon can either be a cuboid, a cylinder, or an ellipsoid. The distinction between these three geons must be based solely on the curvature information in range profiles. The cuboid can be easily distinguished from the other two since it is the only one with no curved surfaces. Thus its

range profiles should not have any curve segments. If curve segments are present, then we check for the presence of corners. Since an ellipsoid is composed entirely of one continuous curved surface, its range profiles should not have any corners. Therefore the geon must be a cylinder if corners are present. In the absence of corners, the geon is most likely an ellipsoid, but in some rare cases it can also be a cylinder. Figure 5.18 shows a view from which a cylinder's range profiles have no corners. Thus, we check the number of curved range profiles. If all the six range profiles are curved, the geon is certainly an ellipsoid, otherwise it is considered a cylinder.



(a) (b)

FIGURE 5.18. (a)A cylinder (b)The range profiles of the cylinder in (a), showing no presence of corners.

The use of decision trees for recognition has a few drawbacks. Firstly, for many complex real world problems decision trees can become too large to be understandable or manageable. We did not have to deal with this problem due to the small number of geon classes considered. Secondly, decision trees tend to be inflexible - if we were to add another class of geons to the seven we already have, the whole tree would have to be restructured, which may be difficult. Thirdly, a tree demands discrete, unambiguous answers (unless a "fuzzy tree" is used). In particular, a binary tree can only accept YES or NO answers, and therefore cannot take into account any

uncertainty or quantitative inputs. Nevertheless, the desire for speed and simplicity prompted us to adopt this approach for our decision-making process.

In this chapter we have described the preprocessing stage which "cleans up" the range data prior to recognition. This involves background removal and median filtering to alleviate noise. The methods of corner/curve detection and inferring evidence of tapering and bending were also presented. Finally, the last section explained the structure of the decision tree for the recognition process.

CHAPTER 6

Experiments

Three sets of experiments were performed to test different aspects of our active vision and geon recognition system:

(i) The testing of our geon recognition algorithm using computer-generated superquadrics.

(ii) The testing of our geon recognition system's performance on real range data.

(iii) The testing of the overall system, which includes the robot navigation, active vision, and the geon recognition using the visual system mounted on a mobile robot.

The approach to each set of experiments and their results are discussed in the first three sections of this chapter.

1. Recognition on Simulated Data

Our geon recognition algorithm was tested using computer simulation. 3-D models of geons were generated using the software **makesq**[1], which requires the user to enter 18 parameters for a superquadric shape. The software takes the 18 parameters and generates the 3-D points for the superquadric's surfaces. The surface points corresponding to the locations of the six laser stripes were extracted as scan data for geon

[1] Written by Kenong Wu[**65, 66**]

recognition. For each geon, four variations were produced in order to test the effectiveness of our recognition algorithm in identifying geons with different quantitative attributes, such as different degree of tapering and bending.

The viewing sphere approach [**19, 55, 56, 57**] was adopted to test our recognition algorithm. In this approach, an imaginary sphere centered around the geon is tessellated into small viewing cells, and recognition is performed from each viewing cell on the sphere. Since our research assumes that the vision system is mounted on a mobile robot, and that the geons to be recognized always lie below the vision system's height, we do not need to consider the entire surface of the viewing sphere. Instead, only the shaded region of the sphere, shown in Figure 6.1(a), will be considered. As indicated in Figure 6.1(b), the region spans the range of elevation angle from 30° to 70°, and the entire azimuth of 360°. The viewing region is tessellated into viewing cells, each spanning 10° of elevation angle and 10° of azimuth angle, making a total of $4 \times 36 = 144$ viewing cells.

Each of the 28 objects (7 geon types with 4 variations of each) can also have more than one stable position. Recognition was performed by applying the viewing sphere to every possible stable position of each object. The results are presented and discussed in the following subsections.

1.1. Tapered Geons. Each of the two tapered objects (tapered cylinder and tapered cuboid) have four variations in shape (see Figure 6.2), ranging from very tapered (object 1) to slightly tapered (object 4). The tapered objects have three stable positions, as illustrated in Figure 6.3. Recognition was performed through all the 144 viewing cells for every stable position for each object. Tables 6.1 and 6.2 show the recognition rate of the tapered cuboids and cylinders, respectively. The columns of numbers in boldface indicate the percentage of correct recognition, while other columns of numbers indicate the percentage of incorrect recognition.

Table 6.1 shows that the recognition rate for the tapered cuboids ranges from 75% to 98% for the first three objects only, which are more prominently tapered. For object 4, whose degree of tapering is very slight, the rate of recognition takes a

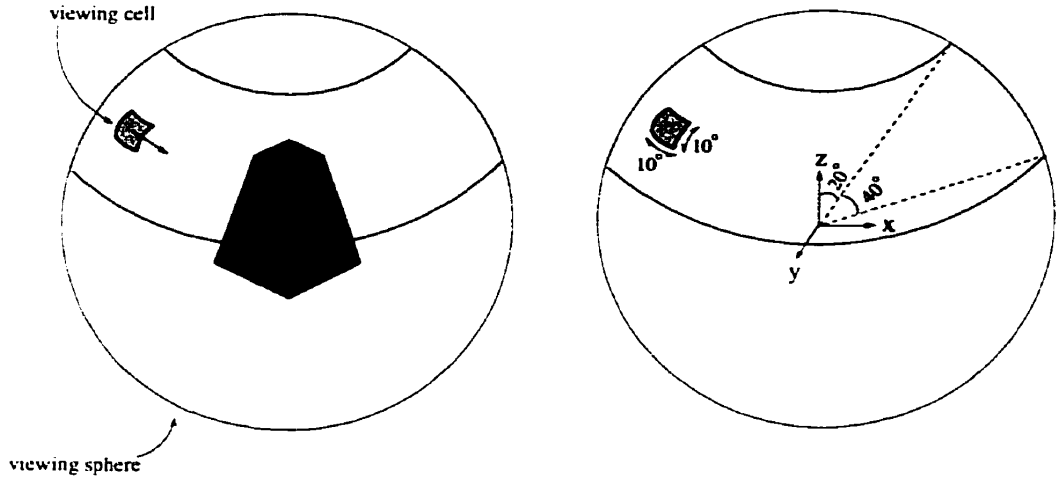


FIGURE 6.1. The viewing sphere.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	**94**	0	0	0	6	0	0
	2	**98**	1	0	0	1	0	0
	3	**92**	6	0	0	3	0	0
Object 2	1	**83**	0	0	0	17	0	0
	2	**93**	4	0	0	3	0	0
	3	**81**	14	0	0	5	0	0
Object 3	1	**75**	11	0	0	8	5	0
	2	**84**	4	0	0	11	1	0
	3	**92**	5	0	0	3	0	0
Object 4	1	**62**	10	0	0	26	2	0
	2	**62**	10	0	0	16	12	0
	3	**75**	0	0	0	25	0	0

TABLE 6.1. Rate of recognition for tapered cuboids (%)

significant drop. Due to the subtlety of its tapering, the recognition system begins to mistake the tapered cuboid for a simple cuboid. This raises an interesting question: When should an object be considered tapered? That is, how subtle can the tapering be for the object to still be considered a tapered object? This is a rather open-ended question which is subject to opinion. Psychophysical experiments similar to Brown et al.'s "pop-up" experiments[**14**] can be performed to observe how human subjects

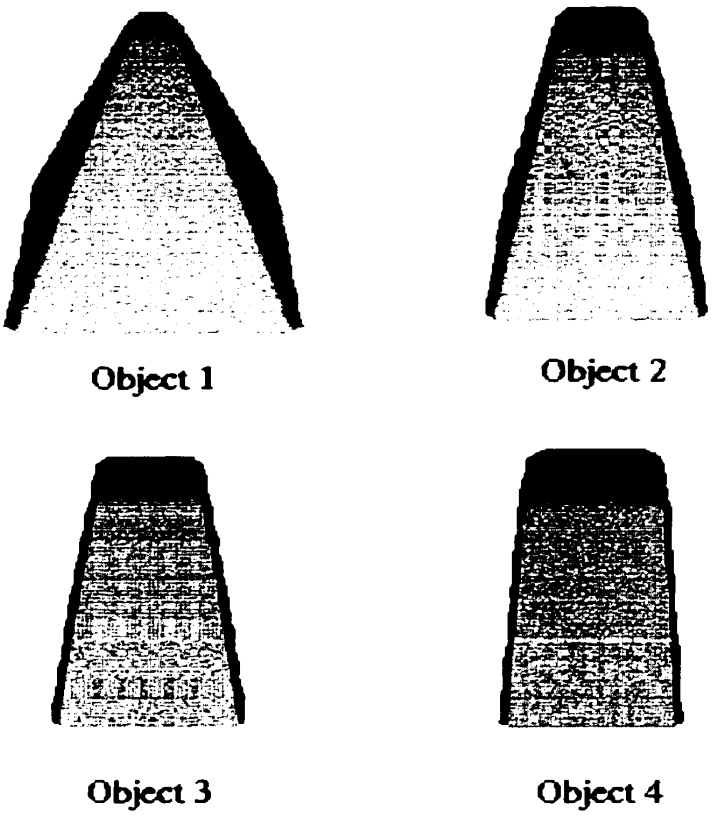


FIGURE 6.2. The four different tapered cuboids tested.

respond to this question. The human observers can be shown objects with different degrees of tapering, and asked to quickly determine whether or not the object is tapered.

The results in Table 6.1 also indicate that the tapered cuboid is often mistaken as a tapered cylinder. This suggests occasional failure of our corner/curve detection algorithm.

The results for the tapered cylinders (Table 6.2) are considerably better. This is mostly due to the fact that while our curvature detection algorithm quite often mistakes straight lines and corners as curves, the reverse is not true. Curve segments with a reasonable degree of curvature are always correctly identified as curves. This bias stems from the nature of O'Gorman's curvature detection method[**41**]. Under

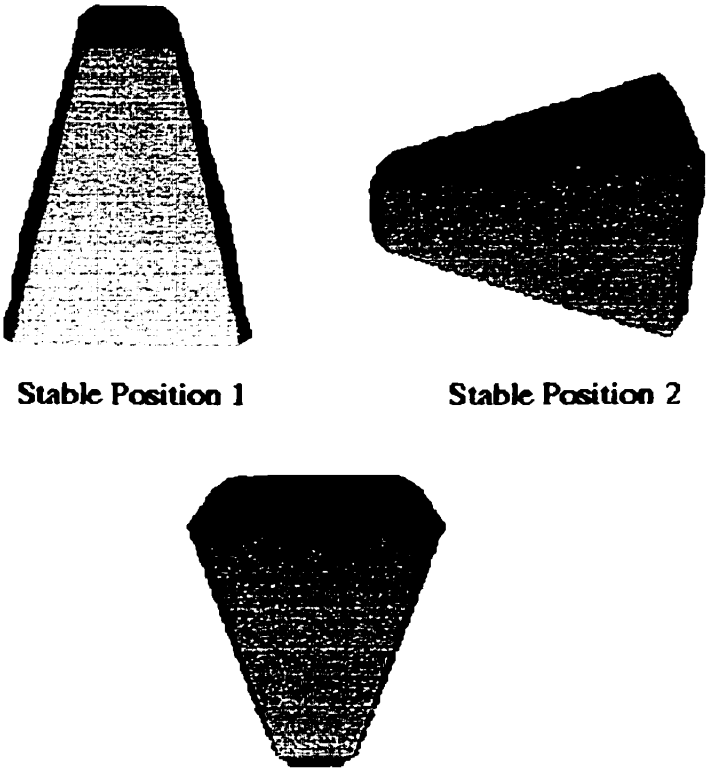


FIGURE 6.3. The stable positions of tapered cuboids.

the effect of random noise, the θ-plot of a straight line will have non-zero θ values, and thus will be easily mistaken as the plot of a curved segment. The θ-plots for curved segments, on the other hand, already have non-zero values for θ, and random noise will merely add some perturbation to the already non-zero θ's, leaving the final decision unchanged.

1.2. Bent Cuboid. Similar to the tapered geons, four variations of each bent geon were tested, ranging from very bent (object 1) to slightly bent (object 4), as shown in Figure 6.4. Three stable positions exist for the bent cuboids (see Figure 6.5). Note that stable position 3 only applies to the bent cuboids, but not the bent cylinders.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	**100**	0	0	0	0	0
	2	1	**88**	0	0	2	23	6
	3	0	**100**	0	0	0	0	0
Object 2	1	0	**100**	0	0	0	0	0
	2	0	**78**	0	0	0	4	18
	3	0	**100**	0	0	0	0	0
Object 3	1	0	**100**	0	0	0	0	0
	2	0	**90**	0	0	0	6	4
	3	0	**100**	0	0	0	0	0
Object 4	1	0	**100**	0	0	0	0	0
	2	0	**90**	0	0	0	10	0
	3	0	**100**	0	0	0	0	0

TABLE 6.2. Rate of recognition for tapered cylinders (%)

Tables 6.3 and 6.4 show the results for the bent cuboids and the bent cylinders, respectively. As expected, the recognition rate declines as bending becomes less prominent. The bent cuboids are also quite frequently mistaken for bent cylinders, and vice versa. This is a rather difficult problem to tackle, since both geons can produce range profiles with curve segments, line segments, and corners. As mentioned in Chapter 5, the decision tree distinguishes the two geons by the number of range profiles with curve segments. Since a bent cylinder has more curved surfaces than a bent cuboid, the decision tree identifies the geon as a bent cylinder if more than three of its six scans contain curve segments. This is not a reliable rule per se: rather, it is chosen for its statistical likelihood to produce correct results.

1.3. Cuboids, Cylinders, and Ellipsoids. Figures 6.6, 6.8, and 6.10 show the different variations of cuboids, cylinders, and ellipsoids, respectively. The variations are different degrees of elongation of the geons. Results are tabulated in Tables 6.5, 6.6, and 6.7. Those for the cuboids are relatively poor, while those of the cylinders are nearly perfect. This is due to the same reason discussed in Section 5.1.1: our curvature detection algorithm has a tendency to mistake straight line segments and corners as curves because of noise, but not vice versa.

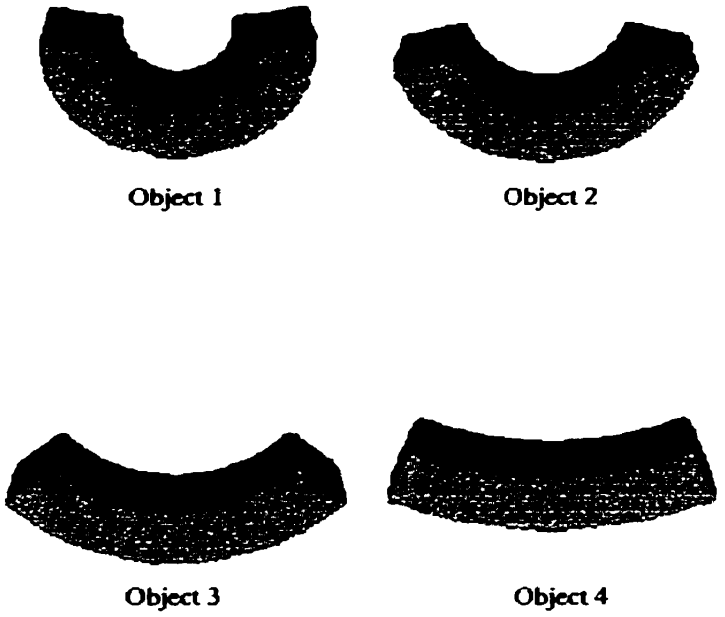


FIGURE 6.4. The four different bent cuboids tested.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	0	**98**	2	0	0	0
	2	0	16	**76**	2	0	6	0
	3	0	1	**87**	12	0	0	0
Object 2	1	0	0	**83**	17	0	0	0
	2	0	15	**75**	4	0	6	0
	3	0	0	**86**	14	0	0	0
Object 3	1	0	0	**85**	15	0	0	0
	2	0	21	**73**	0	0	6	0
	3	0	3	**86**	11	0	0	0
Object 4	1	0	0	**83**	17	0	0	0
	2	15	18	**61**	0	0	6	0
	3	4	10	**83**	3	0	0	0

TABLE 6.3. Rate of recognition for bent cuboids (%)

The results for the first two ellipsoids are perfect, but for ellipsoids that are severely elongated, the results become very poor. In most cases they are mistaken as cylinders. This should not be considered as a serious flaw of the recognition algorithm.

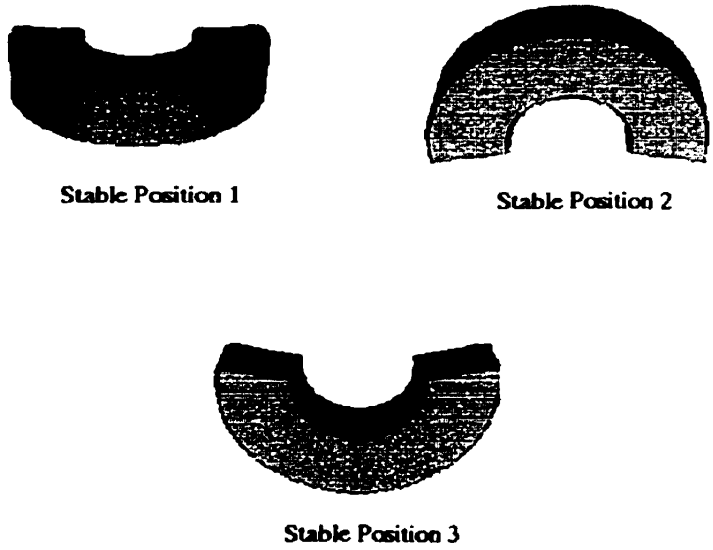


FIGURE 6.5. The stable positions of bent cuboids.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	0	27	**73**	0	0	0
	2	0	0	0	**83**	0	0	17
Object 2	1	0	0	40	**60**	0	0	0
	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Object 3	1	0	0	43	**57**	0	0	0
	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Object 4	1	0	0	52	**48**	0	0	0
	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A

TABLE 6.4. Rate of recognition for bent cylinders (%)

since a very elongated ellipsoid does indeed resemble a cylinder, even to the human observer.

1.4. Summary. In this part of the experiment we have constructed simulated 3-D surface points for a total of 28 objects (four variations for each of the seven geons), and all possible stable positions of each object were tested using the viewing sphere approach. The purpose of the four variations is to test the behaviour of our system when the geons vary in their degree of tapering, bending, and elongation. Results

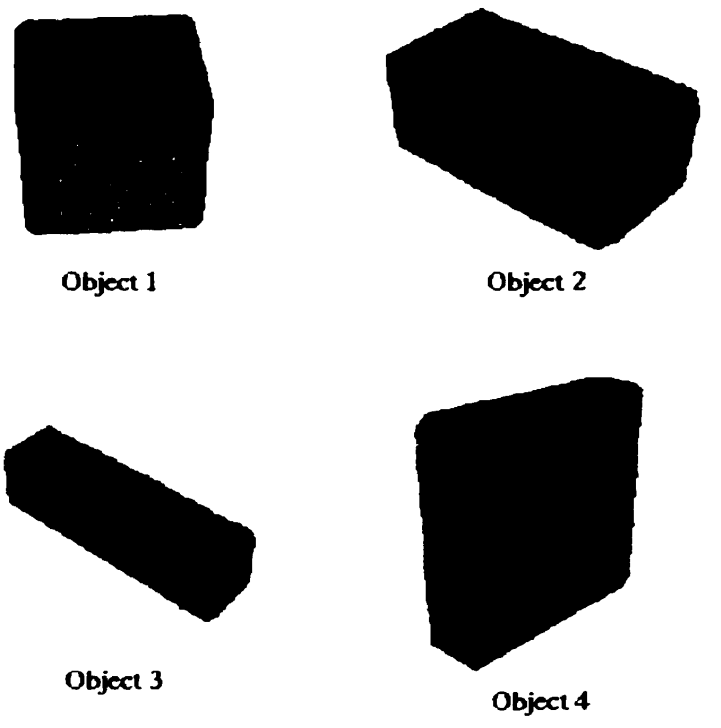


FIGURE 6.6. The four different cuboids tested.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	0	0	0	**83**	17	0
	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Object 2	1	0	0	0	0	**89**	11	0
	2	0	0	0	0	**67**	33	0
Object 3	1	0	0	0	0	**88**	12	0
	2	0	0	0	0	**86**	14	0
Object 4	1	0	0	0	0	**94**	6	0
	2	0	0	0	0	**94**	6	0

TABLE 6.5. Rate of recognition for cuboids (%)

show that our recognition system's performance degenerates gracefully under extreme conditions. For instance, the system begins to mistake a very slightly bent cuboid as a cuboid, or a very slightly tapered cylinder as a cylinder; such behaviour is expected even for the human observer.

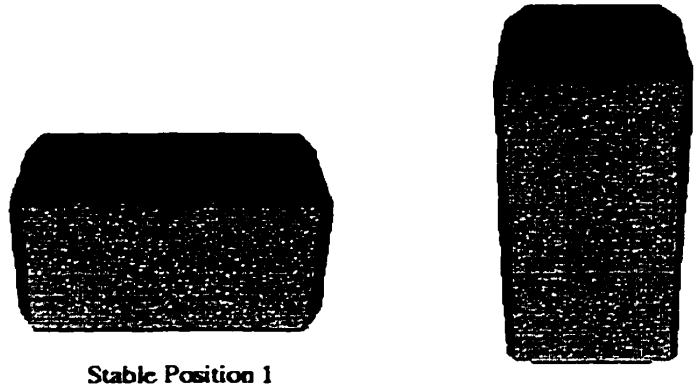


FIGURE 6.7. The stable positions of cuboids.

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	0	0	0	0	**100**	0
	2	0	0	0	0	0	**100**	0
Object 2	1	0	0	0	0	0	**97**	3
	2	0	0	0	0	0	**100**	0
Object 3	1	0	0	0	0	0	**100**	0
	2	0	0	0	0	0	**100**	0
Object 4	1	0	0	0	0	0	**100**	0
	2	0	0	0	0	0	**100**	0

TABLE 6.6. Rate of recognition for cylinders (%)

Object	Stable Position	Tapered Cuboid	Tapered Cylinder	Bent Cuboid	Bent Cylinder	Cuboid	Cylinder	Ellipsoid
Object 1	1	0	0	0	0	0	0	**100**
Object 2	1	0	0	0	0	0	0	**100**
Object 3	1	0	6	0	0	0	15	**78**
Object 4	1	0	53	0	3	0	25	**19**

TABLE 6.7. Rate of recognition for ellipsoids (%)

Other erroneous results can be attributed to the occasional failure of our corner/curve detection mechanism, as well as certain "accidental views" from which the

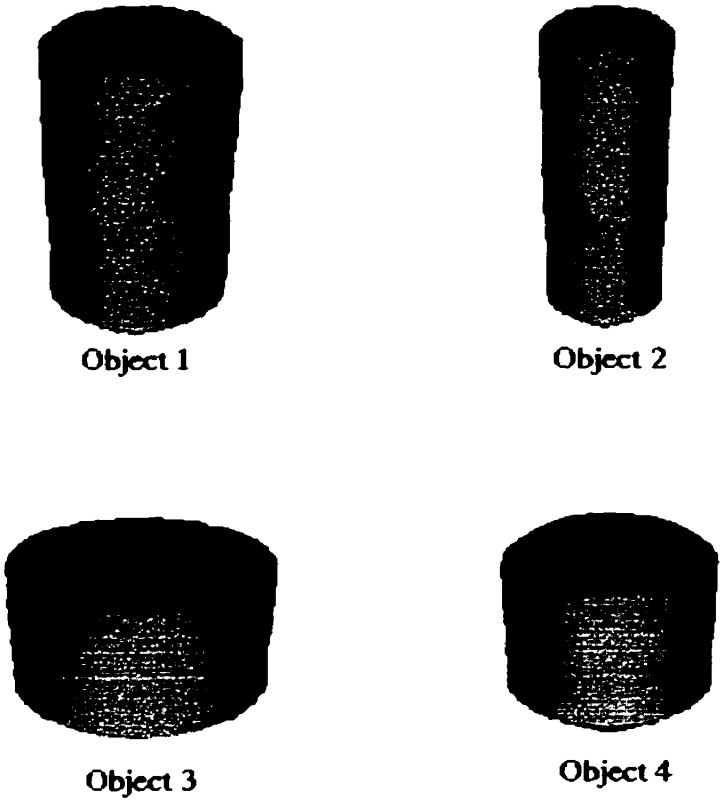


FIGURE 6.8. The four different cylinders tested.

range profiles fail to convey crucial information such as evidence of bending and tapering. Figure 6.11 shows a graph that indicates the relationship between the rate of incorrect recognition and the location of the viewing cell. The percentage error ranges from as low as 3% in some viewing cells to as high as 30% in others. This clearly indicates that our system performance is worse for certain viewing cells, where accidental views are likely to occur.

2. Recognition of Real Geons

In this part of the experiment, real geons were used to test our computer vision system. The vision system includes the laser rangefinder, the colour camera, and the Pan Tilt Unit (PTU). The seven geons were constructed of wood, with the exception of the bent cylinder, which was made of clay.

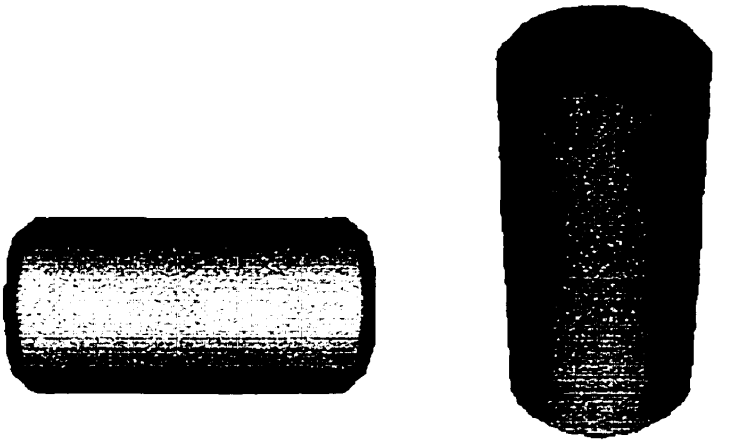


FIGURE 6.9. The stable positions of cylinders.

All the geons were painted red, and the active vision module of our vision system performed colour segmentation on the image by regarding all red pixels as foreground and anything else as background (refer to Chapter 4 for the details of colour segmentation). Red was chosen because it appears to produce the strongest reflection from the laser.

The vision system was placed on a bench, 80cm above the floor. Video signals were fed directly to a Silicon Graphics Indy workstation. Since the system itself cannot achieve any translational movement, the geon must be placed within the operating range of the laser rangefinder, which is 70cm to 200cm.

The first step of the process was to search for red objects in the surroundings, panning and tilting the PTU as required. Once the geon was within the colour camera's field of view, geometric reasoning was used to deduce the location of the geon and the pan/tilt angles required for the proper placement of laser stripes. The six range profiles were then acquired and recognition was performed. We tested our system with no less than ten unique poses for each geon. As an example, Figure 6.12 shows the 11 poses we used for the tapered cuboid.

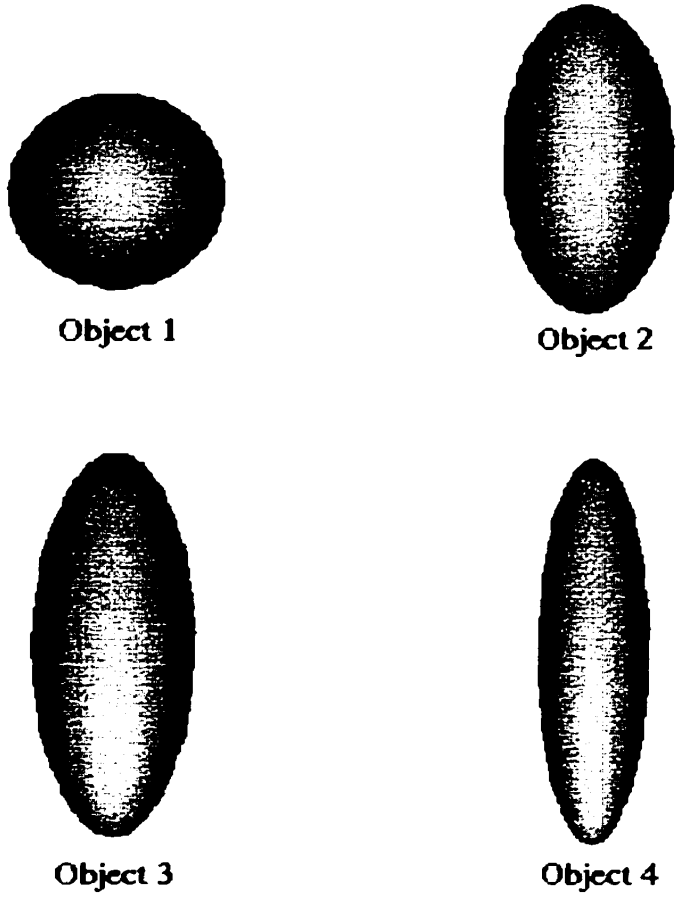


FIGURE 6.10. The four different ellipsoids tested.

We also investigated the effect of increasing distance on the performance of our recognition system. Therefore, each geon was tested at three different ranges: 100cm, 130cm. and 160cm from the camera. The overall results are shown in Tables 6.8, 6.9, and 6.10.

The results on real geons are somewhat poorer than those in the previous section. This is expected since the recognition of real geons involves several problematic issues which are absent in the recognition on simulated data. Firstly, real range data contains more noise than simulated data. Secondly, our simulation assumes perfectly accurate placement of laser stripes on the geons, whereas laser stripe placement on

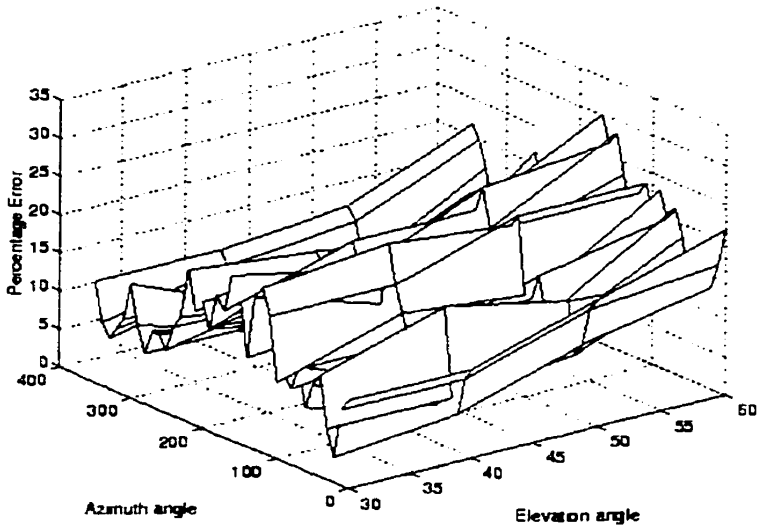


FIGURE 6.11. Relationship between error rate and location of viewing cell.

Object	Bent Cuboid	Bent Cylinder	Tapered Cuboid	Tapered Cylinder	Cuboid	Cylinder	Ellipsoid
Bent Cuboid	**83**	17	0	0	0	0	0
Bent Cylinder	17	**75**	0	0	0	0	8
Tapered Cuboid	0	0	**64**	9	27	0	0
Tapered Cylinder	0	0	0	**82**	0	18	0
Cuboid	0	0	10	0	**90**	0	0
Cylinder	0	10	0	10	0	**80**	0
Ellipsoid	0	0	0	0	0	0	**100**

TABLE 6.8. Rate of recognition at range 100cm (%)

real geons is inferred from geometric reasoning, which can be error-prone (see Chapter 4). Lastly, with real geons, the issue of background segmentation has to be accounted for, and as mentioned in Chapter 5, background filtering sometimes leads to "good" data being discarded.

The above results also indicate a decline in performance over increasing range. A major reason that leads to this decline is that the level of noise rises as range increases, and a higher level of noise results in poorer performance in our curvature detection algorithm. The other reason for poorer rate of recognition at long range stems from

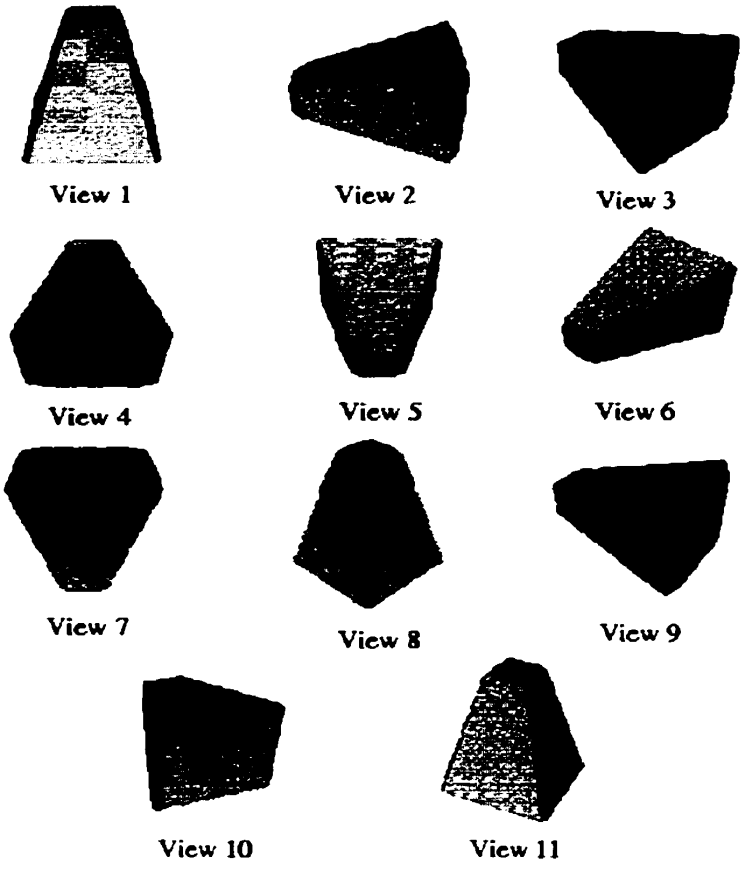


FIGURE 6.12. The 11 unique views of the tapered cuboid used to test the recognition system.

the background removal method. As mentioned before, background removal was partly based on geometric calculations. The accuracy of those calculations declines with increasing range, thus leading to more good data being filtered out. In some cases range data from the laser, reflected off the floor, are mistaken as surface points on the geons.

Object	Bent Cuboid	Bent Cylinder	Tapered Cuboid	Tapered Cylinder	Cuboid	Cylinder	Ellipsoid
Bent Cuboid	**75**	25	0	0	0	0	0
Bent Cylinder	17	**67**	0	8	0	0	8
Tapered Cuboid	0	0	**73**	18	9	0	0
Tapered Cylinder	9	0	0	**82**	0	9	0
Cuboid	0	0	10	0	**80**	10	0
Cylinder	0	0	0	20	0	**80**	0
Ellipsoid	0	0	0	0	0	0	**100**

TABLE 6.9. Rate of recognition at range 130cm (%)

Object	Bent Cuboid	Bent Cylinder	Tapered Cuboid	Tapered Cylinder	Cuboid	Cylinder	Ellipsoid
Bent Cuboid	**58**	42	0	0	0	0	0
Bent Cylinder	25	**67**	0	8	0	0	0
Tapered Cuboid	0	0	**55**	36	9	0	0
Tapered Cylinder	9	9	0	**73**	0	9	0
Cuboid	10	0	10	0	**50**	30	0
Cylinder	0	30	0	10	0	**60**	0
Ellipsoid	0	10	0	0	0	10	**80**

TABLE 6.10. Rate of recognition at range 160cm (%)

3. Geon Recognition Combined with Autonomous Robot Navigation

The last part of the experiments is to integrate our vision system with an autonomous robot, and make the robot navigate to search for geons in an indoor environment. The robot navigation algorithm described in Section 4.3 was employed. The robot's movements were restricted to within a "pen" inside the laboratory, so as to minimize the danger of collision with furniture and equipment. The pen was roughly 3.0m x 4.5m in size, and was constructed with plastic boards 65cm high. The geons were either placed on the floor or on a desk of known height. Instead of using a real desk, a 70cm tall box was used as a substitute. This is because the sonar sensors on the robot may fail to pick up the legs of a desk, and thus may result in a collision.

The robot was able to navigate within the pen swiftly without colliding with obstacles, in most cases succeeding to detect and approach the geons within 30 seconds. In several of the trials, however, the robot got trapped at a local minimum and failed to converge to its destination. Since this problem only concerns the robot navigation algorithm, it will not be accounted for in this thesis.

The results for 30 runs in which the robot successfully moved close to the geon were recorded. Out of those 30 runs, only 60% of them gave correct recognition. Half of the runs were done with the geon on the floor, and the other half with the geon on the "desk" (box). The rate of recognition for the latter case was 73%, which was better than those of the former, 47%. The reason for this difference was that with the geon on the box, the range between the geon and the rangefinder was relatively short, approximately 100cm. With the geon on the floor, the range was much longer, approximately 150cm. This property of diminishing accuracy over range led to the poorer performance with the geon on the floor.

Another reason for the mediocre overall performance has to do with the robot's surroundings. The floor in the laboratory was composed of raised floor tiles. As the robot travels across the room, its wheels would go into the grooves between the tiles, causing the robot to tilt by a small angle. As mentioned in Chapter 5, background removal for the range data was accomplished through geometric reasoning. That small tilt angle in the robot was enough to upset the geometric calculations, resulting in occasional failure in background rejection.

In summary, this chapter has presented experimental data to evaluate the performance of our recognition algorithm under three scenarios: recognition based on simulated range data, recognition of real geons, and combining the vision system with mobile robot navigation to detect and identify real geons. We have also discussed issues which affect the outcome in each scenario. For instance, recognition of real geons involve disparaging factors that are absent in recognition based on simulated data, such as diminishing accuracy over increasing range. Some measures could be taken to improve the results, and these will be addressed in the next chapter.

CHAPTER 7

Conclusions

In this research the feasibility of identifying a small set of simple volumetric shapes based on a few strategically placed laser stripes was investigated. The use of qualitative 3-D shapes, called geons, was inspired by Biederman's Recognition By Components (RBC) theory [**8, 9**]. An evidence-based mechanism was used in the recognition process, and a mobile robotic visual system was implemented in order to apply our research techniques to real data. This chapter gives a brief discussion of our work and experimental results, as well as suggestions for improvements.

1. Thesis Summary

Our task of geon recognition with mobile robot can be loosely divided into two parts. The first part, referred to as the active vision module, involves navigation of the robot in the environment, visual search for objects of interest, and obtaining relevant range information. These actions call for the need of a mobile robot, a colour camera, a laser rangefinder, and a pan-tilt unit for camera movements. The details of the implementation and integration of this hardware were given in Chapter 3. The goal of the active vision module is to obtain quality range data for recognition by all available means. This requires a robot navigation algorithm to bring the robot close to the geon, a colour segmentation algorithm to establish focus of attention, and a method for proper laser stripe placement. These issues were addressed in Chapter 4.

The other part of the task is the recognition module. This module is responsible for processing the range data acquired by the active vision module, and deducing the geons' identities based on qualitative attributes of the range profiles.

Chapter 6 presented the experimental results. Our recognition algorithm was first tested with simulated range data. Results were less than perfect, mostly due to certain accidental views from which the range data of a geon failed to convey some of it characteristics. Not surprisingly, results on real range data were poorer than those on simulated data, since real data brought additional complications such as noise, inaccurate laser stripe placement, and poor background rejection. The results also illustrated an important point: under extreme conditions, such as a tapered cylinder with very slight tapering, our system had the tendency to overlook the tapering and mistake the geon as a cylinder. Also, a very elongated ellipsoid was often mistaken as a cylinder. This kind of error is expected even for a human observer. This suggests that a qualitative approach to object recognition such as ours may be appropriate for mimicking human vision.

2. Discussion

This research investigates an approach for identifying simple volumetric shapes using sparsely spaced laser stripes. Very little work in object recognition has been done using this technique. Qiang et al[**47**] proposed a method to recognize polyhedral objects using as few as one laser-stripe scan. The range data is matched against polyhedral object models in the database using quantitative constraints such as absolute lengths of scanned segments and heights of the segments' endpoints. This calls for the need to store in the database precise geometric measurements of every object model, which leads to substantial storage requirements and a lengthy matching process. Our algorithm, on the other hand, makes decisions based on dichotomous or trichotomous attributes. The main reason is to conform to the qualitative nature of geons as proposed by Biederman.

Traditional range-based object recognition utilizes a complete range map of the scene, and often requires lengthy procedures such as computing surface normals to every point on the image. Our recognition algorithm, on the other hand, uses only a small number of one-dimensional range profiles and thus requires minimal processing. However, the overall recognition process remains quite slow, even though the data processing alone takes merely a fraction of a second. By far the biggest bottleneck in the overall process is the Pan-Tilt Unit movement. PTU movements are required to place laser stripes on designated parts of the geons, but this step can be avoided if a different rangefinder design is adopted. For instance, Sato's range-finder design [**54**] uses a technique that combines temporal laser switching and a small galvano mirror to achieve light pattern generation, thus eliminating the need for mechanical moving parts.

Colour was chosen as the cue for focus of attention, mainly because the acquisition and processing of colour information is relatively inexpensive and straightforward. Speed is an important issue in our colour segmentation algorithm, since almost a third of a million pixels need to be processed in each colour image. We used a simple thresholding method to separate background and foreground. The thresholding is done in the RGB colour space so as to avoid the need for colour-space transformations. This led to fast but somewhat unreliable colour segmentation. The process is sensitive to changes in lighting conditions. In the experiments, colour segmentation tends to fail when the geon is placed under direct sunlight. Some of the sophisticated colour clustering techniques [**53**] are likely to give much better performance, but they are too slow for real-time applications. Perhaps future advances in computer technology can make those techniques fast enough to be feasible.

The removal of background range data was a difficult issue. Focal attention based on colour imaging could only localize the geons in the surroundings. Additional measures had to be taken in order to reject the portions of a laser stripe reflected off the floor. Some research showed that simple shapes can be extracted from a range image by fitting hyperquadric models to them [**29**]. This approach was not feasible

due to the sparse range data we used. Eventually we chose to use geometric reasoning to solve this problem. Experiments showed that this method occasionally failed, and in the scenario where the mobile robot was involved, performance was severely compromised when the robot traversed slightly uneven floors, thereby rendering the geometric calculations inaccurate. One solution that is likely to succeed in rejecting background range data is to use a polychromatic laser range sensor, a technology developed recently by the National Research Council of Canada [4]. This device can provide both range and colour intensity measurements. With this sensor, the desired range data can be extracted simply by choosing those with the same colour as the geons.

Further research can be done to improve the generality of the system. For instance, a larger number of geons can be considered, and the restriction of the geons being in their stable positions can be relaxed. To do that, more laser scans may be required to gather the necessary discriminating information. It may even be feasible to extend the system to recognize more complex objects composed of several geons. This will most likely require object segmentation based on intensity/colour images, combined with range-based recognition of the segmented geons.

3. Conclusions

This thesis has presented a vision system that combines colour imagery and range data to search for and identify geons. Experimental results show that the recognition is reasonably accurate in both simulated range data and real data. We believe that performance can be further enhanced by implementing some of the suggested changes mentioned in the previous section, and that recognition of simple objects based on very sparse range data is feasible.

REFERENCES

[1] John Aloimonos and Amit Bandyopadhyay. Active vision. In *The IEEE 1st int. Conf. on Computer Vision*, pages 35–54, Univ. of Pennsylvania,Pennsylvania,PA,USA, June 1987.

[2] Ruzena Bajcsy. Active perception. In *Proceedings of the IEEE, v.76 n.8*, pages 996–1005, Univ. of Pennsylvania,Pennsylvania,PA,USA, August 1988.

[3] Tucker Balch, Gary Boone, Tom Collins, Harold Forbes, Doug MacKenzie, and Juan Carlos Santamaria. Io, ganymede and callisto - a multiagent robot trash-collecting team. In *AI Magazine. v 16 n 2*, pages 39–51, Georgia Institute of Technology, Atlanta, Georgia, USA, 1995.

[4] R. Baribeau, M. Rioux, and G Godin. Color reflectance modeling using a polychromatic laser range sensor. In *Pattern Analysis and Machine Intelligence*, volume 14, pages 263–269, 1991.

[5] H. G. Barrow and J. M. Tenenbaum. Computational vision. In *Proceedings of the IEEE*, volume 69, pages 572–595, 1988.

[6] R. Bergevin and M.D. Levine. Recognition of 3-d objects in 2-d line drawings: An approach based on geons. In *Technical Report TR-CIM-88*, McGill Univ., Montreal, PQ, Canada, Nov 1988.

[7] R. Bergevin and M.D. Levine. Generic object recognition: Building coarse 3d descriptions from line drawings. In *IEEE Computer Society Workshop on Interpretation of 3D Scenes*, McGill Univ., Montreal, PQ, Canada, Nov 1989.

[8] Irving Biederman. Recognition-by-components: A theory of human image understanding. In *Psychological Review, v.94, n.2*, pages 115–147, SUNY at Buffalo, NY, USA, 1987.

[9] Irving Biederman, John E. Hummel, Peter C. Gerhardstein, and Eric E. Cooper. From image edges to geons to viewpoint invariant object models: A neural net implementation. In *Proceedings of Spie - the International Society for Optical Engineering. v 1708*, pages 570–578, Univ. of Southern California, Los Angeles, CA, USA, 1992.

[10] Francois Blais, Marc Rioux, and Jacques Domey. Optical range image acquisition for the navigation of a mobile robot. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 3, pages 2574–2580, 1991.

[11] T. E. Boult and A. D. Gross. Recovery of superquadrics from 3d information. In *Proceedings - Spatial Reasoning and Multi-Sensor Fusion Workshop*, pages 128–137, 1987.

[12] R. D. Boyle and R. C. Thomas. *Computer Vision - a First Course.* Blackwell Scientific Publications, 1988.

[13] J P Brady, Nandhakumar. N., and J K Aggarwal. Recent progress in the recognition of objects from range data. In *Proceedings - International Conference on Pattern Recognition 9th.*, pages 85–92, 1988.

[14] James M. Brown, Naomi Weisstein, and James G. May. Visual search for simple volumetric shapes. In *Perception and Psychophysics, v.51*, pages 40–48, Univ. of Georgia, Athens, GA, USA, 1992.

[15] Byung-Hwa Chang, Seong-Dae Kim, and Jae-Kyoon Kim. Threshold selection algorithm for extracting a uniform colour region in an image. *Electrionic Letters*, 23:1362–1363, 1987.

[16] Roland T. Chin and Charles R. Dyer. Model-based recognition in robot vision. In *Computing Surveys*, volume 18, pages 67–108, 1986.

[17] Sven J. Dickinson, Alex P. Pentland, and Azriel Rosenfeld. A representation for qualitative 3-d object recognition integrating object-centered and viewer-centered models. In *Science of Vision*. Springer-Verlag, 1990.

[18] Li Du and Roger Munck-Fairwood. Shape using volumetric primitives. In *Image and Vision Computing*, volume 11, pages 364–371, 1993.

[19] David W Eggert, Kevin W Bowyer, Charles R Dyer, Henrik I Christensen, and Dmitry B Goldgof. Scale space aspect graph. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, pages 1114–1130, 1993.

[20] El-Hakim, Sabry F. Beraldin, and J. Angelo. Integration of range and intensity data to improve vision-based three-dimensional measurements. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 2350, pages 306–321, 1994.

[21] Francois Ennesser and Gerard Medioni. Finding waldo, or focus of attention using local color information. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 17, pages 805–809, 1995.

[22] H. R. Everett. *Sensors for Mobile Robots*. A K Peters, Ltd., 1995.

[23] D.A. Forsyth. A novel approach to colour constancy. In *Second Int Conf on Comput Vision*, pages 9–18, Oxford University, Oxford, UK, 1988.

[24] Koji Fujiwara, Richard Alan Peters II, and Kazuhiko Kawamura. Colored-object detection for a mobile robot. In *Proceedings of Spie - the International Society for Optical Engineering. v 2298*, pages 457–469, Minolta Camera Co., Ltd., Nashville, TN, USA, 1994.

[25] Brian Funt and Graham Finlayson. Color-constant color indexing. In *IEEE Transactions on Pattern Analysis and Machine Intelligence. v 17 n 5*, pages 522–529, Simon Fraser University, Vancouver, BC, Canada, May 1995.

[26] Brian Funt and Jian Ho. Color from black and white. In *Second Int Conf on Comput Vision*, pages 2–8, Simon Fraser University, Vancouver, BC, Canada, 1988.

[27] W. E. L. Grimson, T. Lozano-Perez, N. Noble, and S. J. White. An automatic tube inspection system that finds cylinders in range data. In *IEEE Computer Vision and Pattern Recognition*, pages 446–452, 1993.

[28] Peter Gvozdjak and Ze-Nian Li. From nomad to explorer: Active object recognition on mobile robots. In *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics. v.5*, pages 4079–4084, Simon Fraser University, Vancouver, BC, Canada, 1995.

[29] Song Han, Dmitry B. Goldgof, and Kevin W. Bowyer. Using hyperquadrics for shape recovery from range data. In *IEEE 4th International Conference on Computer Vision*, pages 492–496, 1993.

[30] Jeri R. Hanly, Elliot B. Koffman, and Joan C. Horvath. *C Program Design for Engineers*. Addison-Wesley Publishing Company, 1995.

[31] A. J. Hanson. Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds. In *Computer Vision, Graphics, and Image Processing*, volume 44, pages 191–210, 1988.

[32] Glenn Healey and David Slater. Using illumination invariant color histogram descriptors for recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, pages 355–360, Univ of California, Irvine, CA, USA, 1994.

[33] Anil K. Jain and Richard Hoffman. Evidence-based recognition of 3-d objects. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 10, pages 783–802, 1988.

[34] J. J. Koenderink and A. J. van Doorn. The singularities of the visual mapping. In *Biological Cybernetics*, volume 24, pages 51–59, 1976.

[35] E. Krotkov. *Exploratory visual sensing for determining spatial layout with an agile stereo camera system.* PhD thesis, University of Pennsylvania, 1987.

[36] Wei-Chung Lin and Tsu-Wang Chen. Csg-based object recognition using range images. In *Proceedings - International Conference on Pattern Recognition 9th.,p 99-103*, 1988.

[37] Janusz Marszalec and Elzbieta Marszalec. *Integration of Lasers and Fiber Optics into Robotic Systems.* SPIE Optical Engineering Press, 1994.

[38] Fumio Murakami. Accuracy assessment of a laser triangulation sensor. In *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, volume 2, pages 802–805, 1994.

[39] Q. L. Nguyen. Geon-based object representation using range imaging. Master's thesis, McGill University, 1991.

[40] Lawrence O'Gorman. An analysis of feature detectability from curvature estimation. In *Computer Vision and Pattern Recognition*, 1988.

[41] Lawrence O'Gorman. Curvilinear feature detection from curvature estimation. In *Proceedings - 9th International Conference on Pattern Recognition*, pages 1116–1119, 1988.

[42] Yu-Ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Color information for region segmentation. In *Computer Graphics and Image Processing*, volume 13, pages 222–241, 1980.

[43] Jun Ohya, Daniel DeMenthon, and Larry S. Davis. Recognizing objects in range images and finding their position in space. In *Proceedings of the 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 252–257, 1990.

[44] N. E. Pears. An intelligent active range sensor for vehicle guidance: System overview. In *Proceedings - International Conference on Intelligent Robots and Systems*, volume 1, pages 81–88, 1996.

[45] A. Pentland. Recognition by parts. In *Proceedings - 1st International Conference on Computer Vision*, pages 242–249, 1987.

[46] A.P. Pentland. Perceptual organization and the representation of natural form. In *Artificial Intelligence, v.28*, pages 293–331, May 1986.

[47] Shijia Qiang, Roger Mohr, and Karl. Tombre. Recognizing and locating polyhedral objects from sparse range data. In *Proceedings - International Conference on Pattern Recognition 9th.*, pages 104–106, 1988.

[48] Narayan S. Raja and Anil K. Jain. Recognizing geons from superquadrics fitted to range data. In *Image and Vision Computing, v.12, n.3*, pages 179–189, Michigan State Univ., East Lansing, MI, USA, April 1992.

[49] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. In *Computer Graphics and Image Processing*, volume 1, pages 244–256, 1972.

[50] Luc Robert. Camera calibration without feature extraction. In *Proceedings - International Conference on Pattern Recognition*, volume 1, pages 704–706, 1994.

[51] A. Rosenfeld. Image analysis: Problems, progress, and prospects. In *Pattern Recognition*, volume 17, pages 3–12, 1984.

[52] John C. Russ. *The image processing handbook*. CRC Press, Inc., 2nd edition, 1995.

[53] Alireza Sarabi and J. K. Aggarwal. Segmentation of chromatic images. In *Pattern Recognition*, volume 13, pages 417–427, 1981.

[54] Yukio Sato and Otsuki Masaki. Three-dimensional shape reconstruction by active rangefinder. In *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 142–147, 1993.

[55] John Stewman and Kevin. Bowyer. Creating the perspective projection aspect graph of polyhedral objects. In *Second Int Conf on Comput Vision*, pages 494–500, 1988.

[56] John H Stewman and Kevin W. Bowyer. 'viewing sphere' approach to computer vision. In *Conference Proceedings - IEEE SOUTHEASTCON*, pages 25–29, 1987.

[57] John H Stewman and Kevin W. Bowyer. Direct construction of the perspective projection aspect graph of convex polyhedra. In *Computer Vision, Graphics, and Image Processing*, volume 51, pages 20–37, 1990.

[58] M.J. Swain and D.H. Ballard. Color indexing. In *International Journal of Computer Vision*, volume 7, pages 11–32, 1991.

[59] A. M. Treisman. Preattentive processing in vision. In *Computer Vision, Graphics, and Image Processing*, volume 31, pages 156–177, 1985.

[60] Toshio Uchiyama and Michael A. Arbib. Color image segmentation using competitive learning. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 16, pages 1197–1206, 1994.

[61] Kazunori Umeda, Kenji Ikushima, and Tamio Arai. Fusion of range image and intensity image for 3d shape recognition. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 1, pages 680–685, 1996.

[62] Richard Volpe, Todd Litwin, and Larry Matthies. Mobile robot localization by remote viewing of a colored cylinder. In *IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 257–263, 1995.

[63] Karin Wall and Per-Erik Danielsson. A fast sequential method for polygonal approximation of digitized curves. In *Computer Vision, Graphics, and Image Processing*, volume 28, pages 220–227, 1984.

[64] Reg G Willson and Steven A Shafer. What is the center of the image? In *Journal of the Optical Society of America*, volume 11, pages 2946–2955, 1994.

[65] Kenong Wu and Martin D. Levine. 3-d object representation using parametric geons. In *Technical Report TR-CIM-93-13*, McGill Univ., Montreal, PQ, Canada, Sept 1993.

[66] Kenong Wu and Martin D. Levine. Shape approximation: from multiview range images to parametric geons. In *Proceedings - International Conference on Pattern Recognition*, volume 1, pages 622–625, 1994.

[67] J. S. Zelek. *SPOTT: A Real Time, Distributed and Scalable Architecture for Autonomous Mobile Robot Control.* PhD thesis, McGill University, 1996.