

A Probabilistic Approach to Autonomous Planetary Science with Prime and Scout Rovers

by

Marc Gallant, B.Eng

A Thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario, Canada

August 2011

Copyright ©

2011 - Marc Gallant, B.Eng



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-83036-9
Our file *Notre référence*
ISBN: 978-0-494-83036-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Thesis

**A Probabilistic Approach to Autonomous Planetary Science
with Prime and Scout Rovers**

Submitted by **Marc Gallant, B.Eng**
in partial fulfilment of the requirements for the degree of
Master of Applied Science

A.Ellery, Co-supervisor

J. A. Marshall, Co-supervisor

M. Yaras, Department Chair

Carleton University

2011

Abstract

Autonomous science augments the capabilities of planetary rovers by shifting the identification and selection of science targets from the operator to the rover itself. This shift frees the rover of unnecessary idle time as it waits for commands. Additionally, as features can now be evaluated based on their scientific value, data collection can become more selective—allowing only the most valuable data to use the limited bandwidth between the rover and operator.

This thesis presents an autonomous science system that is comprised of three components: a Bayesian network that uses image data to identify features, an evaluation algorithm that selects the best identified feature, and a path-planning algorithm that guides the rover to the most scientifically valuable feature. As autonomous science is a relatively young field, efforts to date include partial implementations (e.g., identification or sorting without evaluation) or lack a machine learning component.

In addition to the development of an autonomous science system, one of the primary contributions of this thesis is an investigation into the effectiveness of pairing a larger prime rover with a smaller scout rover for improved autonomous science. In this scenario, the scout rover provides the prime rover with a science-driven feature map that can be used for path-planning, obstacle avoidance and science target selection. This scenario was compared to the traditional scenario of the prime rover solo in a simulated environment, which revealed the strengths and weaknesses of the two approaches.

The probabilistic identification and evaluation of objects using the Bayesian network and evaluation algorithm was validated experimentally in a real environment. Quantitative results measure the effectiveness of the approach while qualitative results reveal its behaviour and indicate possible areas of improvement. Together, the simulated and real tests provide suggestions for future research.

Acknowledgments

I would personally like to thank my supervisors, Alex Ellery and Joshua Marshall, for their guidance, inspiration and motivation. My interest in space is due in no small part to the entertaining and thought-provoking lectures Alex provided in his classes. As for robotics, Josh's enthusiasm and competency on the topic managed to turn a passing interest into what I hope will someday be my livelihood. It was only natural I would pursue space robotics as a thesis topic to combine both these interests and keep Josh and Alex around for the duration of my degree.

I would like to extend my gratitude to the Kapvik rover team at Carleton, a group which I was privileged to be part of. The wide range of expertise that was freely available by being part of such a talented team was tremendously helpful. In particular, I would like to thank Adam Mack for providing the vision data that was used in this thesis, and Rob Hewitt and Tim Setterfield for providing their insight on everything from technical issues to beer suggestions. Although not part of the Kapvik team, I would also like to thank Stefan Radacina Rusu for his consultation and support.

Finally, I owe a very special thanks to my parents, Maurice and Donna, for their support and interest in my work, and to Natalie MacDonald for her understanding, encouragement and for instilling confidence in me when I needed it most.

Table of Contents

| | |
|--|------------|
| Abstract | iii |
| Acknowledgments | v |
| Table of Contents | vi |
| List of Tables | x |
| List of Figures | xi |
| Nomenclature | xiv |
| 1 Introduction | 1 |
| 1.1 Motivation | 2 |
| 1.2 Scope | 3 |
| 1.2.1 Detecting, Localizing and Identifying Features | 4 |
| 1.3 Goals | 5 |
| 1.4 Overview | 6 |
| 2 Background and Literature Review | 8 |
| 2.1 Path-planning | 8 |
| 2.1.1 Cost Maps | 9 |
| 2.1.2 A* | 12 |

| | | |
|----------|---|-----------|
| 2.1.3 | Theta* | 13 |
| 2.1.4 | Fast Re-planning | 17 |
| 2.2 | Bayesian Networks | 18 |
| 2.2.1 | Bayes' Theorem | 21 |
| 2.2.2 | Network Modelling | 22 |
| 2.2.3 | Worlds | 24 |
| 2.2.4 | World Probability Updates with Evidence | 24 |
| 2.2.5 | Initializing and Training | 27 |
| 2.3 | Fuzzy Logic | 28 |
| 2.3.1 | Membership Functions | 30 |
| 2.3.2 | Initializing and Optimizing Membership Functions | 33 |
| 2.4 | Other Topics | 36 |
| 2.4.1 | Machine Vision | 36 |
| 2.4.2 | State Estimation with the Extended Kalman Filter | 38 |
| 2.4.3 | Velocity-constrained Control | 38 |
| 2.5 | Literature Review on Autonomous Science | 39 |
| 2.5.1 | Thompson, Niekum, Smith and Wettergreen [1] | 40 |
| 2.5.2 | R. Castano, Estlin, Anderson, Gaines, A. Castano, Bornstein, Chouinard and Judd [27] | 44 |
| 2.5.3 | Woods, Shaw, Barnes, Price, Long and Pullan [28, 29] | 47 |
| 2.5.4 | Other Work in Autonomous Science | 51 |
| 2.5.5 | Remarks | 52 |
| 3 | Algorithm Development | 53 |
| 3.1 | Algorithm Overview | 53 |
| 3.1.1 | Prime Rover Solo | 54 |
| 3.1.2 | Prime/scout Rover Pairing | 56 |

| | | |
|----------|---|-----------|
| 3.2 | Feature Position Estimation | 58 |
| 3.2.1 | Feature and Measurement Model | 58 |
| 3.2.2 | Position Estimation with an EKF | 60 |
| 3.3 | Terrain Discretization and Mapping | 63 |
| 3.3.1 | Occupancy Grid Mapping | 63 |
| 3.3.2 | Science Vertices | 66 |
| 3.4 | Feature Identification | 68 |
| 3.4.1 | Bayesian Network Model | 68 |
| 3.4.2 | Initializing the Fuzzy Membership Functions | 69 |
| 3.4.3 | Training the Bayesian Network | 69 |
| 3.4.4 | Entering Evidence | 72 |
| 3.5 | Feature Evaluation | 73 |
| 3.5.1 | Benefit Bias Equation | 73 |
| 3.5.2 | Filtering and Selecting Features | 76 |
| 4 | Test Environments | 80 |
| 4.1 | MATLAB Simulator | 80 |
| 4.1.1 | Rover Model | 81 |
| 4.1.2 | Environment | 82 |
| 4.1.3 | Bayesian Network | 82 |
| 4.1.4 | Machine Vision | 84 |
| 4.1.5 | Test Scenarios | 84 |
| 4.2 | Laboratory | 85 |
| 4.2.1 | Hardware | 85 |
| 4.2.2 | Features | 88 |
| 4.2.3 | Bayesian Network | 88 |
| 4.2.4 | Test Trials | 89 |

| | | |
|----------|---|------------|
| 5 | Results and Analysis | 92 |
| 5.1 | Simulator | 92 |
| 5.1.1 | Prime Rover Solo | 93 |
| 5.1.2 | Prime/Scout Rover Pairing | 97 |
| 5.1.3 | Scenario Comparison | 101 |
| 5.1.4 | Possible Improvements | 106 |
| 5.2 | Laboratory | 107 |
| 5.2.1 | Training | 108 |
| 5.2.2 | Feature Identification Performance | 114 |
| 5.2.3 | Feature Evaluation Performance | 120 |
| 5.2.4 | Possible Improvements | 133 |
| 5.2.5 | Transitioning to Natural Features | 134 |
| 6 | Conclusion | 136 |
| 6.1 | Status of Goals | 136 |
| 6.2 | Achievements and Contributions | 138 |
| 6.3 | Future Work | 139 |
| | List of References | 142 |
| | Appendix A Simulation Parameters | 147 |
| | Appendix B Laboratory Test Features | 149 |
| | Appendix C Laboratory Attribute Parameters | 151 |
| | Appendix D Simulation Test Results | 153 |
| | Appendix E Laboratory Training Results | 155 |
| | Appendix F Laboratory Identification Results | 160 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Example of conditional probability tables | 23 |
| 2.2 | Listing of all the possible worlds of a Bayesian network | 25 |
| 2.3 | Updated probability of worlds after evidence is introduced | 26 |
| 2.4 | Updated probability of worlds after soft evidence is introduced | 27 |
| 2.5 | Example of data set used to train a Bayesian network | 29 |
| 3.1 | Training a BN with a continuous data set | 71 |
| 3.2 | The <i>a priori</i> and <i>a posteriori</i> world probabilities with $B = \text{true}$ | 72 |
| 3.3 | Example of results from the first observation of two features | 75 |
| 3.4 | Example of results from the second observation of two features | 75 |
| 4.1 | Point Grey Bumblebee XB3 CCD Camera technical specifications | 86 |
| 4.2 | The importance weights used in the laboratory tests | 91 |
| 5.1 | Quantitative comparison of the two test scenarios | 102 |
| 5.2 | Evaluation results of the first laboratory test trial | 122 |
| 5.3 | Evaluation results of the second laboratory test trial | 124 |
| 5.4 | Evaluation results of the third laboratory test trial | 127 |
| 5.5 | Evaluation results of the fourth laboratory test trial | 130 |
| A.1 | Simulation parameters | 148 |
| C.1 | The trapezoidal parameters for the attribute states in the laboratory tests | 152 |
| D.1 | Detailed simulation test results | 154 |

| | |
|--|-----|
| E.1 Bayesian network training results for six attributes in the laboratory tests | 156 |
|--|-----|

List of Figures

| | | |
|-----|--|----|
| 1.1 | The Kapvik micro-rover | 4 |
| 2.1 | Simple example of a partially populated cost map | 10 |
| 2.2 | Determining the occupancy of vertices in the cost map | 11 |
| 2.3 | Example path whose length is equal to its Chebyshev distance | 13 |
| 2.4 | Comparison of A* and any-angle planned paths | 16 |
| 2.5 | Simple example of a Bayesian network | 23 |
| 2.6 | Two types of fuzzy membership functions | 32 |
| 2.7 | Set of fuzzy membership functions that are not sum normal. | 34 |
| 2.8 | Histogram used to initialize membership functions | 35 |
| 2.9 | Initial membership functions based on data set | 35 |
| 3.1 | Block diagram describing the prime rover solo scenario | 55 |
| 3.2 | Block diagram describing the prime/scout rover pairing scenario | 57 |
| 3.3 | Visualization of the feature measurement model | 59 |
| 3.4 | Visualization of the estimated measurement noise of the camera | 61 |
| 3.5 | Determining the occupancy status of vertices from the state of a feature | 65 |
| 3.6 | Example showing the effect of feature uncertainty on path-planning | 67 |
| 3.7 | Definition of the science vertices of a feature | 68 |
| 3.8 | Generalized model of the BN used in this thesis | 69 |
| 3.9 | Example of the effect of science vertex selection on path-planning | 78 |
| 4.1 | Example rock field used in the simulator | 83 |

| | | |
|------|---|-----|
| 4.2 | Point Grey Bumblebee XB3 CCD Camera | 86 |
| 4.3 | Stereo camera mounted on a mobile robot | 87 |
| 4.4 | Training grid used for the laboratory training set | 89 |
| 4.5 | Example of image used for laboratory training set | 90 |
| 4.6 | The 30 feature locations in the laboratory tests | 91 |
| 5.1 | Example of path and map generated by the prime rover solo | 94 |
| 5.2 | Example of kinks in the path of the prime rover solo | 95 |
| 5.3 | Example of backtracking by the prime rover solo | 96 |
| 5.4 | Example of premature feature selection by the prime rover solo | 97 |
| 5.5 | Examples of paths and the occupancy grid generated by the scout rover | 99 |
| 5.6 | Example of backtracking by the scout rover | 100 |
| 5.7 | Example of imperfect information provided by the scout rover | 101 |
| 5.8 | Histograms illustrating the distribution of the simulator results | 103 |
| 5.9 | Simulation trial with similar results in both scenarios | 105 |
| 5.10 | Comparison of training results for similar objects. | 111 |
| 5.11 | Image comparison of white and yellow plastic balls | 113 |
| 5.12 | Abbreviated identification results in the laboratory tests | 118 |
| 5.13 | Comparison of the training vs. test results for two different objects | 119 |
| 5.14 | Path and scoring results for the first laboratory trial | 123 |
| 5.15 | Path and scoring results for the second laboratory trial | 126 |
| 5.16 | Path and scoring results for the third laboratory trial | 129 |
| 5.17 | Path and scoring results for the fourth laboratory trial | 132 |
| B.1 | The objects used for the laboratory tests. | 150 |
| F.1 | Identification results in the laboratory tests | 167 |

Nomenclature

$\frac{S_b}{L_D}$ Score-to-detour ratio

ϕ The bearing of a feature in the rover reference frame

Σ The normalized standard deviation of the probabilities

$\Pr(\bar{A})$ The probability that event A is false

$\Pr(A)$ The probability that event A is true

$\Pr(A = p)$ The probability that event A equals state p

a, b, c, \dots, n The importance weights for minerals A, B, C, \dots, N

A_j The area of influence of feature j

a_j, b_j The semi-major and semi-minor axes of the ellipse defining A_j

D Detour coefficient

d The unsigned distance between (X_r, Y_r) and (X_f, Y_f)

h The maximum dimension of the rover's footprint

K Benefit bias equation scale constant

$L_{s_a \rightarrow s_b}$ The path length from vertex s_a to s_b

m_D Dynamic benefit score threshold multiplier

| | |
|-------------|---|
| P | The apparent size of a feature |
| p_k | The pose (X_r, Y_r, θ_r) of the rover in the global coordinate frame at time k |
| q_k | The state (X_f, Y_f) of a feature at time k |
| r_j | The estimated radius of feature j |
| r_{sci} | The science radius |
| r_{sen} | The up-close sensor radius |
| S_b | Benefit score |
| $S_{b,min}$ | Threshold benefit score |
| $s_{q,i}$ | The i^{th} science vertex of feature q |
| y_k | The measurement (d, ϕ) at time k |

AEGIS Autonomous Exploration for Gathering Increased Science System

ANN Artificial neural network

BN Bayesian Network

BT Bayes' Theorem

BWL Bowl

CPT Conditional probability table

DAG Directed acyclic graph

EKF Extended Kalman filter

IDM Inverse difference moment

JPL Jet Propulsion Laboratory

LFB Large football

LPT Large pot

LSB Large soccer ball

MF Membership function

MPT Medium pot

OASIS Onboard Autonomous Science Investigation System

PVB Pink volleyball

SBP Small brown pot

SFB Small football

SGP Small green pot

SLAM Simultaneous localization and mapping

SSB Small soccer ball

SVS Science value score

UKF Unscented Kalman filter

WPB White plastic ball

WVB White volleyball

YPB Yellow plastic ball

Chapter 1

Introduction

Robotic explorers have had a critical role in the exploration of planetary bodies in our solar system. Surface exploration of planetary bodies began with the robotic Luna missions on the Moon by the Soviet Union, with the first successful landings in the early 1960s. However, unlike near-Earth missions such as lunar exploration, missions to other planetary bodies become inefficient if a level of autonomy is not granted to the robot. The great distances between robotic explorers and their Earth-based operators means there are limited opportunities for communication and considerable time-delays in the issuing of commands. Limited autonomy results in robots that sit idle as they wait to receive commands or send telemetry.

The current state-of-the-art method of surface exploration of planetary bodies is mobile rovers equipped with a selection of tools and sensors that can measure and manipulate its environment. The mobility of rovers allows for large areas to be explored and documented, enabling a better understanding of geologic trends and subtleties. Past rovers such as NASA's Mars Exploration Rovers have greatly advanced our knowledge of that planet, and other better-equipped rovers are currently scheduled to follow in their footsteps. Improved rover technology means this next generation of rovers will have a far greater range and data collection capabilities than any of their predecessors. However, limited bandwidth and narrow communication

windows create a bottleneck in the transmission of data. The advantages of these new rovers would be greatly enhanced if they were permitted to autonomously observe their local surroundings and make decisions based on those observations, reducing idle time and ensuring only the most scientifically valuable data is collected.

To this end, a new field called *autonomous science* is emerging, which grants the rover science-influenced decision-making abilities that were previously reserved for Earth-based operators. With these systems, rovers use exteroceptive sensors to evaluate the scientific value of their surroundings and use that information to influence their actions. The most promising scientific targets can be autonomously identified, and more detailed measurements can be taken without operator intervention. This addresses both the idle time and transmission bottleneck issues that afflict non-autonomous systems.

Autonomous science systems to date have focused on two key areas: the detection, identification/classification and evaluation of features, and the integration of evaluated features into the planned activities of the rover. The first area has been nearly universally addressed using image processing (although expansion to include other sensors is possible), partly because cameras are nearly universally included on rover platforms. The second area has been investigated to a far less degree, and mostly has been integrated into existing planner platforms. Overall, autonomous science systems are a relatively novel development and have extremely limited heritage, due in part to the scarcity and cautious nature of rover missions.

1.1 Motivation

As autonomous science systems are a relatively recent development for planetary rovers (and are actively being developed), approaches to detecting, identifying/classifying and evaluating features have not been standardized. However, one

system [1] has been sufficiently developed to be incorporated into the Autonomous Exploration for Gathering Increased Science System (AEGIS), which was uploaded to the MER rover Opportunity in 2010. A unique aspect of this system was its use of machine learning. Bayesian networks (BNs) were used to detect and classify rocks into classes; however, to date it does not use this information to influence the actions of the rover. This thesis will continue the investigation of classifying features using a BN, but will evaluate the classified features and use this information to make guidance-related decisions.

Even with the inclusion of autonomous science, the high cost of rover missions would restrict the collection of scientific data by conservatively having the rover avoid risky targets that could cause potentially mission-ending scenarios. Also, the energy used by a large, multi-sensor, rover (i.e., the *prime* rover) is better spent performing scientific analysis than searching for targets. To this end, it would be beneficial to pair a primary rover with one or more micro-rover “scouts”. These lightweight, quick, rugged and possibly expendable scouts could explore, map and evaluate areas of potential scientific interest. They could also afford to take more risks in finding valuable scientific targets. An example of a rover that fits this description is the *Kapvik* micro-rover that is currently under development in part by Carleton University (Figure 1.1). This thesis will investigate how a scout rover equipped to perform autonomous science could aid the primary rover in its collection of valuable science measurements.

1.2 Scope

The scope of this thesis can be divided into three separate but related processes: (i) detecting, localizing, and identifying features, (ii) evaluating features based on their estimated identity and their location, and (iii) using the evaluated features to

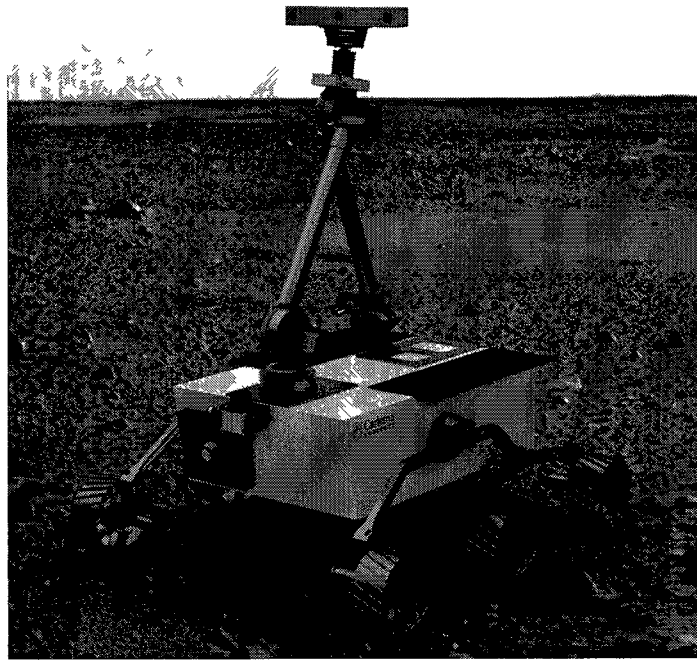


Figure 1.1 – A rendering of the Kapvik micro-rover currently under development in part by Carleton University. Its footprint is approximately 0.782×0.850 m.

influence the guidance systems of a prime and scout rover. For the first process, the focus of the thesis does not encompass the all aspects of the process and is subjected to some assumptions. The focal point of this process and the assumptions that were made are clarified below.

1.2.1 Detecting, Localizing and Identifying Features

An image captured by a stereo camera is processed to detect features, localize them relative to the rover, and estimate their physical attributes. The work performed for this thesis did not include the development of the algorithms that perform these steps. However, suitable algorithms were made available from the output of collaborative work within the author's research group. This is detailed later in the thesis. A BN was developed to identify features based on physical attributes extracted from the processed images. For simplification, the BN was limited to discrete rather than

continuous variables. This assumption reflects the type of observations a geologist would make in the field, as geologic features are usually identified qualitatively (e.g., dark, pitted, layered, etc.) and can vary slightly among rocks of the same type. Exact continuous measurements (e.g., 36 % colour intensity, 19 % pit coverage, average bedding thickness of 4.4 mm) are often unnecessary for identification.

When features are localized relative to the rover, two assumptions were made that simplify the process of turning this relative localization into a global feature map. First, to transform feature locations from the rover to the global reference frame, the estimated position of the rover in the global reference frame is required. This thesis assumes that this information is known. The required algorithms that perform this type of estimation are widely used, including extended Kalman filter simultaneous localization and mapping (EKF SLAM), unscented Kalman filter (UKF) SLAM and FastSLAM [2]. These algorithms also recursively estimate the position of the features and their uncertainty. Although the localization of the rover was assumed, feature localization was developed for this thesis (Section 3.2). Second, it is assumed that features detected in a measurement can be matched with previous measurements (data association). Data association is often required in conjunction with SLAM, and as a result, has been developed for specialization in that area [3]. Overall, a SLAM algorithm that includes a data association component would be a good approximation of the information gained from the assumptions made above.

1.3 Goals

Autonomous science systems usually require several algorithms working in harmony, including: machine vision, image processing, machine learning, guidance, navigation and control. As was described in Section 1.2, the scope of this thesis is to develop and integrate a machine learning technique with a guidance algorithm. This work

was done in collaboration with machine vision and image processing algorithms, developed separately but in conjunction with this thesis. As is discussed in Section 2.5, developments in the relatively novel field of autonomous science combine different approaches to the algorithms mentioned above. Often, only specific parts of the algorithms are developed. Therefore, there remains many approaches to this problem that have not been developed. As a result, the goal of this thesis is to develop, combine and test aspects of autonomous science that have not previously been developed. This goal will attempt to produce a system with the following properties:

- (i) The rover should be capable of identifying (i.e., not just sorting) several different objects via the machine vision, image processing and machine learning techniques.
- (ii) The machine learning technique should be expandable and tuneable to include new feature types.
- (iii) A map of identified features should be built with sufficient accuracy that a rover can safely plan efficient paths from one point in the map to another.
- (iv) The rover should be able to identify features, plan paths, and drive to goals without any intervention from an operator.

1.4 Overview

The next chapter provides background information on topics that are used in the development of the algorithms used in this thesis. Sufficient detail is provided on the most critical topics; however, references are provided for more detailed explanations and for processes not critical to the scope of the thesis. A literature review on autonomous science systems, with a focus on the three latest large-scale efforts, is also presented.

Chapter 3 outlines how the overall system merges several algorithms together into an autonomous science system. It provides the theory and details on the function of each of these algorithms. This includes how features are localized and mapped, and how their attributes are estimated. The methods used to evaluate the features are discussed, and there is an overview of the method in which these evaluations are incorporated into the guidance system of a rover.

Chapter 4 describes the environments in which the algorithms were tested, as well as the conditions of those tests. A full implementation of the system was simulated using MATLAB, which was tuned using the results of real-life tests. The hardware used in the laboratory testing is specified and the features that defined the environment are described. The laboratory testing focused on the machine vision and machine learning aspects of the system.

The results of the full-implementation simulated testing and the partial implementation laboratory testing are presented in Chapter 5. The results of the simulated tests focus on the merging of the machine learning and guidance algorithms, specifically outlining their effects on different rover exploration scenarios. The results of the laboratory tests focus on the tuning and performance of the machine vision and machine learning algorithms.

Finally, Chapter 6 summarizes the content of the thesis and adds concluding remarks about the contributions that were made. A few suggestions are made on possible directions of future work.

Chapter 2

Background and Literature Review

This chapter provides background material for the algorithm development that is presented in Chapter 3. The essential information necessary to implement the path-planning algorithms used in this thesis are presented in Section 2.1. An overview of the aspects of Bayesian networks that were necessary for the development of the feature identification algorithm is shown in Section 2.2, including an example on how they can be trained from data sets. Background information on fuzzy logic and how it can be used with Bayesian networks is overviewed in Section 2.3. Other topics that were necessary but not the focus of the thesis are presented in Section 2.4, which includes machine vision, state estimation with the extended Kalman filter and velocity-constrained control. Finally, Section 2.5 presents a literature review on previously developed autonomous science systems, with a focus on the last three major efforts. Also included are remarks on the relevance of previous work to the goals of this thesis.

2.1 Path-planning

All autonomous planetary rovers to date have used graph-based path-planning algorithms. This includes Sojourner [4] and Spirit and Opportunity [5]. NASA's next

Mars rover, Curiosity, will also use this type of algorithm [6]. One advantage of using graph-based path-planning is its guarantee to find a solution if such a solution exists. It also provides flexibility on how the terrain is described in the form of *cost maps*. Cost maps are formed by discretizing the terrain into vertices and assigning each vertex a traversal cost. Once the rover is localized to a vertex in this map, a goal vertex is specified. The path planner plans the optimal path (i.e., the minimal cost based on the cost map) from the rover vertex to the goal vertex. A widely used algorithm to plan graph-based paths is A* [7], from which many modern algorithms have been derived. One such derivation is Theta* [8], which enhances A* by allowing any-angle path-planning.

2.1.1 Cost Maps

A cost map is used to describe the cost of traversing different features in the terrain. For example, the cost of travelling over a rock could be a function of its size, the cost of climbing a slope could be a function of its grade, or the cost of navigating crevasses could be a function of their depth. Deriving the functions that describe these cost-terrain relationships depend on the chassis design of the rover and would be determined with testing. The cost itself is often described in terms of power, energy, or distance.

To create a cost map, the terrain must be discretized into a grid of vertices. A simple illustration of this process is shown in Figure 2.1. The resolution of the grid is selected based on the computational capabilities of the rover. A smaller resolution allows for more precise path-planning, but requires more calculations (as more paths are possible). Also, the connectivity of the grid (i.e, how nodes are connected to their neighbours) can affect the possible travel headings of the rover if paths are limited to the vertex connections. In this thesis, an eight-connected grid (i.e., each vertex has eight neighbours as shown in Figure 2.1) is used; however, graph-based path-planning

can be used with any degree of connectivity.

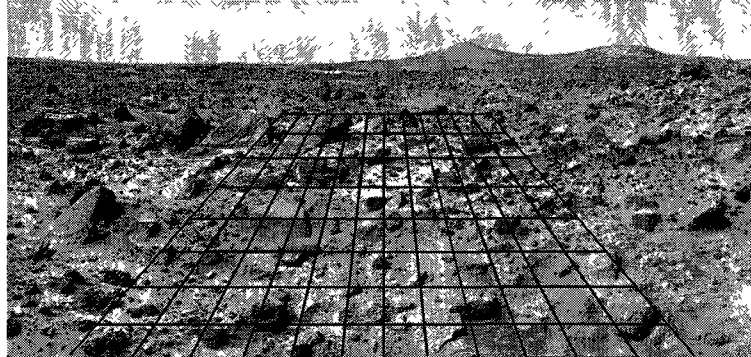


Figure 2.1 – A simplified example of a partially populated cost map (background image from [9]).

Each vertex in the cost map must be given a traversal cost. These costs are usually derived from observations made with exteroceptive sensors while also considering capabilities of the rover’s chassis. Testing must be done to accurately describe the relative costs of travelling over different types of terrain. For example, if tests showed that the rover used 30 % more power to travel over a rock that is half a wheel diameter in height, then the cost of vertices containing these rocks must be adjusted accordingly relative to empty vertices. Note that useful cost calculations rely on the accuracy of the sensors making the observations (e.g., using a camera to measure the height of a rock), as well as the accuracy of the localization of the rover (to properly place the coordinates of the rock in the cost map).

To accommodate a graph-based path planner, the position of the rover is usually reduced down to one vertex. To validate this alteration, half of the largest dimension of the rover’s footprint is added to the size of features. This allows for paths to be planned while maintaining a safe distance from features. Furthermore, as the edges of features rarely coincide with the edges between vertices, vertices partially occupied by features are considered fully occupied as a safety measure to ensure adequate distance from features is maintained. This process is illustrated in Figure 2.2.

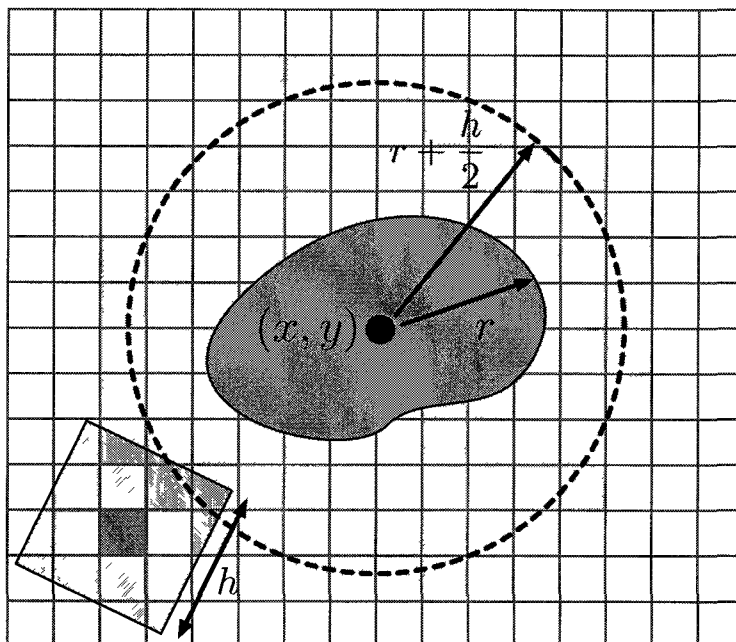


Figure 2.2 – Expanding the size of a feature by half of the largest dimension of the rover's footprint. The rover is represented by the square, and the shaded vertices are those whose costs are affected by the feature. (x, y) is the estimated location of the feature, r is the estimated radius of the feature, and h is the rover's largest dimension.

2.1.2 A*

A* path-planning is an extension of Dijkstra's algorithm that improves calculation time by using heuristics. It begins by expanding the start vertex to select the best neighbour to expand next. Expanding the vertex s consists of selecting among the unexpanded (and reachable) neighbours of s the vertex s' with the lowest heuristic cost $f(s')$. This cost is calculated with the function

$$f(s') = g(s') + h(s'), \quad (2.1)$$

where $g(s')$ is the cost of travelling from the start vertex s_{start} to s' and $h(s')$ is an admissible heuristic of the cost of travelling from s' to the goal vertex s_{goal} . This means that $h(s')$ must always be less than or equal to the actual minimum cost. However, maximizing $h(s')$ can decrease the number of expansions that will be required to calculate the path. As $h(s') \rightarrow 0$, the speed advantage of using A* in place of Dijkstra's algorithm decreases, and A* essentially becomes Dijkstra's algorithm when $h(s') = 0$. For eight-connected grids, the Chebyshev distance is often used to calculate $h(s')$; i.e.,

$$h_{straight}(s') = |s'_x - s_{goal_x}| + |s'_y - s_{goal_y}| \quad (2.2a)$$

$$h_{diag}(s') = \min(|s'_x - s_{goal_x}|, |s'_y - s_{goal_y}|) \quad (2.2b)$$

$$h(s') = R_g \left[h_{straight}(s') + (\sqrt{2} - 2)h_{diag}(s') \right] \quad (2.2c)$$

where R_g is the resolution of the grid. In this example, the cost of travelling between vertices is assumed to be the distance between them (i.e., the cost of travelling to a vertically or horizontally adjacent vertex is R_g , and the cost of travelling to a diagonally adjacent vertex is $\sqrt{2}R_g$). This will always yield $h(s') \leq cost(s', s_{goal})$. Figure 2.3 shows the case where $h(s') = cost(s', s_{goal})$. Note that multiple paths exist

that yield the same Chebyshev distance.

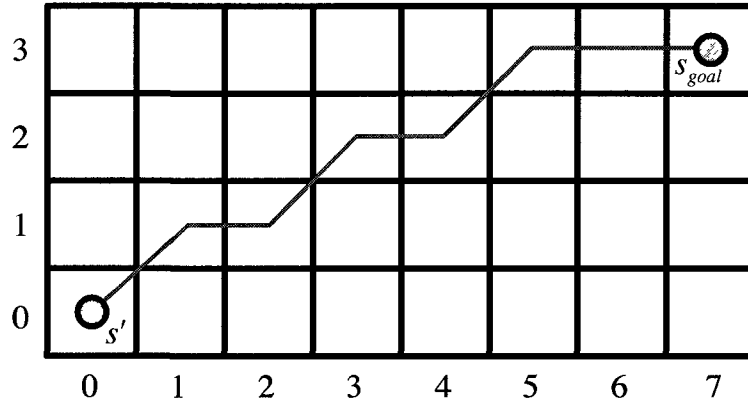


Figure 2.3 – One of many variations of the shortest path between s' and s_{goal} whose path length is also the Chebyshev distance ($\approx 8.243R_g$ in this example).

The algorithm continues to expand vertices until s_{goal} is selected in an expansion. The optimal path is extracted by starting from s_{goal} and repeatedly selecting the vertex that led to its expansion (its *parent* vertex) until s_{start} is found. The above describes the general concept of A*, with some details omitted for brevity (e.g., updating $g(s')$ if the latest expansion of one of its neighbours creates a better path from the start to s'). A full description is shown in Algorithm 1. The *open list* in Algorithm 1 is a list of vertices that are in consideration for expansion (i.e., they were a neighbour of an expanded vertex at least once), and the *closed list* is a list of vertices that have been expanded.

2.1.3 Theta*

A major restriction of using A* path-planning is constricting movement to only the neighbours of the currently occupied vertex. In an eight-connected grid, this means the heading of a path is limited to increments of $\frac{\pi}{4}$. To remedy this restriction, any-angle path-planning has been developed that does not require paths to be limited to the edges between vertices. An example of the path improvement of any-angle

```

Input: Cost map, start vertex  $s_{start}$ , goal vertex  $s_{goal}$ .
Output: Optimal list of vertices connecting  $s_{start}$  and  $s_{goal}$ .
1 Main()
   // Initialize the open and closed lists, then add the start to
   // the open list.
2    $open = closed = \emptyset$ ;
3    $g(s_{start}) = 0$ ;
4    $parent(s_{start}) = s_{start}$ ;
5    $open.Insert(s_{start}, g(s_{start}) + h(s_{start}))$ ;
   // While the open list is not empty, expand the vertex  $s$  with
   // the lowest cost  $f(s)$ .
6   while  $open \neq \emptyset$  do
7      $s = open.LowestF()$ ;
8      $open.Remove(s)$ ;
9     if  $s = s_{goal}$  then
10    |    $ExtractPath()$ ;
11    |   return "path found";
12    end
13     $closed.Insert(s)$ ;
14    foreach  $s' \in neighbor_{visib}(s)$  do
15    |   if  $s' \notin closed$  then
16    |   |   if  $s' \notin open$  then
17    |   |   |    $g(s') = \infty$ ;
18    |   |   |    $parent(s') = NULL$ ;
19    |   |   end
20    |   |    $UpdateVertex(s, s')$ ;
21    |   end
22    end
23  end
24  return "no path found";
25 end

```

Algorithm 1: The A* path-planning algorithm [7]. The functions **UpdateVertex**, **Compute** and **ExtractPath** are shown in Algorithm 2.

```

27 UpdateVertex( $s, s'$ )
    // Updates  $g(s')$  if the cost of travelling from  $s$  to  $s'$  is lower
    // than any other paths seen previously. If  $g(s')$  is improved,
    // add  $s'$  to the open list.
28    $g_{current} = g(s')$ ;
29   ComputeCost( $s, s'$ );
30   if  $g(s') < g_{current}$  then
31     if  $s' \in \text{open}$  then
32       |  $\text{open.Remove}(s')$ ;
33     end
34      $\text{open.Insert}(s', g(s') + h(s'))$ ;
35   end
36 end

37 ComputeG( $s, s'$ )
    // If the cost of travelling from  $s$  to  $s'$  is lower than any
    // other paths seen previously, update  $g(s')$  and make  $s$  its
    // parent.
38   if  $g(s) + \text{cost}(s, s') < g(s')$  then
39     |  $\text{parent}(s') = s$ ;
40     |  $g(s') = g(s) + \text{cost}(s, s')$ ;
41   end
42 end

43 ExtractPath()
    // Extracts the path by starting at the goal and travelling
    // through the ‘‘family tree’’ of vertices (i.e., always
    // moving to the parent of a vertex) until  $s_{start}$  is found.
44    $\text{path} = \emptyset$ ;
45    $\text{currentVertex} = s_{goal}$ ;
46   while  $\text{currentVertex} \neq s_{start}$  do
47     |  $\text{path.Append}(\text{currentVertex})$ ;
48     |  $\text{currentVertex} = \text{parent}(\text{currentVertex})$ ;
49   end
50 end

```

Algorithm 2: The functions **UpdateVertex**, **Compute** and **ExtractPath** used by A* in Algorithm 1.

path-planners is shown in Figure 2.4. There are two predominant any-angle path planners for graph-based cost maps: Theta* and Field D*. Field D* [10] does not edge-constrain paths by linearly interpolating the cost of crossing an edge based on the costs of the vertices the edge is connecting. Theta* [8] achieves a similar result by allowing the parent of a vertex to be any vertex to which it has line-of-sight. In this context, two vertices have line-of-sight if it is possible to connect them with a straight line that does not cross any untraversable vertices. Field D* and Theta* have been compared for paths planned on randomly generated cost maps [8]. It was found that Theta* found shorter paths than Field D*, and gave the best overall trade-off between runtime and path length. For these reasons, along with its algorithmic simplicity, Theta* was selected as the path-planner for this thesis.

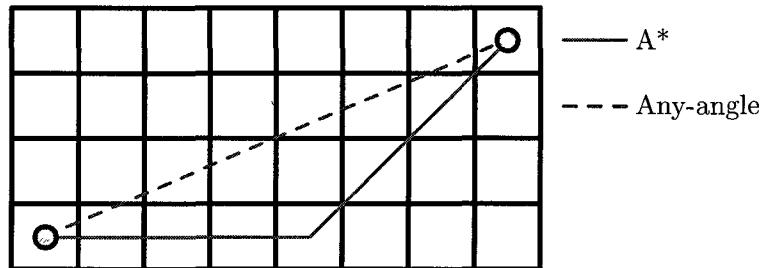


Figure 2.4 – Any-angle path-planning generates shorter paths than A* by eliminating the restriction that paths must travel along graph edges.

There are minimal algorithmic differences between A* and Theta*. The primary difference is how $g(s')$ is calculated (i.e., function **ComputeG** in Algorithm 2). Rather than only considering the path from s to s' , the cost from $parent(s)$ to s' is also considered if there is line-of-sight between $parent(s)$ and s' . This change is shown in the updated version of **ComputeG** shown in Algorithm 3 (note that all other functions are identical to those used in Algorithm 1).

In addition to the changes to **ComputeG**, the admissible heuristic $h(s')$ must be updated. Now that paths can be planned at any angle, the Chebyshev distance will not always be less than the path length planned by Theta*. As a result, $h(s')$ should

```

37 ComputeG( $s, s'$ )
    // If the cost of travelling from  $s$  to  $s'$  (or from  $parent(s)$  to
    //  $s'$  if there is line-of-sight between them) is lower than any
    // other paths seen previously, update  $g(s')$  and make  $s$  (or
    //  $parent(s)$ ) its parent.
38 if LineOfSight( $parent(s), s'$ ) then
39     if  $g(parent(s)) + cost(parent(s), s') < g(s')$  then
40          $parent(s') = parent(s);$ 
41          $g(s') = g(parent(s)) + cost(parent(s), s');$ 
42     end
43 else
44     if  $g(s) + cost(s, s') < g(s')$  then
45          $parent(s') = s;$ 
46          $g(s') = g(s) + cost(s, s');$ 
47     end
48 end
49 end

```

Algorithm 3: The new ComputeG function for Theta*.

now be the cost of a straight line between s' and s_{goal} , as this path is possible when planning with Theta*.

To calculate if there is line-of-sight between two vertices, Bresenham's line algorithm [11] is commonly used, which was originally developed as a line-drawing algorithm in computer graphics. This algorithm indicates which vertices are crossed when drawing a straight line through a Cartesian grid. If none of the crossed vertices are untraversable, line-of-sight exists. The algorithm uses only logical and integer operations, which are beneficial in speeding up repeated line-of-sight checks.

2.1.4 Fast Re-planning

In mobile robotics, "fast re-planning" versions of any-angle path planners are used to update paths when there are changes to the cost map. This is beneficial when rovers are navigating previously unexplored environments, as new observations must

constantly be reflected as changes in the cost map. Without fast re-planning, no information is retained from previous planned paths and a new path must be planned each time there is a change in the cost map. With fast re-planning, vertex information is stored and re-used after the cost map has been updated. If the changes to the cost map do not greatly affect the previous path, a path is re-planned in much less time than the original. As changes to the cost map become greater and more chaotic, the re-plan time approaches that of the original planning time. There are fast re-planning versions of Field D* [10] and Theta* [12] (known as Incremental Phi*).

It should be noted that fast re-planning Field D* has substantial heritage in rover missions [4, 5, 6]. This alone is a convincing proponent in using a similar algorithm in the autonomous science system presented in this thesis. However, extending Theta* into the fast re-planning Incremental Phi* is certainly a viable option that lacks heritage simply because it is a relatively recent (2009) development. It is recommended that future refinements to the autonomous science system weigh the benefits of the two systems prior to implementation.

2.2 Bayesian Networks

Granting greater autonomy to a robot requires some form of machine learning, such that useful information gained from sensors or empirical data can positively influence the behaviour of the robot. In mobile robotics, exteroceptive sensors are often used to interpret the local environment surrounding the robot, and that information is used to influence motion. One use of machine learning is classification, where the data from these sensors (e.g., camera, laser scanner) is algorithmically placed in a pre-defined category. *Probabilistic classification* enhances this process by calculating the probability that the data belongs in each category. These probabilities can be used to measure how well data fits into a category, which allows for the preservation

of multiple hypotheses that can be improved with new measurements. This benefit would be very useful for rover-based mineral classification as observations made by the rover are inherently uncertain and change depending on many conditions. For these reasons, probabilistic classification was selected to categorize mineral types.

A commonly used method for autonomous decision-making and classification is rule-based expert systems. These algorithms classify data through a nodal flow chart, where each node represents a query about the input data, much like a human expert would approach this problem. However, there are weaknesses to this approach that would cause problems in developing it for probabilistic classification. These include difficulties in using training sets to refine the model and also adapting the model to handle missing inputs [13]. Unfortunately, adapting the system to different environments will require different training sets; therefore, it is desirable that this process be simple and efficient to enhance flexibility. Also, there is no guarantee a full set of inputs will be available at each step of the classification. For these reasons, this type of classification was not explored further.

A review of the literature showed two prevailing forms of environment classification in rover-based systems: Bayesian networks (BNs) and artificial neural networks (ANNs). BNs have been used for both detection and classification of rocks on a rover platform [1] and ANNs have been used to detect carbonates using Mars analogue targets [14]. To select the most appropriate method, the following criteria were considered: (i) accuracy, (ii) computational requirements, (iii) ease of incorporating *a priori* knowledge, (iv) human interpretability, and (v) heritage. The following comparisons concerning these criteria are largely based on the work of Zhang and Bivens [15].

Accuracy

BNs have been shown to be more accurate than ANNs if only small sets of training data are available. This becomes less pronounced as the size of data

sets increases, but is amplified if the model is using discrete (as opposed to continuous) data sets.

Computational Requirements

The computational time of ANNs remains nearly static despite an increase in the size of data sets. On the contrary, BNs have a nearly linear relationship of decreasing speed as the size of data sets increases. As a result, if there is a possibility of requiring a model to be expanded in the future, the increase in computation time for BNs must be considered in the initial selection of the classification algorithm.

Ease of Incorporating *A Priori* Knowledge

When developing BNs, the associativity and dependence of events can easily be incorporated into the model. The graphical representation of BNs makes this process relatively simple. Also, the integration of known probabilities into the model is straightforward. Likewise, methods have been developed to incorporate *a priori* knowledge into ANNs. This is done by introducing a hidden layer of neurons with an equal number of neurons as the input layers. Associativity between neurons is done by forming connections between the input neurons and all the associated neurons in the hidden layer.

Human Interpretability

A BN is usually depicted as a directed acyclic graph (DAG), whose connections between events show direct causal relationships (i.e., an event at the head of a directed edge causes the event at the tail end). This makes human interpretability of BNs relatively intuitive. Conversely, the complexity of ANN models can hinder human interpretation (sometimes considered a “black box” classifier).

Heritage

Rock classification with a rover-mounted camera in a natural environment has been shown to be effective using a BN [1]. ANNs have been used for detecting carbonates and other minerals in rocks in a Mars analogue environment [14].

Based on the above criteria, it was decided a BN best suits the current application. The ability to create an accurate classifier with limited training data is a useful advantage of BNs. Also the network will be relatively simple and will use discrete events, negating much of the speed and resource advantages of ANNs. Additionally, the ease of interpretability also greatly aids in the development of the classification algorithm. Finally, past successes using BNs in similar applications creates a useful foundation of knowledge to build on.

2.2.1 Bayes' Theorem

Bayes' Theorem (BT) allows for the calculation of the conditional probability of an event occurring. This probability is conditional based on the knowledge that a separate event has already occurred. For example, suppose it is known that a student finished his/her thesis before a deadline. Using BT, the conditional probability that the student also had time to regularly partake in social activities can be calculated, if a causal relationship between these two events can be described. The most common form of BT is

$$\Pr(S|F) = \frac{\Pr(F|S)\Pr(S)}{\Pr(F)}. \quad (2.3)$$

This thesis uses the format of $\Pr(S = p)$ for the unconditional probability that S is equal to p (where p is known as the *state* of S). If p is absent (i.e., $\Pr(S)$), S is assumed to be binary, where S is true and \bar{S} is false. The format $\Pr(S|F)$ is the probability that event S is true given the knowledge that the event F is true. In the above example, the event “the student has a social life” would be S , while the event “the student finished his/her thesis before the deadline” would be F . Consequently, using

BT requires that it is possible to obtain (or calculate) the inverse probability (i.e., the probability that the student finished his/her thesis on time knowing they had a social life) as well as the unconditional probabilities of these events occurring. Although this example depicts binary events, each event can take on any number of discrete states. For example, instead of “having a social life”, event S could represent the extent of a student’s social life (e.g., $S \in \{\text{hermit, average, party animal}\}$). In general, BT adjusts the probability of the different states of an event based on evidence.

2.2.2 Network Modelling

In complex systems (i.e., many events, dependencies), it becomes difficult to keep track of inter-dependencies among events such that BT (and its derivations) can be applied to calculate conditional probabilities. A solution to this problem is organizing the events in a BN, which uses a directed acyclic graph (DAG) to show the dependence (and independence) of a collection of events. Each event is represented by a node on the graph, and every node is linked by a directed edge to at least one other node. The state of a node on the head end of a directed edge is called a *parent node*, which is said to cause the state of the corresponding node at the tail end of the directed edge (a *child node*). Note that no cycles can exist in the graph (acyclic). An example of a BN based on the example in Section 2.2.1 (with an additional event) is shown in Figure 2.5.

It is important that the inter-dependencies among the events in the graph structure properly reflect the real system. Methods exist to ensure this compatibility, which should be used when constructing a complex BN [16]. Once the DAG is created, the conditional probabilities of all nodes must be initialized. Different methods used to initialize these probabilities are discussed in Section 2.2.5. The initial conditional probabilities of each node are usually organized in a conditional probability table (CPT) that lists the probability of a node based on the state(s) of its parent

node(s). If the node has no parent nodes, the CPT represents the unconditional probability of the states of that node. Examples of CPTs for the nodes in Figure 2.5 are shown in Table 2.1.

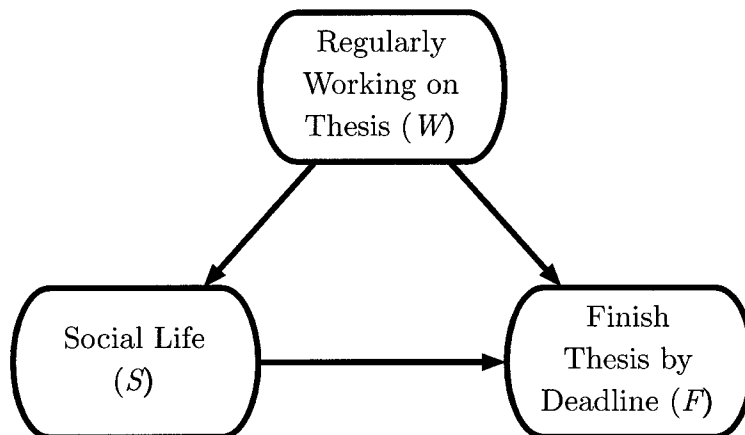


Figure 2.5 – A simple example of a Bayesian network.

Table 2.1 – The conditional probability tables for the nodes in Figure 2.5.

| (a) Node W | | (b) Node S | | | (c) Node F | | | |
|--------------|----------|--------------|-------|------------|--------------|-------|-------|---------------|
| W | $\Pr(W)$ | W | S | $\Pr(S W)$ | W | S | F | $\Pr(F S, W)$ |
| true | 0.85 | true | true | 0.10 | true | true | true | 0.08 |
| false | 0.15 | true | false | 0.90 | true | true | false | 0.92 |
| | | false | true | 0.75 | true | false | true | 0.80 |
| | | false | false | 0.25 | true | false | false | 0.20 |
| | | | | | false | true | true | 0.02 |
| | | | | | false | true | false | 0.98 |
| | | | | | false | false | true | 0.18 |
| | | | | | false | false | false | 0.82 |

Now that the CPTs have been initialized, BT can be used to calculate different conditional probabilities. For example, the probability that a student has a social life

given he/she is regularly working on their thesis is

$$\Pr(S|W) = \frac{\Pr(W|S)\Pr(S)}{\Pr(W)} = \frac{\left(\frac{0.10}{0.10+0.75}\right)(0.10 + 0.75)}{0.85} = 0.118. \quad (2.4)$$

2.2.3 Worlds

A *world* is an instance where each node is a known state. Hence, for the BN in Figure 2.5, there are eight possible worlds, shown in Table 2.2. The sum of probabilities of all worlds must be equal to one as the set of worlds describes all possible outcomes. The probability of an individual world can be calculated as

$$\Pr(w_i) = \Pr(W_{w_i})\Pr(S_{w_i}|W_{w_i})\Pr(F_{w_i}|S_{w_i}, W_{w_i}), \quad (2.5)$$

which is the *chain rule* for BNs. For example, using the conditional probabilities defined in Table 2.1, the probability of w_7 is

$$\begin{aligned} \Pr(w_7) &= \Pr(\bar{W})\Pr(\bar{S}|\bar{W})\Pr(F|\bar{S}, \bar{W}) \\ &= (0.15)(0.25)(0.18) \\ &= 0.0068. \end{aligned} \quad (2.6)$$

2.2.4 World Probability Updates with Evidence

If the state of a node is observed, the probability of all worlds must be updated to reflect this observation, known as *evidence*. For example, if it was observed that a student does not have a social life (\bar{S}), all worlds that contradict this evidence now have zero probability. However, the probability of the remaining worlds must now be adjusted such that the total probability of all worlds remains one. Because this observation does not provide new information about the remaining worlds, the

Table 2.2 – All possible worlds and their probabilities for the BN in Figure 2.5.

| World | W | S | F | Pr(World) |
|-------|-------|-------|-------|-----------|
| w_1 | true | true | true | 0.0068 |
| w_2 | true | true | false | 0.0782 |
| w_3 | true | false | true | 0.6120 |
| w_4 | true | false | false | 0.1530 |
| w_5 | false | true | true | 0.0023 |
| w_6 | false | true | false | 0.1103 |
| w_7 | false | false | true | 0.0068 |
| w_8 | false | false | false | 0.0308 |

scaling factor is the same for each of them. To keep the relative probability among the remaining worlds the same, they are simply normalized. For example, consider the worlds in Table 2.2. With the new evidence (\bar{S}), the probability of w_1 , w_2 , w_5 , and w_6 is zero. Normalizing the remaining non-zero probabilities yields the posterior probability $\text{Pr}'(w_3)$ as

$$\begin{aligned}
 \text{Pr}'(w_3) &= \frac{\text{Pr}(w_3)}{\text{Pr}(w_3) + \text{Pr}(w_4) + \text{Pr}(w_7) + \text{Pr}(w_8)} \\
 &= \frac{0.6120}{0.612 + 0.153 + 0.0068 + 0.0308} \\
 &= 0.191.
 \end{aligned} \tag{2.7}$$

The posterior probability of all worlds after applying Equation 2.7 is shown in Table 2.3. The posterior probability of any state of any event is simply the sum of the probabilities of all worlds where that state occurs. For example, before the evidence was introduced, the overall probability that the student would finish his/her thesis

on time was

$$\begin{aligned}
 \Pr(F) &= \Pr(w_1) + \Pr(w_3) + \Pr(w_5) + \Pr(w_7) \\
 &= 0.0068 + 0.6120 + 0.0023 + 0.0068 \\
 &= 0.6279.
 \end{aligned} \tag{2.8}$$

However, after the evidence is observed (i.e., using Table 2.3), this is updated to $\Pr'(F) = 0.7710$. Thus, the evidence that the student does not have a social life increased the probability that he/she will finish their thesis on time.

Table 2.3 – The updated probabilities of all worlds after the evidence \bar{S} is introduced.

| World | W | S | F | $\Pr'(\text{World})$ |
|-------|-------|-------|-------|----------------------|
| w_1 | true | true | true | 0.0000 |
| w_2 | true | true | false | 0.0000 |
| w_3 | true | false | true | 0.7626 |
| w_4 | true | false | false | 0.1907 |
| w_5 | false | true | true | 0.0000 |
| w_6 | false | true | false | 0.0000 |
| w_7 | false | false | true | 0.0084 |
| w_8 | false | false | false | 0.0383 |

In the above example, evidence directly determined the state of an event. However, it is not uncommon that evidence changes the relative probabilities among the possible states. This is called *soft evidence* and takes the form $\Pr'(A) = z$, where z is the posterior probability of the state of event A . For example, perhaps a professor thought they recognized their student at a bar but was not sure it was them. This led the professor to believe there was a 60 % chance that the student has a social life ($\Pr'(S) = 0.60$). To determine the updated probability that the student will finish his/her thesis on time, the probability of each world must be updated. Much like the previous

example, the posterior probabilities of the world are scaled according to the new evidence, once again maintaining the relative probabilities (this time among two sets of worlds: when the student has a social life and when the student does not have a social life). For $\Pr'(S) = 0.60$, this normalization is

$$\Pr'(w) = \begin{cases} \frac{0.60}{\Pr(S)}\Pr(w), & \text{if } S, \\ \frac{1 - 0.60}{\Pr(\bar{S})}\Pr(w), & \text{if } \bar{S}. \end{cases} \quad (2.9)$$

The posterior probability of all worlds after applying Equation 2.9 is shown in Table 2.4. Using Equation 2.8, the posterior probability that the student will finish his/her thesis on time after entering the soft evidence is $\Pr'(F) = 0.3359$. Thus, the evidence that there is a 60 % chance that the student has a social life greatly reduced the probability that he/she will finish their thesis on time.

Table 2.4 – The updated probabilities of all worlds after the evidence $\Pr'(S) = 0.60$ is introduced.

| World | W | S | F | $\Pr'(\text{World})$ |
|-------|-------|-------|-------|----------------------|
| w_1 | true | true | true | 0.0207 |
| w_2 | true | true | false | 0.2380 |
| w_3 | true | false | true | 0.3050 |
| w_4 | true | false | false | 0.0763 |
| w_5 | false | true | true | 0.0068 |
| w_6 | false | true | false | 0.3350 |
| w_7 | false | false | true | 0.0034 |
| w_8 | false | false | false | 0.0153 |

2.2.5 Initializing and Training

The probabilities in CPTs (e.g., Table 2.1) must be initialized. This is generally done in one of two ways: consulting an expert or training with data sets. An expert uses

his/her knowledge of the field to assign conditional probabilities. Constructing a BN alleviates some of the difficulty in these initializations because only the conditional probabilities of child nodes based on their parent nodes must be specified. For example, consider the BN in Figure 2.5. It is easier to assign $\Pr(F|\bar{S}, W)$ than $\Pr(F)$ because F is conditionally dependent on S and W . It was shown that initializing conditional dependencies can be relatively simple for experts. When initializing a BN used to diagnose diseases based on symptoms, doctors found it relatively simple to specify these conditional probabilities versus specifying the unconditional probability that a patient has a disease [17].

Initializing CPTs with data sets is useful if the sample size is sufficiently large and representative of the total population. Suppose students who had completed their theses were polled, resulting in the data set shown in Table 2.5. Calculating probabilities for the CPTs using BT is relatively straightforward. For example, the probability that a student has a social life if they work on their thesis regularly is

$$\Pr(S|W) = \frac{\Pr(W|S)\Pr(S)}{\Pr(W)} = \frac{(\frac{4}{7})(\frac{7}{15})}{\frac{10}{15}} = 0.4, \quad (2.10)$$

or the probability that a student will finish his/her thesis on time if they work on it regularly and do not have a social life is

$$\Pr(F|W, \bar{S}) = \frac{\Pr(W, \bar{S}|F)\Pr(F)}{\Pr(W, \bar{S})} = \frac{(\frac{5}{7})(\frac{7}{15})}{\frac{6}{15}} = 0.833. \quad (2.11)$$

2.3 Fuzzy Logic

Probabilistic classification using BNs using the methods described in Section 2.2 often requires converting a continuous measurement into a discrete observation. For example, a training set was used to train a BN in Section 2.2.5. This data set

Table 2.5 – An example of a data set generated by polling 15 students who had completed their theses.

| Student | <i>W</i> | <i>S</i> | <i>F</i> |
|---------|----------|----------|----------|
| 1 | true | false | true |
| 2 | true | true | false |
| 3 | true | false | true |
| 4 | true | false | true |
| 5 | false | true | false |
| 6 | true | true | false |
| 7 | false | false | false |
| 8 | false | true | false |
| 9 | true | false | true |
| 10 | true | true | true |
| 11 | true | false | true |
| 12 | true | true | false |
| 13 | false | true | false |
| 14 | false | false | true |
| 15 | true | false | false |

consisted of the results of a survey of recently graduated students that asked the following questions:

- (i) Were you regularly working on your thesis?
- (ii) Did you have a social life?
- (iii) Did you finish your thesis by the deadline?

Realistically, students taking this survey would require more detail to properly answer the first two questions. For example, how often and how long does one have to work on his or her thesis to answer the first question affirmatively? Also, what are the conditions on whether or not a student has a social life? A better way to perform this survey would be to provide criteria that a student can use to determine how they

should answer the question. For example, a criterion could be “a student has a social life if he or she spends more than six hours per week undergoing social activities”. In addition, more information could be gained if partial truths were allowed (i.e., students are allowed to answer “75 % yes”). This use of partial truths to explain membership to the state of a variable is the basis of fuzzy logic.

The goal of fuzzy logic is to logically assign membership of a continuous measurement to a state of a variable, and to indicate the quality of the membership (the degree of truth). Fuzzy set theory originated in 1965 in a proposal by Lotfi Zadeh [18]. Unlike traditional binary logic where variables have only two states (e.g., true or false), a *truth value* describes the degree to which a measurement belongs to a state. Truth values range from zero to one, where zero means “does not belong” to a state, and one means “has full membership” to a state. The result is the discretization of a continuous measurement into degrees of truth pertaining to a set number of possible states.

2.3.1 Membership Functions

The membership function (MF) of a state maps a continuous measurement into a truth value for that state. For example, if a student spent six hours per week socializing, a MF for the state “true” of the variable “has a social life” (S) may produce a truth value of 75 %. In this example, the state was a binary variable, but this is not a requirement to use fuzzy logic. If the state was expanded to include more detail about the extent of a student’s social life (e.g., $S \in \{\text{hermit, average, party animal}\}$), MFs for each state would convert the average social hours per week into truth values for each state. For example, the outcome of this transformation for a student that socializes five hours per week could be $\{0.83, 0.17, 0\}$, which indicates that a student is 83 % a hermit and 17 % average.

The only restriction on the form of a MF is that it maps an input to an output

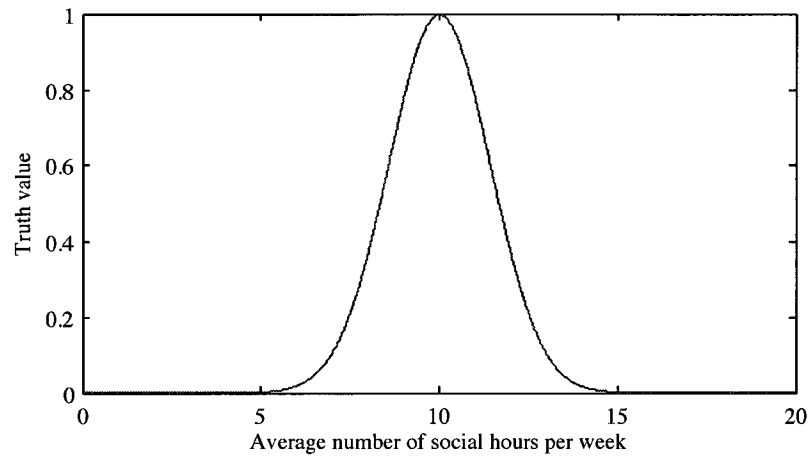
within the interval $[0, 1]$. As a result, many different types of MFs have been investigated. Two common approaches are Gaussian or linear functions. Gaussian MFs are useful if there is a singular “best” value for a state and deviation from that value can be reasonably assumed to be a Gaussian random variable. Linear functions are useful when there is a range of values that describe absolute membership and/or absolute non-membership to a state. They are also relatively simple to implement and train. Examples of a Gaussian and linear MFs are shown in Figure 2.6. This thesis uses fuzzy logic to discretize measurements of the states of different physical attributes. As there is no singular “correct” value for many of these states, but rather a range of acceptable values, a Gaussian approach is impractical. For this reason, along with their simplicity, linear MFs are used in this thesis.

The truth value of a measurement can be derived from a linear piecewise-continuous MF as

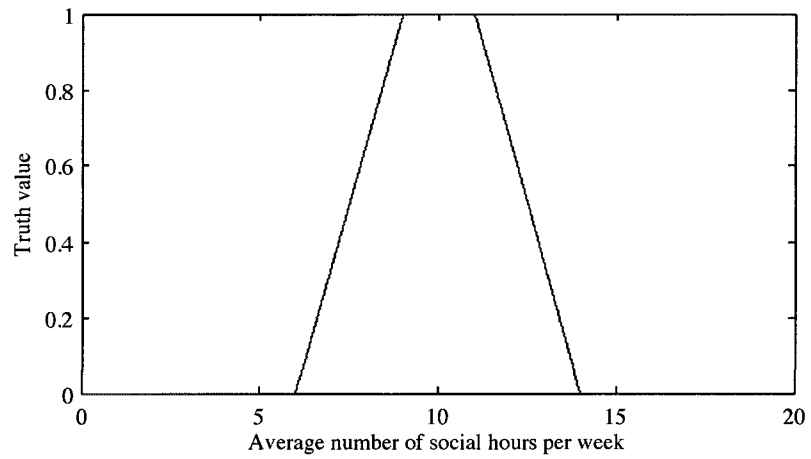
$$\text{Truth Value} = \begin{cases} 0 & \text{if } x \leq l_f, \\ \frac{x-l_f}{l_s-l_f} & \text{if } l_f < x \leq l_s, \\ 1 & \text{if } l_s < x \leq r_s, \\ \frac{r_f-x}{r_f-r_s} & \text{if } r_s < x \leq r_f, \\ 0 & \text{if } x > r_f, \end{cases} \quad (2.12)$$

where x is the measurement and the parameters $\{l_f, l_s, r_s, r_f\}$ are the values of x that make up the left foot, left shoulder, right shoulder and right foot of a trapezoid, respectively. Note that when $l_s = r_s$, the function describes a triangle. The MF shown in Figure 2.6(b) can be described as $\{6, 9, 11, 14\}$.

A set of MFs that describes all possible states of a measurement is *sum normal* if the sum of all truth values for any measurement is equal to one. This is a frequently used but not an essential attribute of fuzzy logic. Sum normality is often desirable due to its aesthetic and intuitive appeal. Also, it can improve algorithmic simplicity



(a) Gaussian.



(b) Linear.

Figure 2.6 – Examples of two types of fuzzy membership functions.

due to the simple reason that it imposes a new constraint on the system. However, sum normality is not enforced in this thesis. The reason for this is a direct result of using fuzzy logic to soften Bayesian evidence. For example, consider the set of MFs that are not sum normal in Figure 2.7. Suppose these MFs were developed based on a training set (as described below in Section 2.3.2). These functions assign a size to a rock based on its width. Now suppose a BN attempts to classify rocks based on their size. The MFs in Figure 2.7 suggest there are three distinct ranges of rock sizes. By allowing the functions to not be sum normal, anomalies (such as measurements directly between states or larger than the largest state) can be properly classified by the BN. For example, if a rock was measured to be 70 cm wide, a sum normal set may classify this as “large”, while the non-sum normal set would indicate that it does not belong to any state. This fact (no state membership) would be propagated through the BN and would help classify the rock as “unknown”, which is desirable behaviour if anomalous rocks are to be detected. Another approach to ensure anomalous rocks are labelled as “unknown” is Dempster-Shafer theory [19]. This theory deals with belief measurements rather than probability, and makes concessions for beliefs such as “unknown”. However, Dempster-Shafer theory is more computationally and algorithmically complex than traditional Bayesian probability [20]. For these reasons, the theory is not pursued in this thesis; however, future investigation into its utility in autonomous science is recommended.

2.3.2 Initializing and Optimizing Membership Functions

The MFs that are selected to describe the states of a variable can greatly affect the quality of the outputs. In this thesis, the MFs map continuous measurements into truth values that are used as soft evidence for a BN. Additionally, the objects that the BN is attempting to identify must be accurately describable by the MFs. For example, if one type of object consistently has a width of 6-8 cm or greater than 20 cm, then

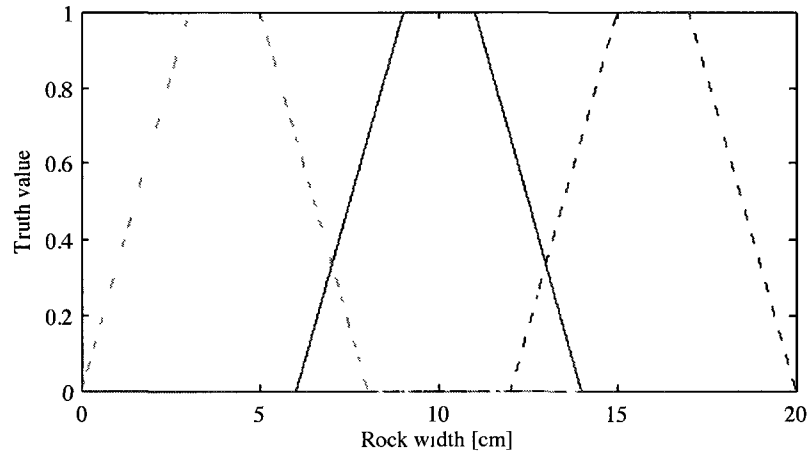


Figure 2.7 – A set of fuzzy membership functions that are not sum normal. The size of a rock is described by the states “small”, “medium”, or “large” (dashed, solid, and dash-dotted lines, respectively).

the MFs in Figure 2.7 would not be a good choice. To ensure compatibility, MFs can be initialized with a data set of the measurements that it will eventually describe. For instance, suppose a data set of 100 rocks were measured and their widths were put into a histogram, as shown in Figure 2.8. From this histogram, one can approximate reasonable MFs by ensuring high density groups of widths are given their own state. An example of a possible set of MFs for the data set is shown in Figure 2.9. These MFs do not form a sum normal set. Therefore, any abnormally large rocks or rocks that fall in between sizes described by the MFs can be labelled as anomalies by the BN.

The MFs of fuzzy logic systems can be optimized after initialization based on the performance of the system. There are many techniques for optimization, including genetic algorithms, neural networks, evolutionary programming, geometric methods, fuzzy equivalence relations, heuristic methods, gradient descent, and Kalman filtering [21]. Some of these methods assume or ensure sum normal MFs. In general, these techniques use either the output of the fuzzy system or the derivative of the output with respect to the MF parameters (or both) to optimize the MF parameters.

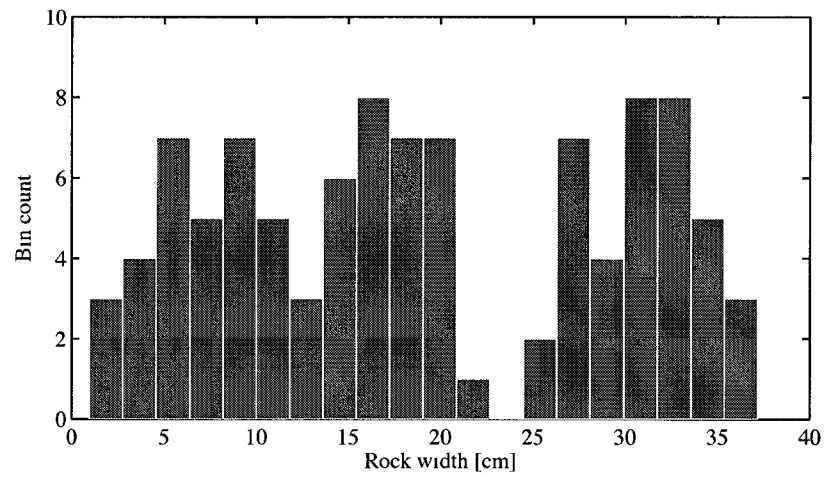


Figure 2.8 – A data set of measurements used to initialize the membership functions of the variable “size”.

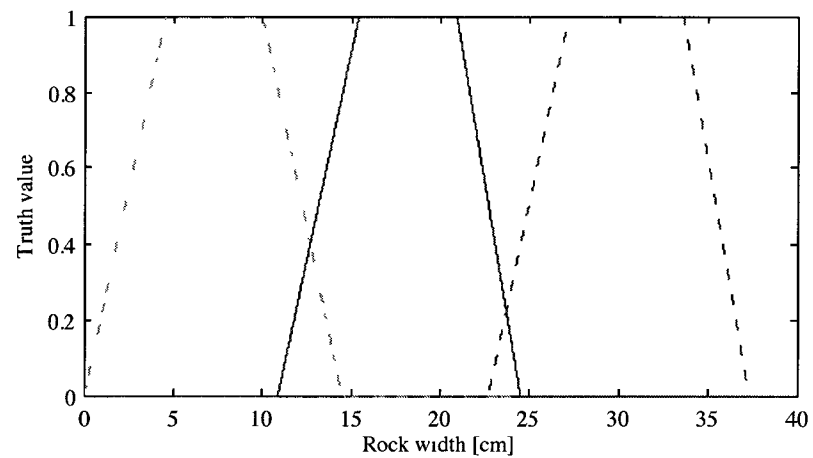


Figure 2.9 – Membership functions for the states “small”, “medium”, and “large” (dashed, solid, and dash-dotted lines, respectively) approximated from a data set.

However, these techniques are not employed in this thesis—the MF parameters were selected manually based on training data.

2.4 Other Topics

The following topics are necessary in the development of the autonomous science system; however, they are not part of focus of this thesis and will not be reviewed in detail here. Their role in the system is described in Chapter 3.

2.4.1 Machine Vision

Recently developed autonomous science systems have universally used machine vision in the visual spectrum for feature detection and/or attribute measurement (see Section 2.5). Machine vision is widely used partly because cameras (especially stereo cameras) are widely used on planetary rover platforms. In addition, image processing algorithms can provide an array of measurements that can be used to identify features. Finally, stereo vision can provide 3D position estimates of observed features through stereo triangulation. Other methods (e.g., infrared measurements of thermal capacitance) can be used in conjunction with machine vision to improve the diversity of measurements.

Appropriate machine vision algorithms were made available from the output of collaborative work within the author’s research group. The algorithms provide four functions: image segmentation, measurement of structural and textural attributes, correlation of features, and position estimation of features using stereo triangulation. Image segmentation applies K-means clustering [22] using a moving 200×200 pixel window. The measured structural attributes are extracted from pixel quantities and the distribution of features, and the measured textural attributes are a subset of the Haralick Textural Parameters [23]. The measurements used in this thesis are

described below.

Size

The projected area (cm^2) of a feature (the pixel count transformed using stereo vision).

Intensity

The average grayscale pixel colour of a feature.

Contrast

A measure of the sharpness of variations in the texture of a feature. High contrast is evidence of sudden, crisp changes in texture.

Energy

The uniformity of the texture of a feature. Low energy indicates an erratic texture.

Entropy

The complexity of the texture of a feature. High entropy indicates a complex texture.

Inverse Difference Moment (IDM)

A measure of the homogeneity of the texture of a feature. High IDM indicates a homogeneous texture.

It should be noted that grayscale images were used throughout the experiments for the reasons stated in Section 4.2.2.

The correlation of features uses both the structural and textural parameters on segmented regions to specify which regions are to be considered features. The estimated range and bearing to each feature are extracted via stereo triangulation, which requires the rectification of the left and right images and the measurement of the displacement of the features between the two images (the disparity map).

2.4.2 State Estimation with the Extended Kalman Filter

State estimation is an important topic in mobile robotics. In the scope of this thesis, the *state* is referring to the position of a feature, and not the pose of the rover (which is assumed to be known, as described in Section 1.2). In general, when measuring the state of a feature, its state cannot be observed directly (i.e., the observation will be subject to the capabilities of the sensor measuring it, as well as the position of the rover that is employing this sensor). Also requiring consideration is the fact that sensors are inherently noisy, and the effect of this noise needs to be factored into the confidence of measurements. A popular method used to update the state of a feature after a measurement and to propagate its uncertainty is the extended Kalman filter, particularly in simultaneous localization and mapping [2]. This thesis uses a similar method to update the state of the features, as described in Section 3.2.

2.4.3 Velocity-constrained Control

There are three basic types of control for wheeled vehicles: point stabilization, trajectory tracking and path following. Point stabilization control attempts to minimize the error between the current pose of the vehicle and a specified final pose, but does not specify the path the vehicle must take between the two poses. Trajectory tracking control attempts to minimize the pose of the vehicle relative to a “virtual” vehicle that is executing the desired path (i.e., the error at each time step is the difference in pose of the vehicle and its virtual counterpart at that time). Finally, path following control attempts to minimize the lateral and heading error between the vehicle and a defined path, independent of time. The control type used in this thesis is a modified version of path following control.

Path following control usually only affords the vehicle one degree of freedom—the angular velocity of the vehicle. The forward velocity is usually specified as constant,

while the angular velocity is controlled to minimize the lateral and heading error relative to the path. However, this approach can be undesirable when the vehicle must undergo turns with large turning radii. Also, vehicles that have the capability to turn “on the spot” (near-zero turning radius), such as the vehicle simulated in this thesis (see Section 4.1 for the vehicle model), lose this capability if the controller requires a constant forward velocity. In addition, velocity constraints (either due to the geometry of the vehicle or imposed by the operator) must not be violated in any control situation. To this end, the controller used in this thesis is the path following, velocity-constrained control developed by Bak *et al.* [24].

This controller has two key properties that make it ideal for the rover model in this thesis executing Theta*-planned paths. First, it allows for the specification of several types of velocity constraints, namely: minimum and maximum vehicle forward velocities, minimum and maximum vehicle angular velocities, minimum and maximum wheel angular velocities, and a desired vehicle forward velocity. The controller attempts to minimize the lateral and heading error from the path while maintaining the desired forward velocity but not violating any of the velocity constraints. Second, because the path is known, the controller can anticipate upcoming turns and adjust the velocities such that desirable turn properties (overshoot, settling time, etc.) are achieved, once again without violating velocity constraints. The combination of these two properties allows for smooth, constrained execution of the planned paths.

2.5 Literature Review on Autonomous Science

Autonomous science systems would greatly benefit existing and future planetary rovers for the reasons discussed in Chapter 1. These systems are often an amalgamation of many different algorithms. In general, the goal of autonomous science is to extract information from observations of a rover’s surroundings and use that

information to make intelligent decisions. Many different approaches to autonomous science have been developed to achieve this goal. However, most systems to date share the following underlying sequence of algorithms:

- (i) Capture an image of the region to be evaluated.
- (ii) Identify rocks in the region using segmentation and detection image processing (and sometimes machine learning) algorithms.
- (iii) Extract physical attributes of the identified rocks using classification image processing (and sometimes machine learning) algorithms.
- (iv) Evaluate or rank the rocks based on their estimated physical attributes.
- (v) (If applicable) Re-plan list of tasks being executed by the rover if a sufficiently interesting rock is found.

Although efforts to date have generally followed this sequence, there is a lot of room for diversity. This includes the type of image processing used for rock identification, methods used for classification (e.g., whether or not machine learning techniques are used, and if they are, how they are used), the system used to evaluate and rank rocks, and how all of this information is used to make intelligent decisions. The review below focuses on the last three major efforts in developing an autonomous science system for planetary rovers. Previous work that contributed to the described systems, as well as the system presented in this thesis, is described in Section 2.5.4.

2.5.1 Thompson, Niekum, Smith and Wettergreen [1]

A Bayesian approach to autonomous science was developed by Thompson *et al.* Their work was performed alongside NASA's Jet Propulsion Laboratory (JPL), eventually contributing to the Autonomous Exploration for Gathering Increased Science System

(AEGIS) project. A version of AEGIS was uploaded to the Mars rover Opportunity rover in 2010 [25]. The authors recognized the necessity of increased autonomy for planetary rovers due to the increase in range and data collection surpassing the available data transfer bandwidth. Their work focussed on detecting and classifying the geological attributes of Mars analogue terrain, with particular emphasis on detection. This was done primarily using image processing, although the possibility of expanding the algorithm with spectroscopy or other sensors was noted. Their algorithm can qualitatively be divided into three sequential parts: segmentation, detection and classification. The goal of the system was to simply gather and classify data (i.e., the processed data did not affect the behaviour of the rover). Below is a description of the parts of the system followed by conclusions that were drawn through field testing.

Segmentation

The purpose of the segmentation algorithm is to isolate potential scientific areas in captured images. It does not isolate individual targets (detection), but simply separates the image into distinct homogenous regions. This was performed using an iterative “shatter and unite” method. Before segmenting the image, a simple Gaussian blur operation is applied to each colour channel. The “shatter” step decomposes the image in a grid of 5×5 pixel regions. The mean pixel value of each region is calculated and regions whose value falls within the threshold of a neighbouring region are merged (the “unite” step). The image continues to be shattered and united until the unification of neighbouring regions ceases to occur. Afterwards, regions outside some defined window size are filtered out (e.g., the sky or benign terrain) leaving only segmented regions that may contain a rock (or other features sufficiently different from the background).

Detection

The regions of the image that were highlighted during segmentation are then assessed for their validity as a science target. Detection extracts an attribute vector from each segmented region, and using a BN, labels it as one of five possible classes: rocks, uniform patch of soil, sky, shadows, or “everything else”. A BN was selected due to its ability to handle missing data and its computational efficiency compared to non-parametric classifiers. The nodes of the BN for detection included: perimeter (the ratio of a region’s perimeter to its pixel area), relative colour (the difference in mean pixel hue, saturation and intensity between a region and the background), relative colour variance (the difference in pixel variance between a region and the background), height (above the ground plane), texture (the surface roughness of a region calculated using a fractal dimension measure of a binary intensity map), and the intensity gradient (the magnitude of the intensity gradient over the pixel surface of a region).

Classification

Once regions were assigned to one of the five classes via detection, geological classification is performed. In particular, the regions that were detected as rocks are classified geologically with a BN. The nodes of this BN are similar to those used for detection; however, as the properties of a target are generally not related to their differences from the background, nodes that exercise this comparison are omitted. In addition, the absolute intensity, saturation and hue of the rock are added to the attribute vector. The geological classifier evaluates each rock and assigns it a class that can be either predefined or autonomously generated. The predefined geological types are selected by an expert, while the autonomously generated types depend on the data. Essentially, the autonomous method creates its own geological types

by classifying rocks with similar attributes together into groups. This method has the advantage of autonomously identifying outliers (i.e., rocks whose attributes are notably different from their neighbours) or selecting the best representative rock of a class. However, allowing the autonomous creation of classes hinders human interpretation by not classifying rocks in categories familiar to experts. To improve the compatibility of the autonomous science algorithm with planetary scientists, a qualitative study was performed which employed the use of a science priority language [26].

Testing and Results

Field testing was performed in the Atacama desert in Chile due to its resemblance to Mars terrain. The “Zoe” test rover from the Carnegie Mellon Robotics Institute was fitted with a 30 cm baseline stereo camera on a pan-tilt unit two metres above the ground, which could capture 1280×960 full colour images. The primary goal of the tests was to evaluate the segmentation and detection algorithms. It was found that small rocks and rocks in the far-field were much more difficult to detect (rocks over 75 pixels in size had a median detection rate of 100 %). It was suggested that expanding the attribute set to include more sophisticated textures and angularity (sharpness) would help improve the detection algorithm. Additionally, it was noted that boosting the minimum pixel size of potential science targets would greatly improve accuracy. Also, including stereo disparity data in the segmentation process was speculated to possibly improve merging appropriate regions together.

2.5.2 R. Castano, Estlin, Anderson, Gaines, A. Castano, Bornstein, Chouinard and Judd [27]

There has been substantial effort by R. Castano *et al.* at JPL in developing an autonomous science system for future rovers. This exertion has resulted in the On-board Autonomous Science Investigation System (OASIS), used to evaluate and autonomously act upon science data gathered by planetary rovers. Although extendable to other sensors and other target types, the primary focus of OASIS is analysing gray-scale images from monoscopic cameras, identifying rocks of interest, retasking the rover to take additional images of a selected target, and continuing with the original mission. The algorithm has been tested in simulation and in the field. The sections below describe how science data is segmented, extracted and evaluated from the images, and incorporated into the mission plan of the rover. The results and conclusions from field testing are also discussed.

Segmentation and Detection

Single grayscale images are used in the segmentation and detection processes. It was acknowledged that there is pertinent value in using stereo images; however, using monoscopic imagery allows for the use of a larger number of cameras (e.g., the Mars Exploration Rovers have several cameras, many of which are not stereo-cameras). Two segmentation modules are used to segregate the image into homogenous regions: a sky detector and a rock detector. The sky detector finds the horizon with an edge detector, then grows homogeneous regions above the horizon (shatter and unite) until the sky is fully described. The second segmentation module ignores the sky and begins by normalizing the image and filtering it with an edge preserving smoother. Edges are enhanced using unmask sharpening and then detected with both Sobel and Canny edge detectors. Closed shapes from these detected edges are found using an

edgewalker, and then the results from both detectors are combined to produce a list of potential rocks. The algorithm was tested on 65 MER images, resulting in 92 % of regions identified as rocks correctly being rocks. The algorithm was found to have a low false alarm rate but a high miss rate.

Extraction

Using OASIS, information can be extracted from three types of features: clouds, dust devils and rocks. This review focuses on the extraction of scientific data from rocks. The rock extraction algorithm estimates four properties: albedo, texture, size and shape. Albedo is measured by computing the average gray-scale value of the pixels describing a rock. This measurement gives information about the mineralogical composition of the rock, but can be affected by shadows and the sun angle. Visual texture is estimated by applying a Gabor filter, which can also help determine mineralogical composition as well as geological history. The size and shape parameters are measured by modelling an ellipse (if no range data is available) or an ellipsoid (with range data) on the shape of a rock. The eccentricity of the ellipse (or ellipsoid) as well as the coverage error can indicate the angularity and ruggedness of the rock, a helpful indicator of its past environmental conditions. In addition to extracting science information from individual rocks, it was noted that identifying geologic boundaries and transitions of a larger region would also be valuable to help describe the context of particular features. However, this type of extraction was not implemented to date.

Science Evaluation

Ranking rocks based on extracted science data is performed in three ways: attribute-based, example-based or novelty-based evaluation. Attribute-based evaluation allows scientists to directly set the importance of different attributes, and rocks are ranked

on the extent that these attributes appear. For example, if elongated rocks are desirable, features with highly eccentric ellipses are given higher priority. Example-based evaluation is performed by selecting a rock with interesting properties from the extracted set and ranking rocks based on their resemblance to the selected rock. Finally, novelty-based evaluation is used to identify rocks with outlying features (either globally or in a defined region). Methods used to determine the novelty of a rock include k -means clustering, a Gaussian mixture model, and feature clouds. Feature clouds are D -dimensional plots (D is the number of attributes extracted from each rock) of all the rocks in a region, where rocks falling outside a specified cloud boundary are considered novel.

Re-planning

The re-planner describes how the rover will behave in the event that it detects a novel rock during a traverse (a “science alert”). A basic reaction is simply the rover halting execution of its current plan and waiting for operator input. More advanced reactions are autonomously collecting more data of the novel feature (e.g., additional images) or altering the path of the rover to get closer for additional measurements. After a science alert and a new science goal is formed, it is passed to the re-planner to see if it can be accommodated. The re-planner must consider a number of things when attempting to form a new science-integrated plan. This includes resource (such as power) and temporal constraints, as well as feasibility from a navigation standpoint. OASIS uses the previously developed Continuous Activity Scheduling, Planning, Execution and Re-planning (CASPER) planner to handle this task. Plans are assigned optimization scores based on the science value and constraints, and the plan with the highest optimization score is executed.

Testing and Results

Testing was performed at JPL's Mars yard with the Field Integrated Design and Operations (FIDO) rover (all-steering, all-driven, six-wheeled rover). The goal of the test was to detect rocks with relatively high albedo (white in colour) that were scattered among a variety of low albedo (dark) rocks, using only single grayscale images. OASIS was paired with navigation, locomotion and stereo-vision modules for traversal on the terrain. Ten runs were performed, each with a different rock density and number of white rocks. Of the 40 white rocks that entered the vision of the rover over the ten tests, 36 were detected with no false positives. It was noted that the likelihood of positively detecting a high albedo rock increased substantially if it appeared in multiple images. Also, the difficulty in matching observed rocks with those that were previously seen (data association) was slated as a necessary improvement in future work.

2.5.3 Woods, Shaw, Barnes, Price, Long and Pullan [28, 29]

The autonomous science concept developed by Woods *et al.* recognizes that it would be favourable that a rover emulated a geologist when exploring previously unseen terrain, specifically due to the bottleneck in bandwidth between a rover and its operators. The overall objective was demonstrating that a rover could travel to a target destination and detect and/or respond to scientifically interesting targets it identified along the way. If a geologist were to perform this task, he or she would select the most promising areas visually and then iteratively focus on smaller areas until the best feature is identified based on its physical attributes. The authors attempted to emulate this behaviour by capturing wide-angle images, identifying a part of the image with potential for high scientific value, capturing a high-resolution image of the selected area, and finally commanding the placement of a robotic arm on the most

promising target. This method recognizes the importance of iteration in observations, as closer inspection of targets are likely to change their assessment. This review will focus on the methods the authors used for scientific assessment of features and not the robotic arm placement.

Segmentation and Detection

The segmentation and detection algorithms were not the focus of the study and image processing algorithms from previous work were employed. In general, the image was first smoothed using a morphological and Gaussian smoothing algorithm. A “shatter and unite” segmentation algorithm was used to distinguish rocks from the surrounding soil. The results from this algorithm were combined with a double differential detection algorithm applied to gray-scaled versions of the original images to enhance the detection of potential targets, i.e.,

$$I_{i,j}^F = f\left(\frac{d^2z}{dx^2}, \frac{d^2z}{dy^2}\right) + f\left(\frac{d^2z}{dxy}, \frac{d^2z}{dyx}\right). \quad (2.13)$$

Scientific Evaluation

Although the value using machine learning techniques for geological classification was noted (and even suggested as future work), scientific evaluation was performed by assigning value to attributes rather than minerals. For example, the extent of bedding on a target was assigned a score rather than estimating the type of rock based on its attributes, and assigning scores based on the rock types. The “basic ingredients” for interpretation were defined to be the structure, texture and composition of a rock. Although many criteria could be used to qualitatively describe these ingredients, this study focused on observing particular attributes: the type and extent of bedding for structure, the presence and density of granules for texture, and albedo and reflectivity for composition. Scores are assigned based on the value given to attributes by experts

as.

$$SVS = \left(\sum As + \sum At + \sum Ac + Ax \right) \cdot Q \cdot B. \quad (2.14)$$

The science value score (SVS) is calculated based on the following: As , At , and Ac are the overall structural, textural and compositional attribute scores, Ax is the composite score (applied when combinations of particular attributes are observed), Q is the quality factor that adjusts the score based on the quality of the measurements, illumination, etc., and finally B is the bias score that adjusts the score based on the context of the observations (e.g., a large B might occur if the rock is an outlier compared to its neighbours). For example, the presence of extensive large bedding on a rock surrounding by dissimilar rocks may be assigned a high structural (As) and bias (B) scores, resulting in a high SVS .

Re-planning

A sufficiently large SVS should autonomously influence the behaviour of the rover to ensure no important science discoveries are overlooked. The authors discussed incorporating science data on the software level of the rover's planning agent. In particular, a study was performed on how science discoveries could be inserted into the European Space Agency's (ESA) software package Time Line Validation and Control (TVCR). An example scenario was presented to discuss how the re-planner might handle possible outcomes of a mission. For example, if the rover was given a sub-goal and three waypoints along the way to look for science candidates, possible outcomes and how the re-planner should handle them are as follows:

- A minor discovery is made at a waypoint. The rover captures and stores the data and continues to the next waypoint.
- A major discovery is made at the final waypoint. The rover stops and waits for

further instructions.

- A major discovery is made at the first or second waypoint. The rover continues to the next waypoint(s) and then travels to and waits at the waypoint with the highest SVS.

Testing and Results

A “half-scale” rover equipped with a wide-angle camera was tested in an indoor Mars analogue environment (sand with some rock coverage) to assess the effectiveness of the scoring system and the re-planner. The rover was given a final goal and three waypoints at which it was to capture wide-angle images (much like the scenario described in Section 2.5.3). No obstacles were placed between the rover and the goal (i.e., the rover drove in a straight line and took images of rocks on its left). The rocks were placed such that low SVS specimens were installed at the first two waypoints and high SVS specimens were installed at the third waypoint. In the tests, the rover correctly assigned low scores to the rocks at the first two waypoints, and the re-planner correctly requested a higher-resolution image of the rocks at the third waypoint. Much of the remaining results dealt with the accuracy of placing a robotic arm on the most desirable target at the third waypoint (not the focus of this review). Overall, it was found that the lighting direction and intensity can greatly affect the uncertainty of attributes extracted from vision algorithms. It was suggested that geological identification and classification could be greatly improved using disparity information from the stereo camera, which would allow for easier extraction of granule sizes and variations in depth of bedding structure. It was concluded that further developments of a similar autonomous science system can and should be developed for a mission such as ExoMars.

Later Work

The work described above has recently evolved and is now known as the Autonomous Science Target Identification and Acquisition (ASTIA) architecture [30]. In this architecture, science target identification is improved through a new knowledge-based approach, which uses rule-based fuzzy logic to classify structure, texture and composition.

2.5.4 Other Work in Autonomous Science

Complete autonomous science systems rely on advancements in the many individual algorithms that comprise the full system. As such, the three major efforts described above, as well as the system presented in this thesis, owe much to this previous work.

Machine learning is a valuable aspect of autonomous science. Although not a full autonomous science system, a rule-based expert system was used with spectrometer measurements to successfully identify carbonates in the Marsokhod field tests [31]. A well known application of using Bayesian networks for target classification is the Nomad robot field tests in Antarctica [32]. The purpose of these experiments was to classify targets into pre-defined classes such as igneous, metamorphic, sedimentary, extraterrestrial, or “other”. Although only a partial implementation of an autonomous science system, this experiment demonstrated the utility of machine learning for mineral classification.

Arvidson *et al.* [33] identified the effectiveness of combining data from different sensors and different viewpoints in generating an overall map of geological trends that can be used to improve rover-based planning. By using simulated orbital and descent images, the geologic setting and regional context of an area could be established which improved the effectiveness of rovers searching for science targets. Pedersen [34] also recognized the utility of identifying geological context of local areas. The possibility

of augmenting the ability of a Bayesian network classification algorithm by exploiting the fact that clusters of objects may be correlated is discussed. It was found that any prior knowledge of mineral distribution as opposed to the assumption of uniform distribution can greatly improve classification.

Roush *et al.* [35] also recognized the usefulness of machine learning and provided important context on how such algorithms could be used. The use of BNs is contrasted against using expert systems for mineral identification. Several key components of using BNs for this purpose are outlined, including the steps that are required for training a BN for mineral identification, a process which was used in this thesis.

2.5.5 Remarks

None of the systems described in Section 2.5 fully satisfy all of the goals outlined in Section 1.3. Specifically, the only system that employed a machine learning technique that partially satisfied goals (i) and (ii) was the work of Thompson *et al.* (attributes rather than objects were identified); however, this system did not satisfy goals (iii) and (iv) as it did not use the knowledge gained from its BN into its guidance, navigation or control algorithms. The other systems did not use machine learning techniques to identify or classify features. Instead of assigning value to types of rocks (i.e., using the observed attributes to identify the type of rock), the attributes themselves are given value (e.g., finding bedding on a rock is given some value). Some of the systems integrated the discovery of important rocks into existing planners, allowing the rover to autonomously re-plan its current mission. However, little detail is given about mapping identified features, specifically whether these maps are accurate enough to satisfy goals (iii) and (iv).

Chapter 3

Algorithm Development

This chapter provides a detailed overview of the algorithms that were developed in order to achieve the goals outlined in Section 1.3. In Section 3.1, an overall high-level description of the system is presented to explain the autonomous science system that was developed for this thesis as a whole. The subsequent sections provide greater detail for each part of the algorithm. Section 3.2 describes how the position of features was estimated such that a map of identified features could be produced. To provide a graph-based terrain map to the path-planner, Section 3.3 outlines how the feature map is discretized. The methods used to extract attributes and identify features is detailed in Section 3.4, including a description of the Bayesian network (BN) model that was used. Once identified, the features are evaluated and filtered using the methods outlined in Section 3.5.

3.1 Algorithm Overview

The autonomous science system is used in two different scenarios for comparison: prime rover solo, and prime/scout rover pairing. The prime rover solo is assumed to have a stereo camera, the autonomous science software, as well as a sensor/tool that performs up-close observations of features (e.g., a robotic arm that requires the rover

to be relative close to a feature). The scout rover is assumed to have a stereo camera but does not have the tools/sensors to perform up-close observations of features. It gathers data about the features and transmits this information back to the prime rover. The two scenarios use the same core algorithms that are described in this chapter. However, the order and frequency that the algorithms are used differ in the two scenarios. The following sections describe these differences.

3.1.1 Prime Rover Solo

A block diagram describing the prime rover solo scenario is shown in Figure 3.1. In this scenario, the prime rover is travelling to a distant goal (e.g., a crater 300 m away). The rover drives towards this goal autonomously and periodically maps and identifies features within the range of its camera using its machine vision and machine learning algorithms for obstacle avoidance and autonomous science. If a suitable science feature is identified (using the criteria described in Section 3.5), the rover travels to that feature for up-close observation and then continues to its goal. For simplification, this thesis assumes the final goal is relatively close (e.g., 10 m) and the rover visits at most one science feature that has passed through all the filters described in Section 3.5.2. In general, the goal of the rover is the final goal unless a suitable science feature exists or a science feature has already been visited.

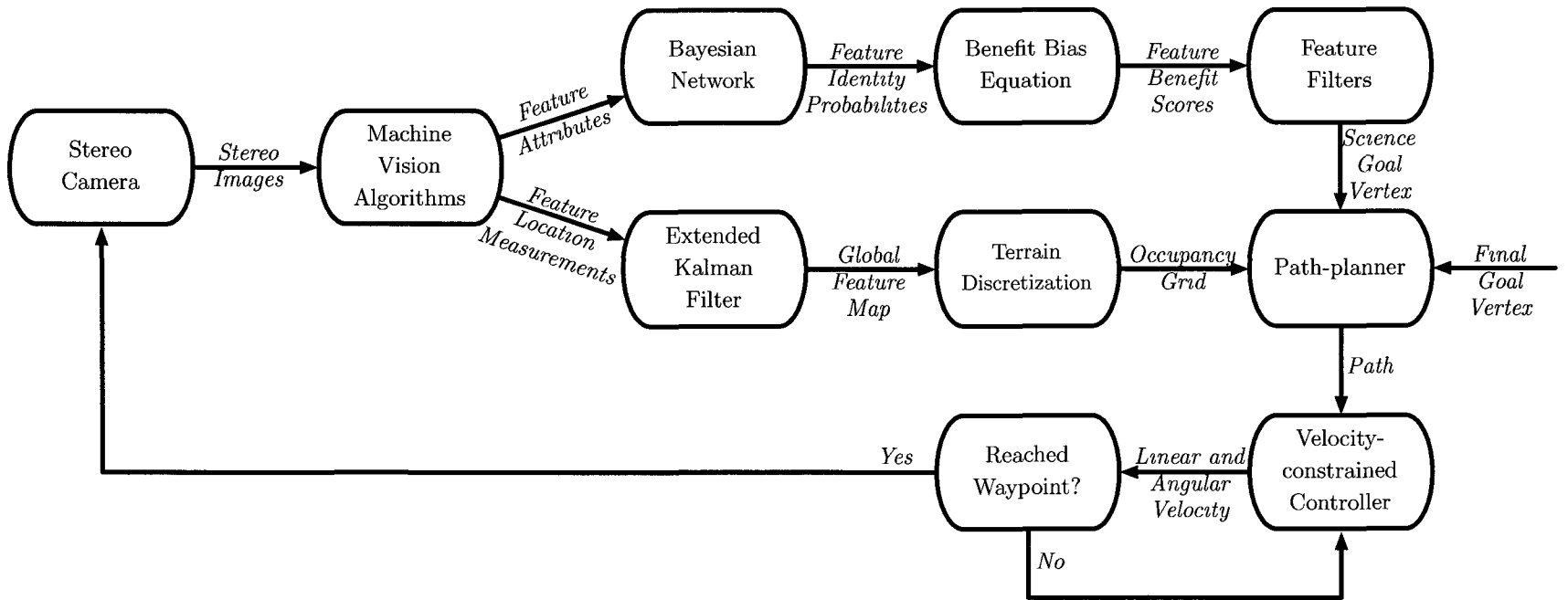


Figure 3.1 – A block diagram of the autonomous science algorithm of the prime rover when no scout is used. The goal vertex of the prime rover is the final goal unless a suitable science goal is found, or a science goal has already been observed up-close.

3.1.2 Prime/scout Rover Pairing

A block diagram describing the prime/scout rover pairing scenario is presented in Figure 3.2. In this scenario, the goals of the rover are the same as the prime rover solo scenario; i.e., travel to a distant goal and visit at most one science feature for up-close observation along the way. However, in this scenario, the goals are achieved differently. The scout rover autonomously travels to the distant goal, periodically mapping and identifying features along the way using its camera and its machine vision and machine learning algorithms. It then sends the global feature map and the science information it extracted from the features back to the prime rover. Effective data transmission between two robotic elements on the Martian surface (the Sojourner rover and the Pathfinder lander) at a rate of 2.4 kbps has been successfully demonstrated [36]. Included in transmitted data is the location of the highest ranking science feature (in terms of the method described in Section 3.5.2). The prime rover then plans a path to that feature and from the feature to the goal. It then executes this path, updating the global feature map as features are re-observed and updating the path if necessary.

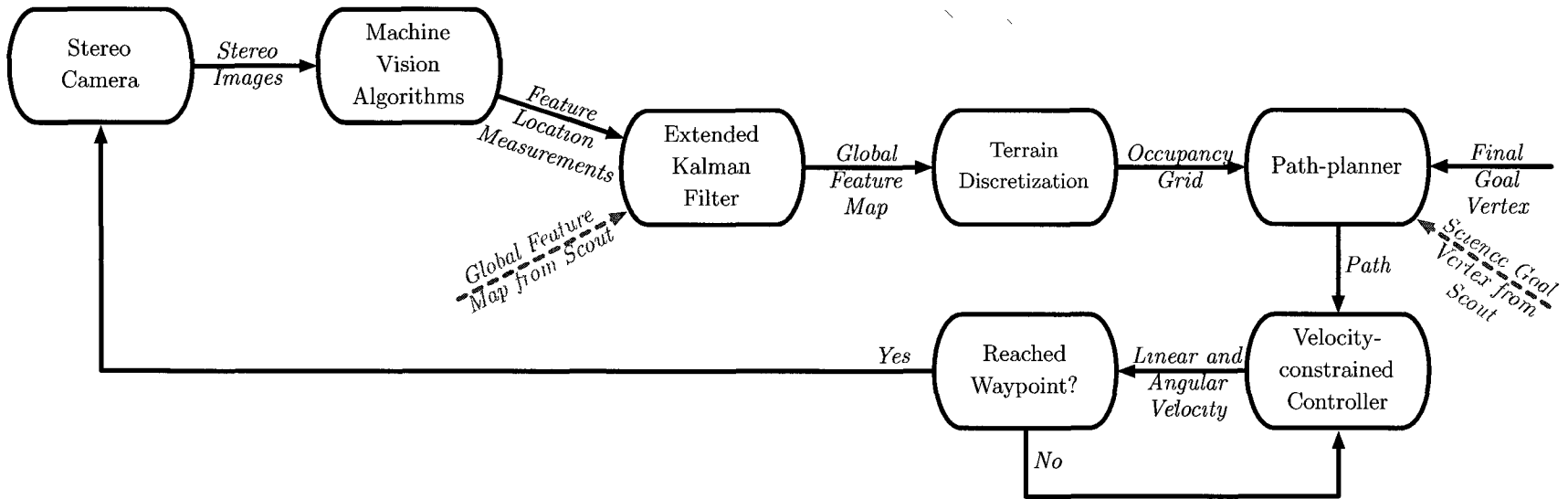


Figure 3.2 – A block diagram of the autonomous science algorithm used by the prime rover when a scout is used. The prime rover receives the global feature map and a science goal from the scout. The goal vertex is the science goal until it has been observed up-close, at which time the final goal becomes the goal vertex.

3.2 Feature Position Estimation

It is important that the rover maintains a map of features that it has observed in global coordinates. This map is important for two key reasons: to ensure the rover can avoid features that present themselves as obstacles, and to be able to find features that present themselves as scientific points of interest. This section details how measurements of the location of features relative to the rover are transformed into a global map, including the propagation of the uncertainty in these measurements.

3.2.1 Feature and Measurement Model

Consider the rover model shown in Figure 3.3. The configuration of the rover is described by its pose $p_k = (X_r, Y_r, \theta_r)$ in the global reference frame (X, Y) . As was discussed in Section 1.2, it is assumed that the pose is always known. Not known, however, are the positions of features within the range of the rover's stereo camera. The state of a feature $q = (X_f, Y_f)$ is estimated by the measurement $y = (d, \phi)$, where d is the unsigned distance between (X_r, Y_r) and (X_f, Y_f) and ϕ the bearing of the feature in the rover reference frame (x, y) . Hence, the model describing the latest estimated position of the stationary feature based on its estimated location at time k is simply

$$q_{k+1} = f(q_k, w_k) = q_k + w_k, \quad (3.1)$$

where the process noise $w_k = \begin{bmatrix} n_x \\ n_y \end{bmatrix} \sim N(0, Q)$. It is assumed that the process noise can be modelled as a Gaussian random variable with covariance $Q = \text{diag}([\sigma_x^2, \sigma_y^2])$. The measurement model is simply derived from the variable definitions in Figure 3.3 as

$$y_k = h(q_k, v_k) = \begin{bmatrix} \sqrt{(X_{f,k} - X_{r,k})^2 + (Y_{f,k} - Y_{r,k})^2} \\ \text{atan2}[(Y_{f,k} - Y_{r,k}), (X_{f,k} - X_{r,k})] - \theta_{r,k} \end{bmatrix} + v_k, \quad (3.2)$$

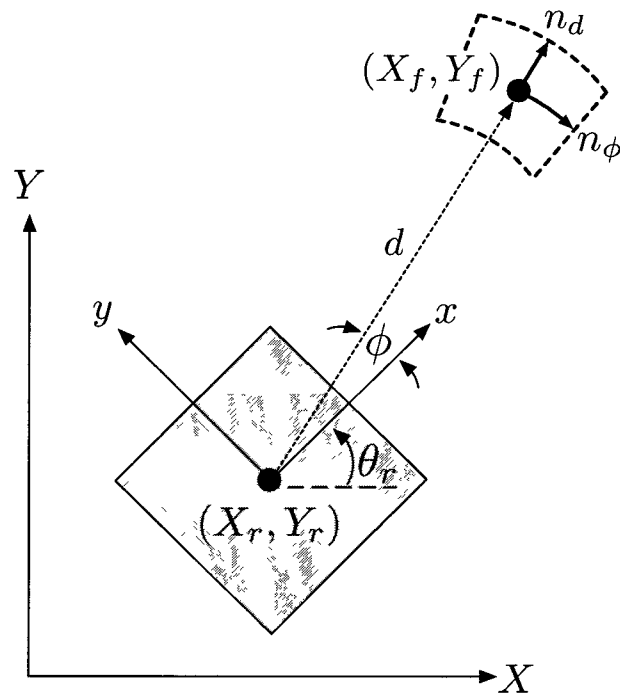


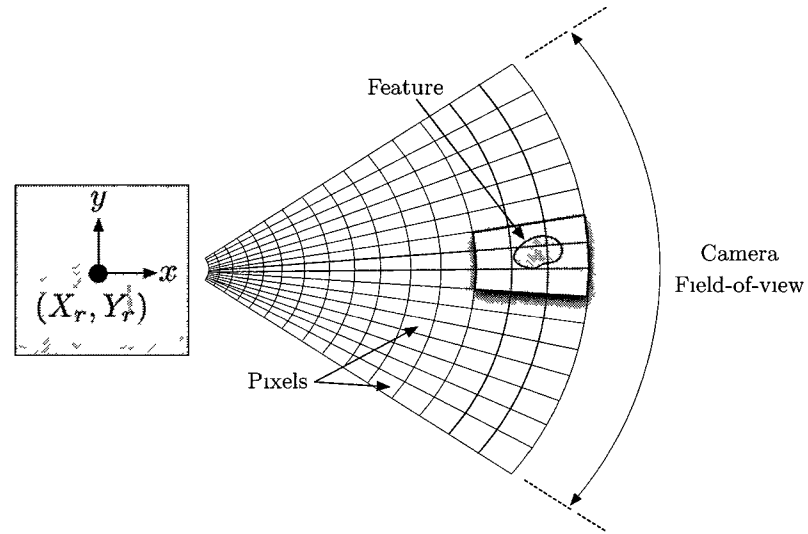
Figure 3.3 – The output of the stereo triangulation algorithm is the estimated distance (d) and angle (ϕ) to a feature in the rover reference frame, and the associated noise in these measurements (n_{range} and n_{angle} , respectively). The pose of the rover and the position of the feature are shown in the global reference frame (X, Y) .

where the function $\text{atan2}(y, x)$ is equivalent to $\tan^{-1}\left(\frac{y}{x}\right)$ except it restricts solutions to be in the range $(-\pi, \pi]$. The measurement noise $v_k = \begin{bmatrix} n_d \\ n_\phi \end{bmatrix} \sim N(0, R_k)$, where $R_k = \text{diag}([\sigma_{r,k}^2, \sigma_{\phi,k}^2])$ and depends on the location of the feature in the captured image. This relationship is described in Figure 3.4. The estimation of range and bearing of a feature is restricted by the pixel size in area of the image where the feature appears. As such, the measurement noise was approximated to be Gaussian but with a dynamic standard deviation. The standard deviation of n_d was taken to be the average radial pixel size around the actual centroid of the feature, and the standard deviation of n_ϕ was taken to be $|\phi_1 - \phi_2|$, where the bearing of the feature's centroid would be ϕ_1 if it was moved to the nearest pixel centre, and ϕ_2 if it was moved to a circumferentially adjacent pixel centre. In general, features closer to the rover have smaller variances.

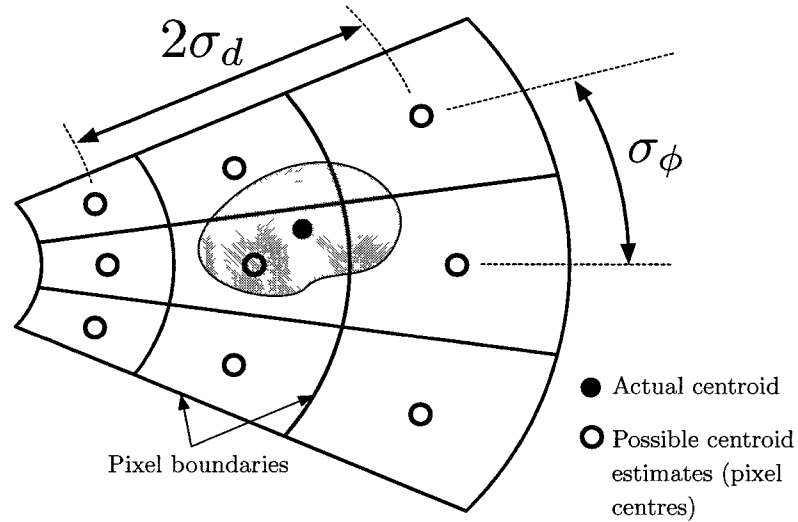
3.2.2 Position Estimation with an EKF

To maintain an up-to-date estimate of the state of a feature (q_k), an EKF is used to update the position estimate each time a new measurement is recorded. This method provides a good way to propagate the uncertainty in the feature states. A general description of the EKF is provided in Section 2.4.2. Note that the feature and measurement models described in Section 3.2.1 allow for some simplification of the EKF because most of the Jacobian matrices¹ are the identity matrix. The Jacobian

¹A Jacobian matrix is the first-order partial derivatives of a vector with respect to another vector.



(a) Pixels projected onto the ground plane expand radially from the rover.



(b) The measurement noise is approximated as a Gaussian random variable, whose variance is a function of the pixel size at the estimated position of the feature.

Figure 3.4 – The approximated measurement noises of camera measurements. The highlighted region in (a) is shown in greater detail in (b).

matrices required for the position estimation updates that are constant are

$$F_k = \left. \frac{\delta f}{\delta q} \right|_{(q_k, 0)} = I_2, \quad (3.3a)$$

$$L_k = \left. \frac{\delta f}{\delta w} \right|_{(q_k, 0)} = I_2, \quad (3.3b)$$

$$M_k = \left. \frac{\delta h}{\delta v} \right|_{(q_k, 0)} = I_2. \quad (3.3c)$$

This greatly simplifies some steps in the EKF, particularly the calculation of the covariance of the state in the *a priori* estimate (Equation 3.4a) and the calculation of the Kalman gain (Equation 3.4b) used in the *a posteriori* estimate,

$$P_{k+1}^- = P_k^+ + Q, \quad (3.4a)$$

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_k)^{-1}, \quad (3.4b)$$

where H is the Jacobian of the measurement model with respect to the state. This matrix is not constant as the measurement model is a function of the state ($y_k = h(q_k, v_k)$). The Jacobian H is

$$H_k = \left. \frac{\delta h}{\delta q} \right|_{(q_k, 0)} = \begin{bmatrix} \frac{X_k - X_r}{\beta} & \frac{Y_k - Y_r}{\beta} \\ \frac{Y_r - Y_k}{\beta^2} & \frac{X_k - X_r}{\beta^2} \end{bmatrix}, \quad (3.5)$$

where $\beta = \sqrt{(X_k - X_r)^2 + (Y_k - Y_r)^2}$, the unsigned distance between the rover and the feature.

The rover maintains a list of features it has observed with their estimated states and covariances. Features found with a new measurement fall into two categories: previously observed and previously unobserved. As discussed in Section 1.2.1, data association is assumed, so previously observed features simply have their estimated

state and covariance updated with the new measurement. Conversely, previously unobserved features must have their state and covariance initialized. The initial state is simply taken as the measurement. The initial covariance is borrowed from a method used in FastSLAM [37]; i.e.,

$$P_k = (H_k^T R_k^{-1} H_k)^{-1}. \quad (3.6)$$

In practice, any diagonal matrix whose entries are positive (a diagonal positive definite matrix) can be used for the initial covariance. The overall update process after each new measurement is shown in Algorithm 4.

3.3 Terrain Discretization and Mapping

As was discussed in Section 2.1, it is necessary to discretize the terrain such that a graph-based path-planner can be used. In this implementation, the map created via discretization is an occupancy grid. An occupancy grid is the simplest version of the cost maps described in Section 2.1.1, where vertices are either occupied (infinite cost) or unoccupied (no cost). Path-planning on this type of map now attempts to minimize the distance travelled rather than minimize the cost of travel. Vertices are occupied if they reside within the area of influence (A_j) of a feature j observed by the stereo camera. The following sections describe how the occupancy grid is populated and how vertices are selected to be scientific points of interest.

3.3.1 Occupancy Grid Mapping

At any given time, the rover maintains a list of features it has observed with their estimated positions and covariances (as described in Section 3.2). The area of influence

Input: Current estimated states and covariances of the n features that have been observed ($\hat{q}_{i,k}$, $P_{i,k}$ for $i = 1, 2, \dots, n$), the process covariance (Q), the latest measurement of the m features currently captured by the camera ($y_{j,k}$ for $j = 1, 2, \dots, m$).

Output: Updated states and covariances of the m measured features ($\hat{q}_{j,k+1}$, $P_{j,k+1}$ for $j = 1, 2, \dots, m$)

```

1 for  $j = 1 \rightarrow m$  do
  // If  $j$  is a previously unobserved feature, initialize its
  // state and covariance.
2  if  $\hat{q}_{j,k} = NULL$  then
    // The initial state is the first measurement transformed
    // to global coordinates.
3     $\hat{q}_{j,k} = \begin{bmatrix} \cos \theta_{r,k} & -\sin \theta_{r,k} \\ \sin \theta_{r,k} & \cos \theta_{r,k} \end{bmatrix} \begin{bmatrix} d \cos \phi_{j,k} \\ d \sin \phi_{j,k} \end{bmatrix} + \begin{bmatrix} X_r \\ Y_r \end{bmatrix};$ 
    // Calculate the Jacobian of the measurement model with
    // respect to the current state.
4     $H_{j,k} = \begin{bmatrix} \frac{\hat{X}_{j,k} - X_r}{\beta} & \frac{\hat{Y}_{j,k} - Y_r}{\beta} \\ \frac{Y_r - \hat{Y}_{j,k}}{\beta^2} & \frac{\hat{X}_{j,k} - X_r}{\beta^2} \end{bmatrix},$ 
    // Calculate the initial covariance.
5     $P_{j,k} = (H_{j,k}^T R_{j,k}^{-1} H_{j,k})^{-1}$ 
6  else
    // Calculate the a priori estimate.
7     $\hat{q}_{j,k+1}^- = \hat{q}_{j,k}^+;$ 
8     $P_{j,k+1}^- = P_{j,k}^+ + Q;$ 
    // Calculate the Jacobian of the measurement model with
    // respect to the a priori state estimate.
9     $H_{j,k+1} = \begin{bmatrix} \frac{\hat{X}_{j,k+1}^- - X_r}{\beta} & \frac{\hat{Y}_{j,k+1}^- - Y_r}{\beta} \\ \frac{Y_r - \hat{Y}_{j,k+1}^-}{\beta^2} & \frac{\hat{X}_{j,k+1}^- - X_r}{\beta^2} \end{bmatrix},$ 
    // Calculate the Kalman gain and the a posteriori estimate.
10    $K_{j,k+1} = P_{j,k+1}^- H_{j,k+1}^T (H_{j,k+1} P_{j,k+1}^- H_{j,k+1}^T + R_{j,k})^{-1};$ 
11    $\hat{q}_{j,k+1} = \hat{q}_{j,k+1}^- + K_{j,k+1} (y_{j,k} - h(\hat{q}_{j,k+1}^-, 0));$ 
12    $P_{j,k+1}^+ = (I_2 - K_{j,k+1} H_{j,k+1}) P_{j,k+1}^-$ 
13  end
14 end

```

Algorithm 4: Updating the estimated position of newly measured features with an EKF.

of feature j is an ellipse whose semi-major (a_j) and semi-minor (b_j) axes are

$$a_j = r_j + 3\sigma_{p,j} + \frac{h}{2}, \quad (3.7a)$$

$$b_j = r_j + 3\sigma_{q,j} + \frac{h}{2}, \quad (3.7b)$$

where $p = x$ and $q = y$ if $\sigma_{x,j} > \sigma_{y,j}$, or $p = y$ and $q = x$ if $\sigma_{x,j} < \sigma_{y,j}$. The estimated radius of the feature (r_j) is conservatively selected as the half of its estimated largest dimension. As was shown in Figure 2.2, the axes are expanded by half the maximum dimension of the rover's footprint (h) to facilitate path-planning. An illustration showing how the occupancy status of vertices is determined is shown in Figure 3.5. All vertices that fall within or are touched by the boundary of the area of influence A_j of a feature are considered occupied. It should be noted that A_j is dynamic, as r_j , $\sigma_{p,j}$

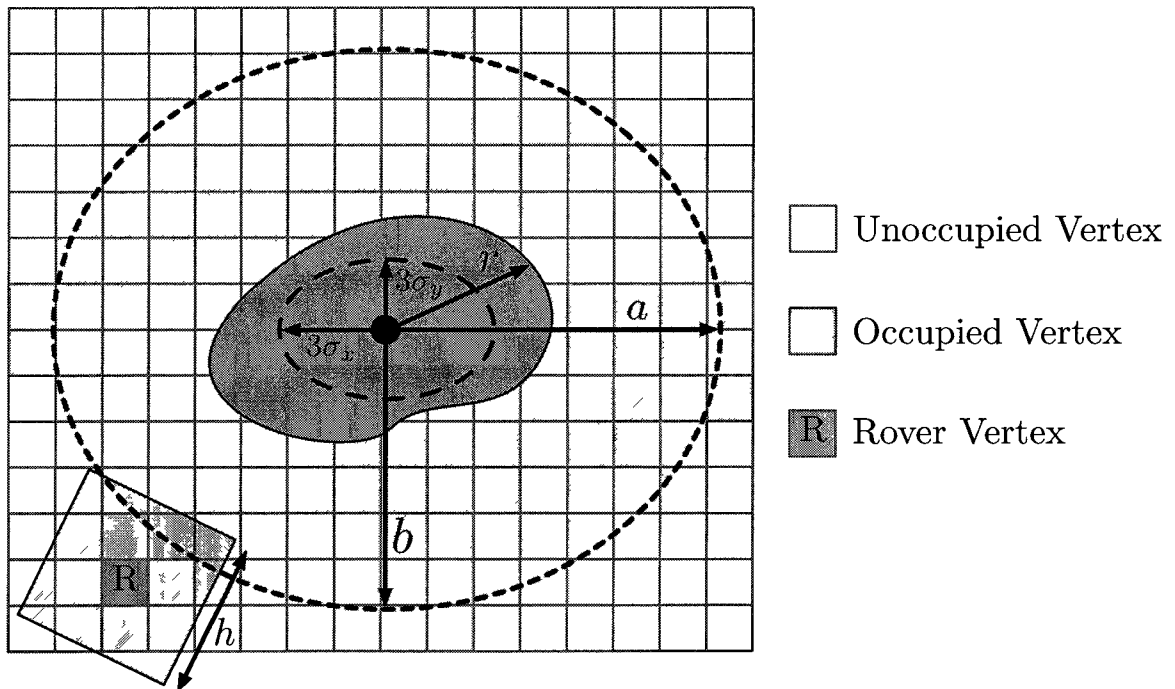


Figure 3.5 – The area of influence of a feature is a function of the feature's radius (r), the uncertainty in its position (σ_x, σ_y) and largest dimension of the rover (h).

and $\sigma_{q,j}$ from Equation 3.7 depend on the current state of the feature. This leads to a

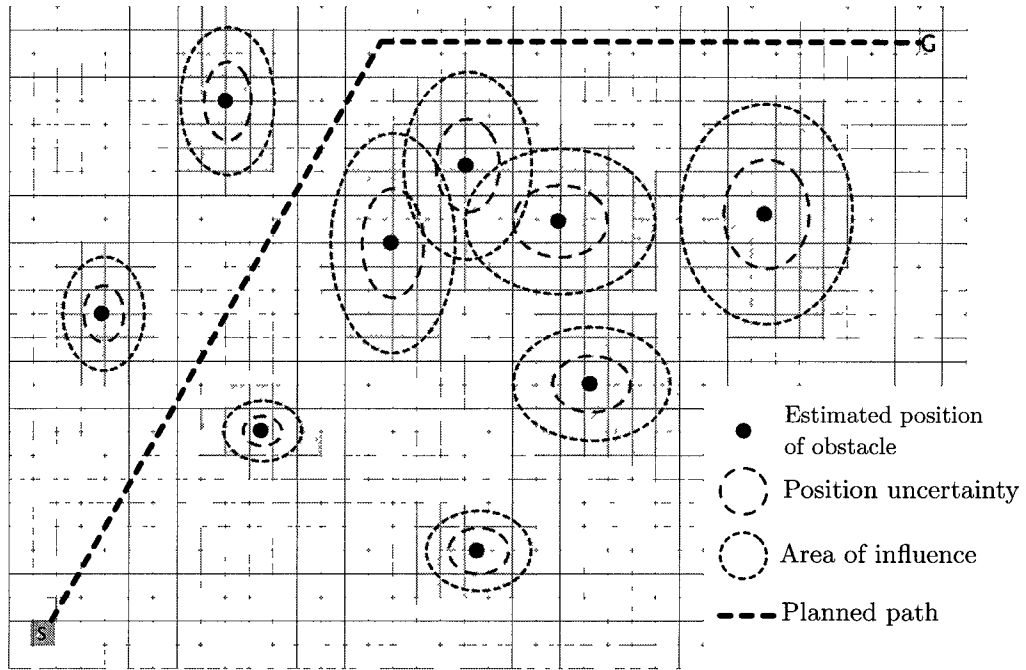
notable effect on path-planning. When features are observed from a greater distance, the covariance of their state tends to be larger, leading to a larger area of influence. As a result, paths planned tend to be more conservative around features with high uncertainty in their state. As more and better measurements of a feature are taken, there is a reduction in its covariance, and by extension, its area of influence. This can alter the original planned path to have a more direct route if previously occupied vertices become unoccupied. An example of this process is shown in Figure 3.6.

3.3.2 Science Vertices

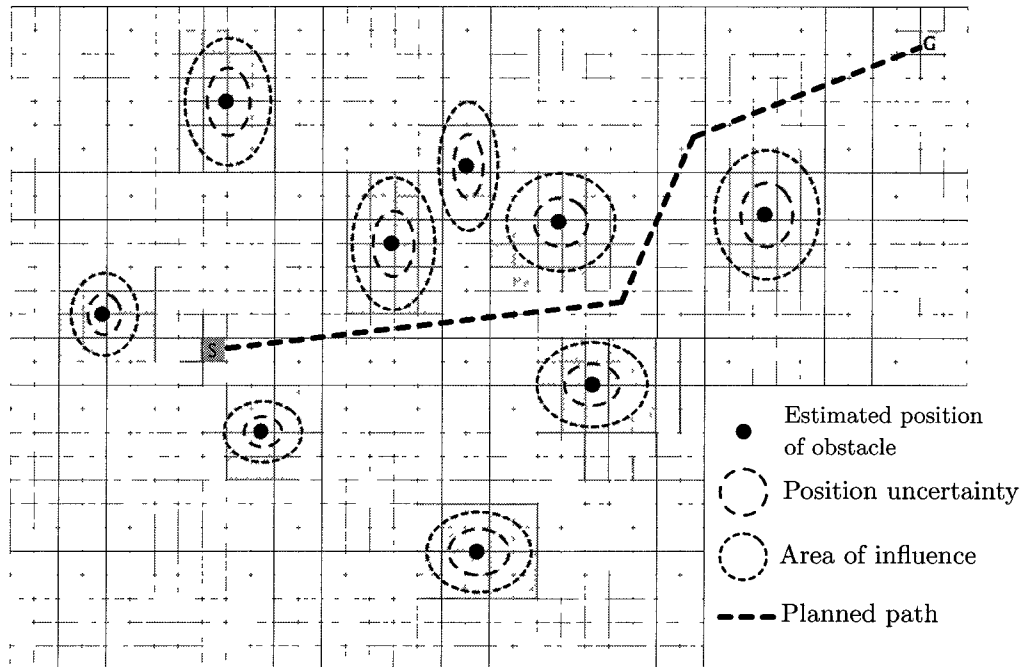
The purpose of the machine vision algorithms is to segment, detect and estimate the position of features. It is often desirable to perform “up-close” observations of the most scientifically valuable features, whether this be with a short-range, narrow field-of-view sensor like a spectrometer, or physically retrieving a sample with a robotic arm. These observations require the rover to travel closer to the feature, to within the range of the sensor(s). Therefore, vertices in the occupancy grid that are both unoccupied and within the sensor radius (r_{sen}) are labelled as “science vertices” for that particular feature. Science vertices are the only vertices that the rover can occupy to perform up-close observations on a feature. Note that the centre of the feature does not need to be within r_{sen} , as reaching the edge of the feature is often sufficient. To ensure reachability for non-round features, the feature’s radius (r) is selected as the half of its estimated smallest dimension. The science radius (r_{sci}) of a feature is defined as

$$r_{sci} = r + r_{sen}, \quad (3.8)$$

where unoccupied vertices within this radius are considered science vertices. This concept is illustrated in Figure 3.7. The set of η science vertices for feature q is described as $\{s_{q,1}, s_{q,2}, \dots, s_{1,\eta}\}$.



(a) High uncertainty initially causes conservative path-planning.



(b) As uncertainty decreases due to closer observations, more efficient path-planning is possible.

Figure 3.6 – An example of how the uncertainty in the estimate of feature positions by the stereo camera can affect path-planning.

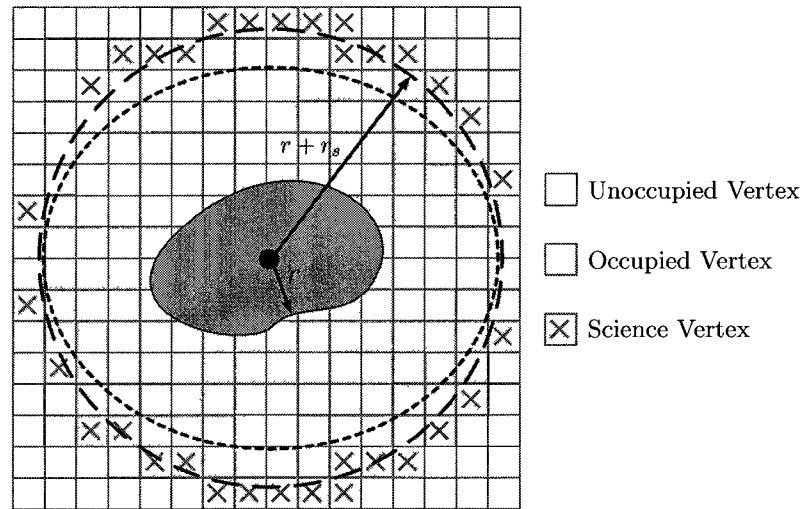


Figure 3.7 – Vertices at which the rover can perform science are all unoccupied vertices from which the science instrument can reach the feature.

3.4 Feature Identification

The purpose of the feature identification algorithm is to estimate the probabilities of the mineralogies of observed features. This allows features to be evaluated and score based on their mineralogy. The algorithm has two key parts to perform identification: object modelling and the softening of evidence using fuzzy logic, and mineralogy estimation using BNs.

3.4.1 Bayesian Network Model

A generalized model of the BN used in this thesis is shown in Figure 3.8. Note the causality of the directed acyclic graph; the N minerals *cause* the states of the M attributes. Every mineral has M child nodes and every attribute has N parent nodes. Minerals have two states (true or false) and attributes A, B, C, \dots, M have $m_A, m_B, m_C, \dots, m_M$ states, where m_i depends on the data set used to initialize the fuzzy membership functions for that attribute.

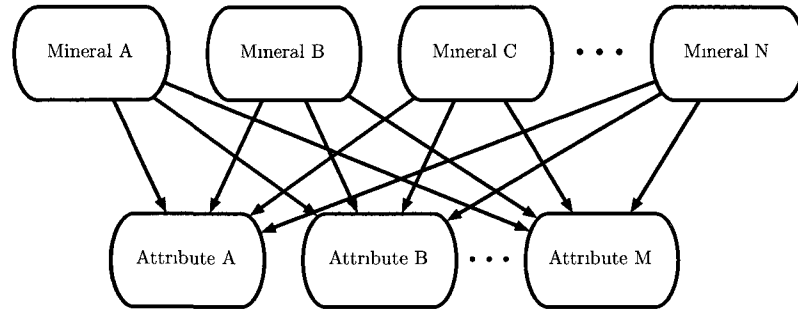


Figure 3.8 – A generalized model of the BN used in this thesis.

3.4.2 Initializing the Fuzzy Membership Functions

The fuzzy membership functions (MFs) are initialized using data sets (as described in Section 2.3.2). Depending on whether the BN will be used to classify real or simulated objects, a data set either consists of images of objects that will be classified (real data set) or set of virtual objects (simulated data set). In both cases, a continuous measurement of each attribute is extracted. For example, if one of the attributes is size, the width of each object in the data set may be extracted. An example of the results of this exercise is shown in Figure 2.8. Depending on the distribution and density of the histogram formed by these measurements, MFs are initialized through observation (one for each possible state).

3.4.3 Training the Bayesian Network

Now that there are MFs for each state of each attribute, a training set that consists of continuous measurements for each attribute can be used to train the discrete BN. This is an extension of the training method described in Section 2.2.5. For example, suppose the BN consisted of two minerals (A and B) and two attributes (Size S and Intensity I). The states of the attributes are $S \in \{\text{small, medium, large}\}$ and $I \in \{\text{low, medium, high}\}$. Using the MFs for each state of each attribute, the continuous measurements of in the data set are converted to truth values, the results of which

are shown in Table 3.1. Using this data the conditional probability tables (CPTs) of each node can be initialized. For example, the conditional probability that a feature is large if it is mineral A is

$$\Pr(S = \text{large}|A) = \frac{0.8535 + 0.9950 + 0.9612 + 0.8980 + 0.8129 + 0.3234}{6} = 0.8073. \quad (3.9)$$

Similarly, the conditional probability that a feature has medium intensity if it is mineral B is

$$\Pr(I = \text{medium}|B) = \frac{0.1247 + 0.5384 + 0.0191 + 0.6509}{9} = 0.1481. \quad (3.10)$$

Note that the truth values for the states of each attribute do not necessarily add up to one (i.e., they are not sum normal). This means there is probability remaining that is reserved for the entries in the CPTs that are not represented in this data set. More specifically, these entries are when the parent nodes of an attribute (the minerals) are multiple minerals or none of the minerals. For example, since it is impractical to suggest that a feature is both mineral A and mineral B, only the latter is used (the feature is neither mineral A nor B). As a result, outlying features that do not match the profiles specified by the data set can be identified as unknown minerals.

Table 3.1 – A data set used to train the Bayesian network. The continuous measurement of each attribute is converted to truth values using the membership functions for each state.

| Measurement | Size Measurements | | | | Intensity Measurements | | | | Mineral A | Mineral B |
|-------------|-------------------|--------|--------|--------|------------------------|--------|--------|--------|-----------|-----------|
| | Continuous | Small | Medium | Large | Continuous | Low | Medium | High | | |
| 1 | 63.8832 | 0.0000 | 0.4447 | 0.5553 | 71.8828 | 0.0000 | 0.1247 | 0.8753 | false | true |
| 2 | 52.0255 | 0.0000 | 0.9190 | 0.0810 | 90.1478 | 0.0000 | 0.0000 | 0.3941 | false | true |
| 3 | 78.6621 | 0.0000 | 0.0000 | 0.8535 | 80.8464 | 0.0000 | 0.0000 | 0.7661 | true | false |
| 4 | 38.3270 | 0.4669 | 0.5331 | 0.0000 | 88.9974 | 0.0000 | 0.0000 | 0.4401 | false | true |
| 5 | 75.1238 | 0.0000 | 0.0000 | 0.9950 | 53.6171 | 0.0000 | 0.8553 | 0.1447 | true | false |
| 6 | 45.0342 | 0.1986 | 0.8014 | 0.0000 | 81.7232 | 0.0000 | 0.0000 | 0.7311 | false | true |
| 7 | 26.1814 | 0.9527 | 0.0473 | 0.0000 | 61.5406 | 0.0000 | 0.5384 | 0.4616 | false | true |
| 8 | 46.3256 | 0.1470 | 0.8530 | 0.0000 | 74.5228 | 0.0000 | 0.0191 | 0.9809 | false | true |
| 9 | 77.7883 | 0.0000 | 0.0000 | 0.8885 | 58.7263 | 0.0000 | 0.6509 | 0.3491 | false | true |
| 10 | 74.0312 | 0.0000 | 0.0388 | 0.9612 | 50.0064 | 0.0000 | 0.9997 | 0.0003 | true | false |
| 11 | 46.0942 | 0.1562 | 0.8438 | 0.0000 | 90.2456 | 0.0000 | 0.0000 | 0.3902 | false | true |
| 12 | 77.5497 | 0.0000 | 0.0000 | 0.8980 | 62.3718 | 0.0000 | 0.5051 | 0.4949 | true | false |
| 13 | 42.2234 | 0.3111 | 0.6889 | 0.0000 | 76.5069 | 0.0000 | 0.0000 | 0.9397 | false | true |
| 14 | 79.6769 | 0.0000 | 0.0000 | 0.8129 | 46.0325 | 0.1587 | 0.8413 | 0.0000 | true | false |
| 15 | 91.9144 | 0.0000 | 0.0000 | 0.3234 | 47.5292 | 0.0988 | 0.9012 | 0.0000 | true | false |

3.4.4 Entering Evidence

Not unlike the data set, measurements entered as evidence must be converted to truth values (soft evidence) using the MFs for the attribute being measured. This soft evidence is used to update the probability all of possible worlds, which can be used to infer the probability that the observed feature is a particular mineral. This process was described in Section 2.2.4. For example, there are nine worlds with non-zero probability where mineral B is true for the BN described in Section 3.4.3. These worlds are shown in Table 3.2. The probability of each world was calculated using (2.5). The total probability that the feature is mineral B is the sum of the probabilities of these worlds ($\Pr(B) = 0.2694$). Now suppose an intensity measurement $I = 59.8621$ was taken from a feature. The truth values that make up the soft evidence are calculated using (2.12) giving evidence of $\{0, 0.6055, 0.3945\}$ for low, medium and high intensities, respectively. The probabilities of the worlds are now updated using (2.9) to yield the new probabilities, shown in the last column of Table 3.2. The new probability that the feature is mineral B is now simply the sum of the new world probabilities ($\Pr'(B) = 0.9236$).

Table 3.2 – The *a priori* and *a posteriori* world probabilities with $B = \text{true}$ after the evidence $I = 59.8621$ is introduced.

| World | S | I | A | B | $\Pr(\text{World})$ | $\Pr'(\text{World})$ |
|-------|--------|--------|-------|------|---------------------|----------------------|
| w_1 | small | low | false | true | 0.0000 | 0.0000 |
| w_2 | small | medium | false | true | 0.0128 | 0.2645 |
| w_3 | small | high | false | true | 0.0319 | 0.0524 |
| w_4 | medium | low | false | true | 0.0000 | 0.0000 |
| w_5 | medium | medium | false | true | 0.0055 | 0.0373 |
| w_6 | medium | high | false | true | 0.1942 | 0.3191 |
| w_7 | large | low | false | true | 0.0000 | 0.0000 |
| w_8 | large | medium | false | true | 0.0110 | 0.2273 |
| w_9 | large | high | false | true | 0.0140 | 0.0230 |

3.5 Feature Evaluation

The probabilities provided by the BN provide some basis on which science-influenced decisions can be autonomously made by the rover. By quantifying the relative scientific value of features based on their mineralogy, the most important features can be identified and queued for close-up investigation. This is the foundation of autonomous science and is critical in achieving the objectives described in Section 1.3.

3.5.1 Benefit Bias Equation

A benefit score (S_b) is assigned to each feature using the *benefit bias equation* (BBE), which is tuned by the operator before a mission to assign higher scores to features whose mineralogies are estimated to match a desirable profile. The BBE takes the form

$$S_b = K \cdot P \cdot \Sigma \cdot [a\text{Pr}(A) + b\text{Pr}(B) + \dots + n\text{Pr}(N)], \quad (3.11)$$

where K is the scale constant, P is the apparent size (also known as angular diameter) of the feature, Σ is the normalized standard deviation of the probabilities, and a, b, \dots, n are the importance weights for minerals A, B, \dots, N . The scale constant K simply scales scores to be in an appropriate scale. It does not affect the relative scores of different features. The apparent size P of a feature is

$$P = 2 \tan^{-1} \left(\frac{r}{d} \right), \quad (3.12)$$

where r and d are the estimated radius and distance to the feature, respectively. The apparent size is measured in radians. It reduces the score of features that have had only a small number of pixels analyzed compared to features with larger numbers of pixels. Another purpose of P is it determines if S_b is updated. Each time a feature is observed, S_b is updated if P is larger than any previous observation. In

general, this means that S_b represents the benefit score calculated when the most visual information was available about that feature.

The normalized standard deviation Σ adjusts S_b based on the standard deviation of the probabilities $\Pr(A), \dots, \Pr(B), \dots, \Pr(N)$. When there is not a lot of reliable information about a feature, the probabilities that it is one mineral versus another are very similar, as it can not be distinguished. As better observations are made, some mineralogies are ruled out as the probabilities of others increase. Eventually, as $\Pr(A) \rightarrow 1, \Pr(B), \Pr(C), \dots, \Pr(N) \rightarrow 0$. The purpose of Σ is to raise the score of features whose probabilities are more distinct and have been narrowed down to one or two minerals; i.e.,

$$\Sigma = \frac{\text{StDev}\{\Pr(A), \Pr(B), \dots, \Pr(N)\}}{\text{StDev}\{i_1, i_2, \dots, i_N\}}, \quad i_k = \begin{cases} 1 & \text{if } k < \frac{N}{2}, \\ 0 & \text{if } k \geq \frac{N}{2}. \end{cases} \quad (3.13)$$

The denominator of (3.13) simply normalizes the standard deviation of the probabilities.

The importance weights a, b, \dots, n for minerals A, B, \dots, N adjust the benefit score based on the current mission of the rover. For example, if mineral A was highly desirable, importance weight a would be given a high value. Conversely, if mineral B was undesirable, b would be given a smaller value relative to a . Note that b could be assigned a negative value if finding mineral B detracts the rover from visiting an area, while assigning b a value of zero would ignore this type of mineral all together.

BBE Example

An example of an application of the BBE illustrating the importance of its variables is as follows. Suppose three minerals (A, B , and C) and their respective importance weights ($a = 2, b = 0.5$ and $c = -1$) have been defined. After capturing an image, two

features are observed (α and β). Assuming a scale factor of $K = 500$, example data extracted from the BN and information calculated using the BBE is shown in Table 3.3. In this example observation, the two features appear to have similar attributes.

Table 3.3 – The results of the observation of features α and β .

| Feature | Pr(A) | Pr(B) | Pr(C) | P | Σ | S_b |
|----------|-----------|-----------|-----------|--------|----------|-------|
| α | 0.28 | 0.21 | 0.16 | 0.0497 | 0.104 | 1.31 |
| β | 0.27 | 0.23 | 0.15 | 0.0523 | 0.106 | 1.40 |

The mineral probabilities are almost identical, as is the normalized standard deviation. However, feature β has a greater apparent size, meaning it occupied more pixels in the image. This resulted in a slightly higher benefit score as more information was available for the analysis. However, in general the benefit scores of both features are low because the normalized standard deviations indicate that there is little consensus in the mineral probabilities. Now, suppose a second observation is later taken of the same two features, resulting in the data shown in 3.4. Interestingly, both features had

Table 3.4 – The results of the second observation of features α and β .

| Feature | Pr(A) | Pr(B) | Pr(C) | P | Σ | S_b |
|----------|-----------|-----------|-----------|--------|----------|-------|
| α | 0.55 | 0.08 | 0.05 | 0.0528 | 0.486 | 13.53 |
| β | 0.55 | 0.31 | 0.24 | 0.0528 | 0.282 | 7.30 |

the same apparent size, meaning their benefit scores were based purely on the results of the BN. In this example, both features had identical probabilities of being mineral A ; however, there remains a much higher probability that feature β is mineral B or C compared to feature α . This indicates greater uncertainty in the true composition of feature β , which is reflected in the normalized standard deviations, and hence the benefit scores.

It should be noted that the effect of the normalized standard deviation on the benefit score is diluted as the number of possible minerals is increased. However, in the case of a high number of mineral possibilities, it is unlikely that every mineral would always maintain a significant probability. Therefore, not including minerals whose probabilities are below some defined threshold in the calculation of Σ in (3.13) may remedy this issue.

3.5.2 Filtering and Selecting Features

It is necessary to filter features that have been detected and evaluated such that the most scientifically valuable features can be identified based on their benefit scores, their location, and the current mission of the rover. A few filters are employed to perform this task. A benefit score filter segregates features based on their benefit score, a detour filter evaluates the reachability of features in terms of their location relative to the rover, and a feature selector determines the best feature that passed through the first two filters.

Benefit Score Filter

The purpose of the benefit score filter is to ignore features with low benefit scores. This could be performed using one of two methods: a static or a dynamic threshold. A static threshold is used by setting a constant benefit score that a feature must exceed to not be ignored by the rover. The advantage of using this method is that it is known with certainty that all features passing through this filter will have a known minimum score. However, as the number of observed features increases, the number of features passing through the filter can increase unbounded. A dynamic threshold is used allowing the benefit score threshold to change depending on the current feature list. For example, the threshold could be set at the mean benefit score, or at one standard deviation above the mean. The advantage of using this method is that the

number of features passing the filter can be controlled as a greater number of features are observed. However, it is possible that a number a features with unacceptably low benefit scores can pass the filter if few high scoring features are observed.

The method employed in this thesis attempts to combine the benefits of both static and dynamic benefit score filters. A static threshold is used if the mean benefit is too low, otherwise a dynamic filter is used; i.e.,

$$S_{b,\min} = \begin{cases} S_{b,\text{stat}} & \text{if } m_D\mu < S_{b,\text{stat}}, \\ m_D\mu & \text{if } m_D\mu \geq S_{b,\text{stat}}, \end{cases} \quad (3.14)$$

where $S_{b,\min}$ is the threshold benefit score that features must exceed to pass through the filter, $S_{b,\text{stat}}$ is the static threshold, μ is the mean benefit score of the feature set, and m_D is an operator-selected multiplier. For example, if a threshold of one standard deviation (σ) above the mean was desired, the multiplier would be $m_D = \frac{\mu + \sigma}{\mu}$.

Detour Filter

The purpose of the detour filter is to ignore features that would require an excessive detour from the current path/mission of the rover to perform an up-close observation. In this scenario, it is assumed the rover has a location it is driving towards, and has limited time or energy to execute its path. Therefore, when features are observed, it is necessary to estimate the cost of altering the path to include a feature as a waypoint to the final goal. This is then compared to a maximum detour allowance set autonomously or by the operator, and features whose detour exceed this allowance are ignored.

All path-planning is performed using Theta* (Section 2.1.3), using binary occupancy grids as cost maps (Section 3.3.1). Calculating the path length of a detour

requires calculating the path length to every science vertex (Section 3.3.2) of a feature and from every science vertex to the goal. This allows for the selection of the science vertex that yields the shortest detour path length. Selecting an incorrect science vertex can moderately increase the detour path length as shown in Figure 3.9. The selected science vertex $s_{q,sci}$ for feature q is the science vertex that satisfies

$$\min [L_{s_{start} \rightarrow s_{q,i}} + L_{s_{q,i} \rightarrow s_{goal}}], \text{ for } i = 1, 2, \dots, \eta \quad (3.15)$$

where $L_{s_a \rightarrow s_b}$ is the path length from vertex s_a to s_b . Features passing through the

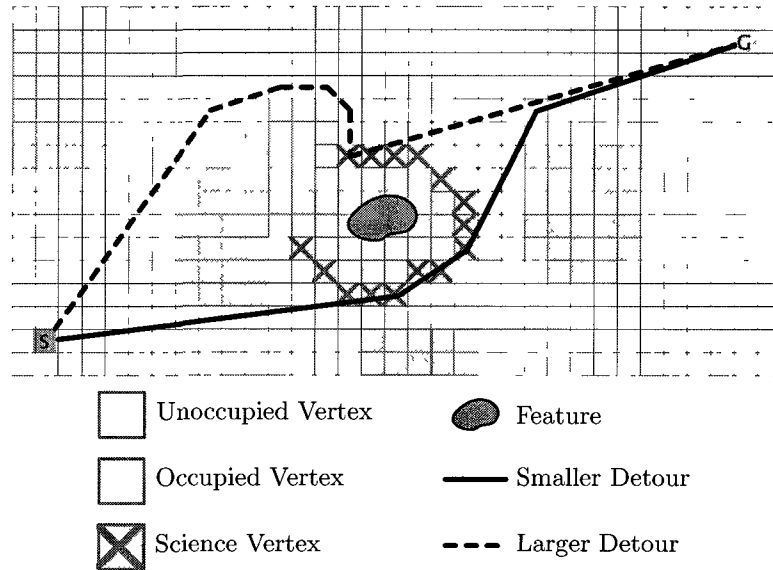


Figure 3.9 – The detour filter requires that the path lengths to all science vertices of a feature are calculated to ensure that the shortest detour is selected.

detour filter must satisfy the inequality

$$L_{s_{start} \rightarrow s_{q,sci}} + L_{s_{q,sci} \rightarrow s_{goal}} \leq DL_{s_{start} \rightarrow s_{goal}}, \text{ for } D \geq 1 \quad (3.16)$$

where D is the detour coefficient. The detour coefficient could be selected using several different methods. For example, a dynamic D could change depending on the

current energy available to the rover and temporal constraints on reaching the goal. In this thesis, a simpler approach is taken. A static D is selected by the operator, which ensures that all detours do not exceed a specified proportion of the most direct path.

Feature Selection

Features that have passed through both the benefit score filter and the detour filter are now ranked by their score-to-detour ratio $\left(\frac{S_b}{L_D}\right)$; i.e.,

$$\frac{S_b}{L_D} = \frac{S_b}{L_{s_{start} \rightarrow s_{q,sci}} + L_{s_{q,sci} \rightarrow s_{goal}}}, \quad (3.17)$$

which specifies the the amount of science value the rover will get per unit distance travelled. The feature with the greatest $\frac{S_b}{L_D}$ is selected as the best target for up-close observation.

Chapter 4

Test Environments

This chapter describes the two environments that were used to test the autonomous science algorithm described in Chapter 3. First, a MATLAB simulation environment was created to test the autonomous science algorithm in its entirety. The simulated rover model, test environment, Bayesian network (BN) and machine vision for this test environment are described in Section 4.1. Next, a “real” laboratory test environment was used to validate using a BN for object identification. The hardware, test features, BN and test scenarios that were used in this validation are detailed in Section 4.2.

4.1 MATLAB Simulator

A simulation environment was created in MATLAB to test the autonomous science algorithm. The goal of this simulation was to compare the two test scenarios discussed in Section 3.1. A simulated rover modelled after a differential drive vehicle was driven through a randomly created rock field with rock size and density approximating Martian terrain. Each time the simulation was run, a simulated data set of features was created that was used to train a BN. Machine vision data was emulated to provide the rover with estimated feature positions and to provide the BN with attribute measurements. The results of the simulator tests are presented in Section 5.1.

4.1.1 Rover Model

The simulated rover was modelled as a 50 x 50 cm differential drive vehicle. It is important to note that the same rover was used to represent both the prime rover and the scout rover in the prime/scout rover pairing scenario. In reality, these rovers should be modelled separately to match the physical and kinematic constraints of true rovers (this point is expanded in Section 5.1.4). The pose of the simulated rover at time k is $p_k = (X_r, Y_r, \theta_r)$, which is illustrated in Figure 3.3. The inputs provided by the controller have the form $u_k = (v_k, w_k)$, where the forward and angular velocities of the vehicle are v_k and w_k , respectively. It is assumed there is no lateral slip on the rover's wheels, which can be modelled as the non-holonomic constraint

$$\underbrace{\begin{bmatrix} \cos \theta_r & -\sin \theta_r & 0 \end{bmatrix}}_{w(p)} \dot{p} = 0. \quad (4.1)$$

Therefore, the rover model consists of all vectors perpendicular to $w(p)$, namely

$$\dot{p} = \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} u. \quad (4.2)$$

As a result, the discrete time model of the pose of the rover is described by

$$\underbrace{\begin{bmatrix} X_{r,k+1} \\ Y_{r,k+1} \\ \theta_{r,k+1} \end{bmatrix}}_{p_{k+1}} = \underbrace{\begin{bmatrix} X_{r,k} \\ Y_{r,k} \\ \theta_{r,k} \end{bmatrix}}_{p_k} + T \begin{bmatrix} \cos \theta_{r,k} & 0 \\ \sin \theta_{r,k} & 0 \\ 0 & 1 \end{bmatrix} u_k, \quad (4.3)$$

at each time step of length T .

4.1.2 Environment

The MATLAB simulated environment that was used to test the autonomous science algorithm was a simplified analog of Martian terrain. A ten-by-ten metre rock field was randomly created for each test run, with a rock size and density distribution modelled after a proposed ExoMars landing site (the Meridiani landing ellipse) [38]. Due to the nature of the cost maps that were used (occupancy grids), small rocks (< 10 cm diameter) were not included in the simulated environment and all other rocks were modelled as insurmountable obstacles (i.e., the rover could drive around but not over them). An example of a typical randomly generated rock field is shown in Figure 4.1. In addition to a location and diameter, each rock was assigned a type of mineral. This mineral type influenced the output of the simulated machine vision algorithm described in Section 4.1.4.

4.1.3 Bayesian Network

A BN was used to identify four different minerals (A , B , C , and D) using evidence of four different attributes (α , β , γ , δ). The BN used a DAG of the generalized form illustrated in Figure 3.8. Each attribute had three possible states (α_1 , α_2 , α_3 , etc.). Each mineral was randomly given a *mineral description*, which is a set of states that nominally describe the physical appearance of the mineral (e.g., mineral A could be given a mineral description of $\{\alpha_2, \beta_3, \gamma_1, \delta_3\}$). The fuzzy membership functions for each of these states were triangles with trapezoidal parameters $\{0, 25, 25, 50\}$, $\{25, 50, 50, 75\}$, and $\{50, 75, 75, 100\}$ for the first, second and third states, respectively (see Section 2.3.1 for more information on trapezoidal parameters).

The BN was trained using a simulated data set of 300 features. Each feature in this set was first randomly assigned one of the four minerals. Using the mineral description of the assigned mineral as a bias, the feature was assigned a random (continuous)

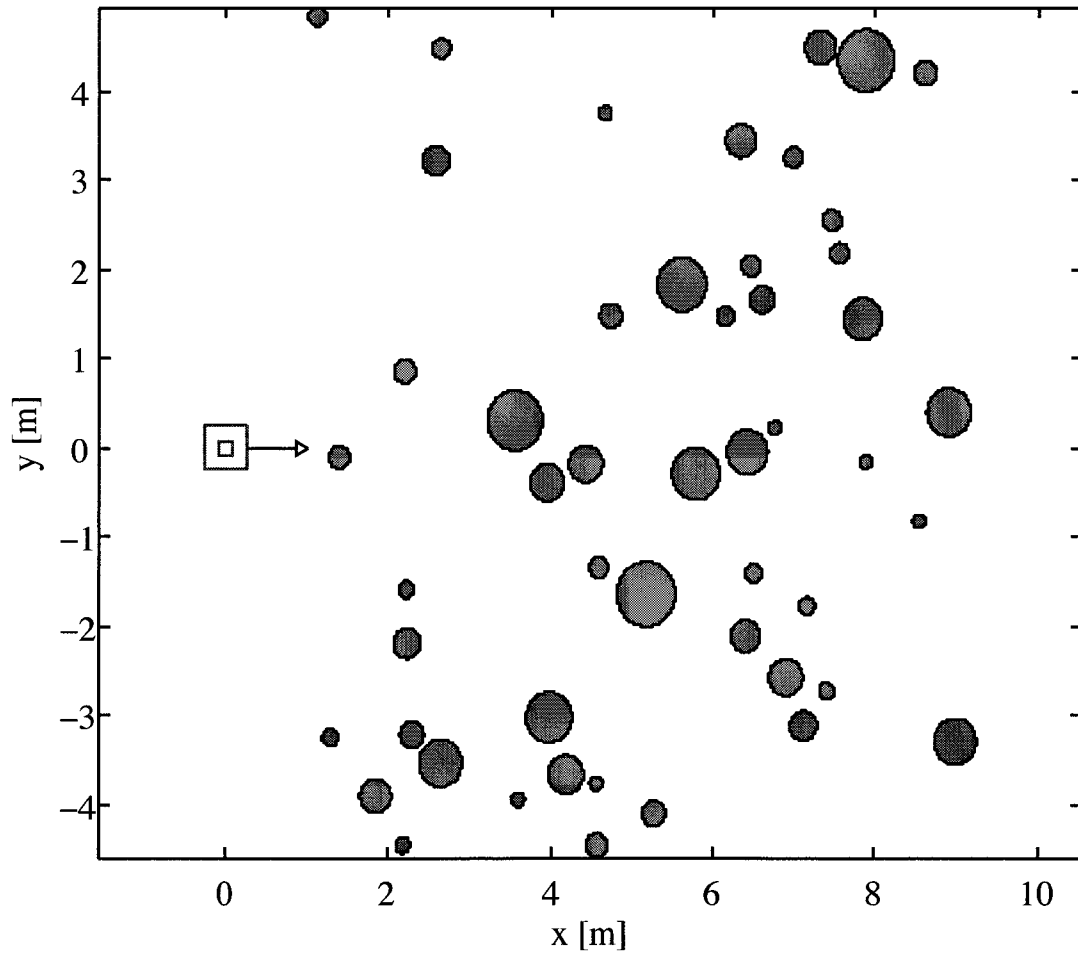


Figure 4.1 – An example of the randomly generated rock fields used in the simulator. The simulated rover is shown at (0,0).

measurement for each of the four attributes (e.g., if the mineral description of the feature for attribute α was α_1 , the randomly assigned α measurement would have a mean of 25). Once the full set of measured features was created, the BN was trained using the method described in Section 3.4.3.

4.1.4 Machine Vision

Machine vision was simulated to supply the data required for the autonomous science algorithm. A brief overview of the algorithms this process attempted to emulate is provided in Section 2.4.1. The machine algorithm provides two pieces of information: an estimate of the ranges and angles to features within the camera's range and field-of-view, and continuous measurements of each of the attributes in the BN. For position and ranges, the features were modelled and updated using the technique described in Section 3.2. For the continuous attribute measurements, the technique that was used to simulate measurements for the training data set described in Section 4.1.3 was used. The attribute measurement of a mineral was its true state (the peak of triangle of its fuzzy membership function) plus some noise. The range and field of view of the camera of the simulated camera were 5 m and 66° , respectively, based on the specifications of the stereo camera described in Section 4.2.1 (although the usable range depends on the image processing algorithms that are used).

4.1.5 Test Scenarios

Two different test scenarios were simulated for comparison. The first (prime rover solo) is described in Section 3.1.1 and is illustrated by the block diagram in Figure 3.1. The second (prime/scout rover pairing) is described in Section 3.1.2 and is illustrated by the block diagram in Figure 3.2. The parameters used for the simulations are listed in Appendix A. Thirty trials were performed. A single trial consisted of running each

of the two scenarios using the same randomly generated feature map and the same BN and BBE parameters.

4.2 Laboratory

Laboratory testing was performed to assess the performance of the Bayesian approach of identifying objects. The focus of this test was the identification segment (Section 3.4) of the autonomous science algorithm, which involves first using a training data set to initialize the BN, and then using the BN to identify features from a series of tests. A stereo camera was used to capture images and the machine vision methods described in Section 2.4.1 were used to segment objects and measure their attributes. The results of the laboratory tests are presented in Section 5.2.

4.2.1 Hardware

The exteroceptive sensor used for the laboratory tests was a stereo camera. As was discussed in Section 2.5, cameras are currently the most commonly used device to capture data to be used for autonomous science. Stereo cameras improve this capability by allowing for the use of stereo vision to estimate the 3D position of features captured in an image. This information is required to allow features to be mapped and eventually used for path-planning, which is critical in achieving the goals outlined in Section 1.3.

The laboratory testing employed the use of a Point Grey Bumblebee XB3 CCD Camera, shown in Figure 4.2. This is a triclops (three lens) camera system, although only the two outer lenses were used as a stereo pair. This model has a field-of-view of 66° , a resolution of 1280 x 960 pixels and a baseline of 24 cm. Images were captured in colour but converted to grayscale before being processed. Detailed specifications of the Bumblebee XB3 are provided in Table 4.1.

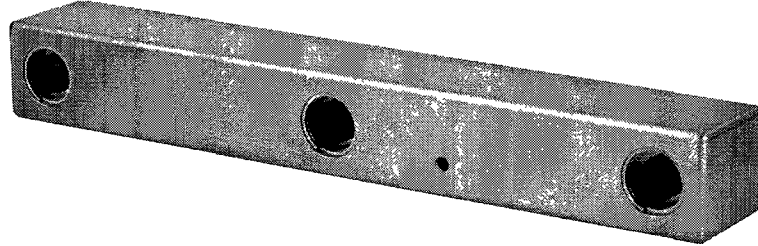
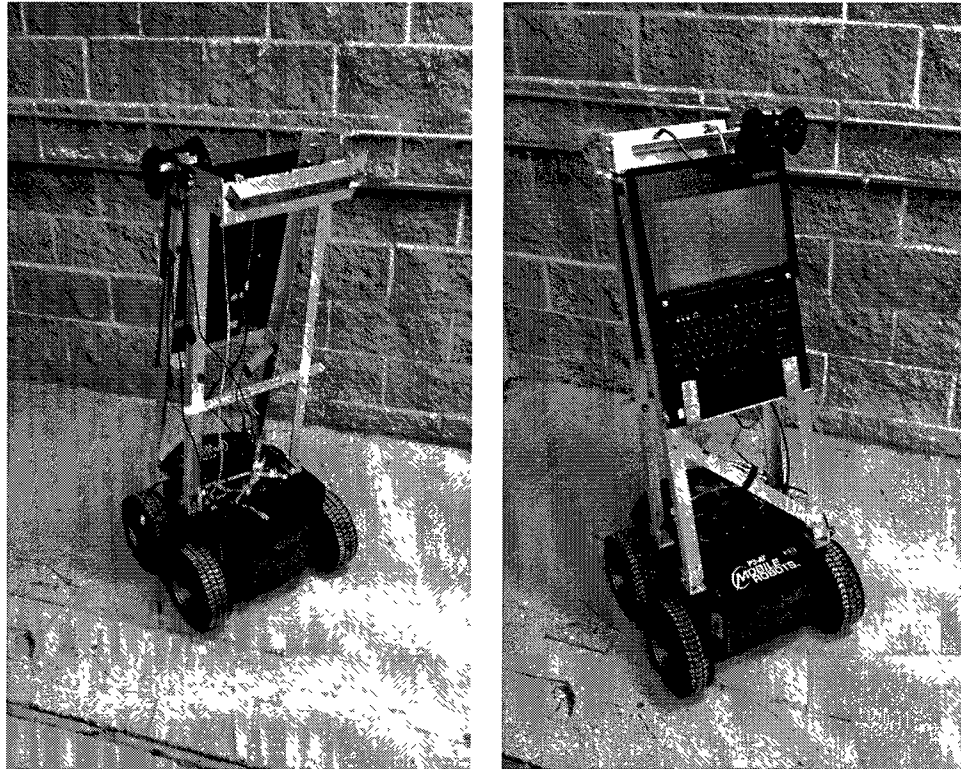


Figure 4.2 – Point Grey Bumblebee XB3 CCD Camera [39].

Table 4.1 – Point Grey Bumblebee XB3 CCD Camera technical specifications [39].

| Property | Value |
|-----------------------|-----------------------|
| Baseline | 24 cm |
| Resolution | 1280 x 960 |
| Focal length | 3.8 mm |
| Aperture | f/2.0 |
| Field-of-view | 66° |
| Signal-to-noise ratio | 54 dB |
| Data interface | IEEE-1394b (Firewire) |

The stereo camera was mounted on a Pioneer 3-AT mobile robot on a static mast at a height of 1.17 m from the ground, tilted downward at an angle of 30° . Features were measured with respect to the left camera. The mobile robot simply acted as a “mobile tripod”. No data was used from its sensors and it was driven with a joystick. Localizing the robot (and hence, the camera) in the global reference frame was performed via manual measurements each time an image was captured. The stereo camera mounted on the mobile robot is shown in Figure 4.3.



(a) The front of the rover showing the static mast and the tilted camera.

(b) The rear of the rover showing the joystick and the laptop used to store the images.

Figure 4.3 – Point Grey Bumblebee XB3 CCD Camera mounted on the Pioneer 3-AT mobile robot.

4.2.2 Features

Thirteen different objects were selected to represent features for the laboratory testing. Included in Appendix B are images of these objects and their identifiers. Although artificial objects were used in place of natural features for the initial validation of the system presented in this thesis, it was attempted to select objects that shared some similar attributes (shape, size, texture, etc.) with each other. Note that images were converted to grayscale before being processed, so colour could not be used to distinguish between objects. Although using colour would have greatly facilitated classification, the stark differences in colour among the objects does not reflect the spectrum one might expect on a planetary mission. Additionally, as one mineral can often vary in colour, colour is used with caution for mineral identification.

4.2.3 Bayesian Network

A BN was used to identify 13 different objects: pink volleyball (PVB), white volleyball (WVB), small soccer ball (SSB), large soccer ball (LSB), bowl (BWL), small football (SFB), large football (LFB), white plastic ball (WPB), yellow plastic ball (YPB), small green pot (SGP), small brown pot (SBP), medium pot (MPT), and large pot (LPT). Evidence from six different attributes (size, intensity, contrast, energy, entropy, IDM) was used. Colour was not used for identification for the reasons stated in Section 4.2.2. The BN used a DAG of the generalized form illustrated in Figure 3.8.

To condition the BN for objects detected in different parts of an image, a training grid with 30 different possible object positions was used to create a training set. This training grid is illustrated in Figure 4.4 and an example of one of the training images is shown in Figure 4.5. Each of the 13 objects was cycled through each of the 30 possible positions, which were at five different headings (-30° , -15° , 0° , 15° , and

30°) and six different ranges (1, 2, 3, 4, 5, and 6 m). A continuous measurement of each of the six attributes was taken for every object that was identified. This data set was used to train the BN using the method described in Section 3.4.3. The attribute states and fuzzy membership functions were determined by creating histograms of the training set, as was discussed in Section 2.3.2.

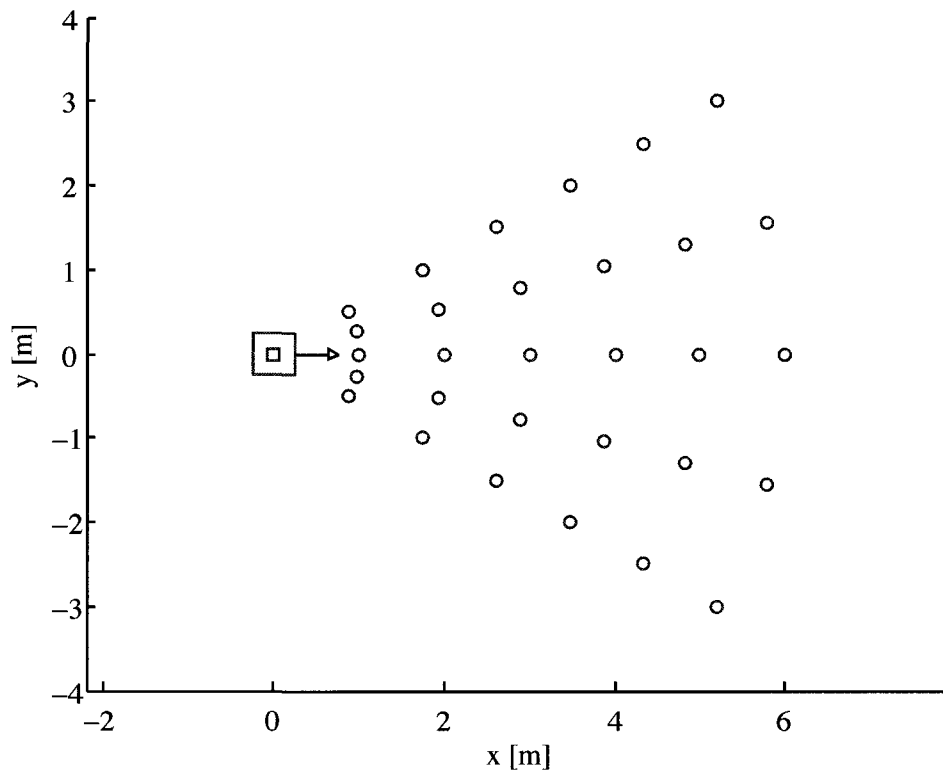


Figure 4.4 – The training grid used for the laboratory training set.

4.2.4 Test Trials

The trained BN was used to identify features in four test trials. In each of these trials, the rover was driven approximately ten metres through a field of 30 features, capturing 9–15 images along the way. The 30 features were comprised of six small

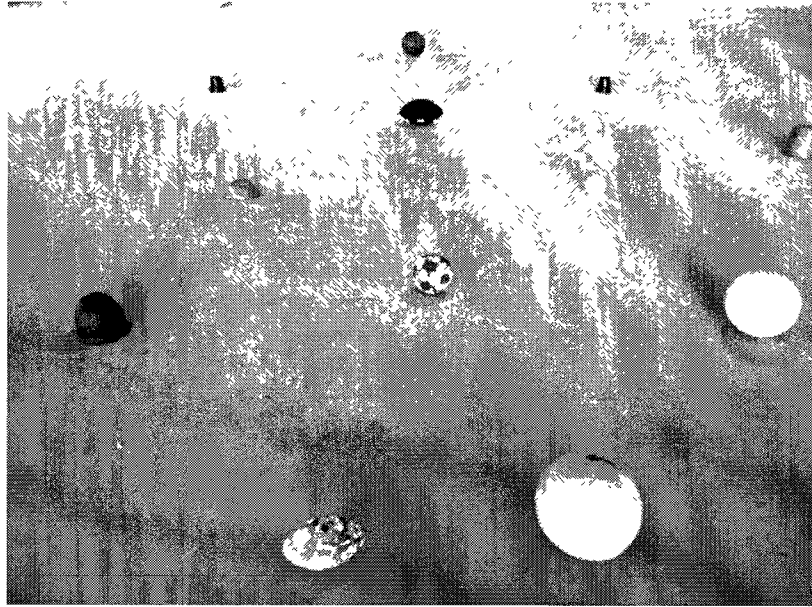


Figure 4.5 – One of the 30 images used to generate the training set for the Bayesian network.

brown pots, six small green pots, three white plastic balls, three yellow plastic balls, four bowls, and one of each of the remaining eight objects specified in Section 4.2.3. Objects were randomly placed in 30 different static feature locations before each trial. An illustration of the 30 feature locations is shown in Figure 4.6. The rover began at the origin in each trial. The machine vision algorithms detected and measured a varying number of features in each image. The mineral type of each feature was estimated using the BN and scored using the benefit bias equation (Section 3.5.1). The scale constant was set to $K = 10$ and the importance weights that were used are shown in Table 4.2. The importance weights were selected such that some objects were valued higher than others, and undesirable objects were given negative weights. This was meant to reflect how one might weigh mineral types differently based on the goals of the rover.

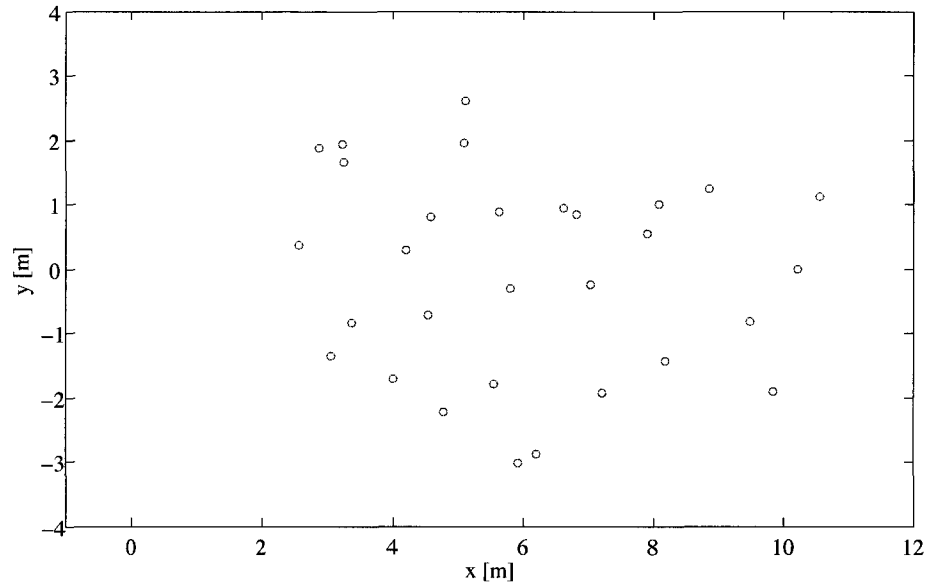


Figure 4.6 – The 30 feature locations in the laboratory tests.

Table 4.2 – The importance weights used in the laboratory tests.

| Object | Importance weight |
|---------------------|-------------------|
| Pink volleyball | -1 |
| White volleyball | +1 |
| Small soccer ball | -1 |
| Large soccer ball | +3 |
| Bowl | -3 |
| Small football | -1 |
| Large football | -1 |
| White plastic ball | +3 |
| Yellow plastic ball | -1 |
| Small green pot | +1 |
| Small brown pot | +1 |
| Medium pot | -3 |
| Large pot | -1 |

Chapter 5

Results and Analysis

This chapter outlines the results of the two types of testing that were performed to evaluate the autonomous science algorithm: the simulator and the laboratory tests. The environments and important parameters used in these tests are outlined in Chapter 4. In Section 5.1, the results from the simulator are presented and analyzed, with a focus on comparing the scenarios outlined in Section 3.1. Additional notes about improvements to the algorithm that could be implemented based on these results are also included. In Section 5.2, the results of the laboratory tests are described in detail. This section is focused on the performance of the feature identification and feature evaluation algorithms presented in Sections 3.4 and 3.5, respectively. Once again, possible algorithmic improvements are also discussed.

5.1 Simulator

The environment and parameters that were used in the simulated tests are described in Section 4.1. The purpose of the simulator was to assess and compare the performance of the autonomous science algorithm in two different scenarios: the prime rover solo and the prime/scout rover pairing. Observations of the behaviour of the each of the two scenarios are outlined in Sections 5.1.1 and 5.1.2. Results comparing

the performance of the two scenarios are presented and discussed in Section 5.1.3. Finally, improvements to the autonomous science algorithm based on the results of the simulator are discussed in Section 5.1.4.

5.1.1 Prime Rover Solo

The prime rover solo scenario is illustrated in Figure 3.1. The rover created and updated a feature map during its traverse. Features whose identities were estimated by a BN were evaluated at each update step. The goal of the rover was to visit the feature with the highest score-to-detour ratio without violating the filters described in Section 3.5.2. After visiting the feature for up-close observation, the rover continued to the final goal. An example of resulting feature map and path travelled by the rover in this scenario is shown in Figure 5.1. In this example, the rover visited the feature in the dashed circle before continuing to the goal located at (10, 0) m.

A number of observations were made concerning the behaviour of the prime rover solo scenario over the course of the thirty trials that were performed. One common issue was the appearance of *kinks* in the path executed by the rover. An example of a kink is shown in Figure 5.2. Kinks were found to occur due to one of two reasons: indecisiveness in the feature selection or closures in the occupancy grid. Occasionally, the feature with the highest score-to-detour ratio would not maintain this status upon having its mineral probabilities updated after a new observation. As a result, the rover would drive towards this feature for a short time before selecting a new feature in a new direction, producing the kink in the executed path. In other cases, the rover would detect a new feature that surpassed the current best feature and would therefore plan a new path. However, previously unseen obstacles would be observed that immediately obstructed the newly planned path, causing the rover to suddenly require a deviation from its current trajectory. In addition, if this new detour surpassed the maximum allowable detour, the feature now violated the detour

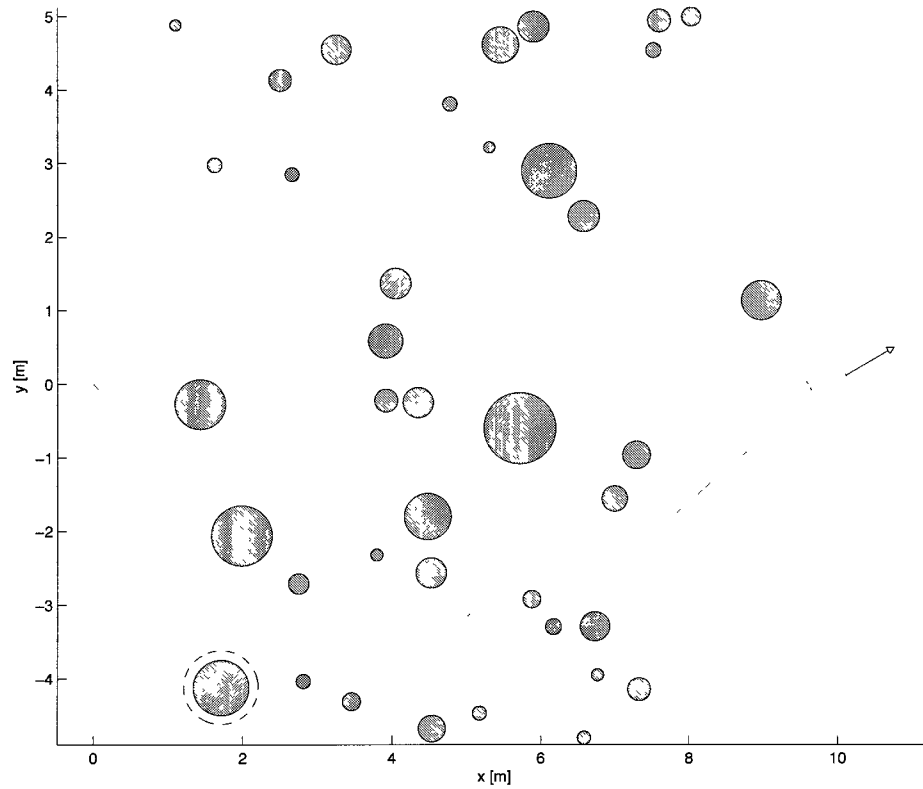


Figure 5.1 – An example of a path travelled by the prime rover solo. The feature visited for up-close observation is labelled with a dashed circle.

filter and a new feature had to be selected, possibly worsening the kink. One method that could be employed to prevent kinks is to penalize turns during path planning, perhaps adding cost as a function of the severity of the turn.

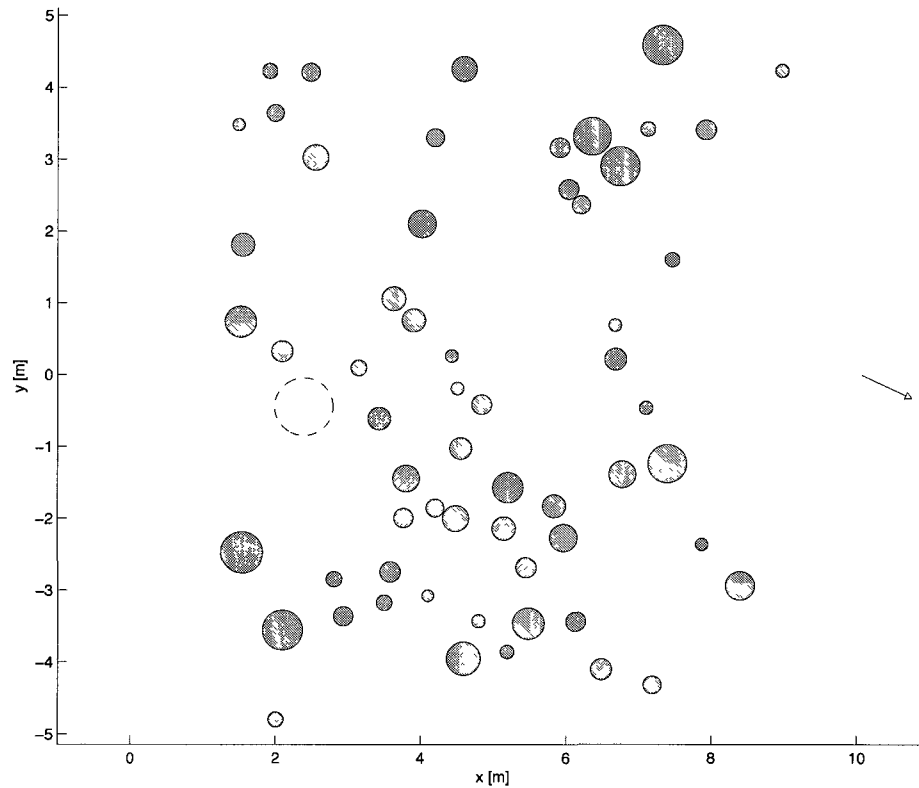


Figure 5.2 – An example of a kink in the path of the prime rover solo is shown in the dashed circle.

The disadvantage of having no *a priori* information about the features was apparent in some of the prime rover solo trials. One of these disadvantages is the possibility of *backtracking* being required to reach the final goal. An example where backtracking occurred is shown in Figure 5.5. In this trial, after travelling to the top feature, the rover discovered previously unobserved obstacles that blocked its path. The path planner calculated that shortest path to the final goal required the rover to back-track. The importance of backtracking (or other unforeseen detours) is that it can cause the violation of the constraint specified by the detour coefficient. If the detour

coefficient was selected based on the power available to the rover at the beginning of the traverse, the rover may not be able to reach the final goal if the detour violation is too large.

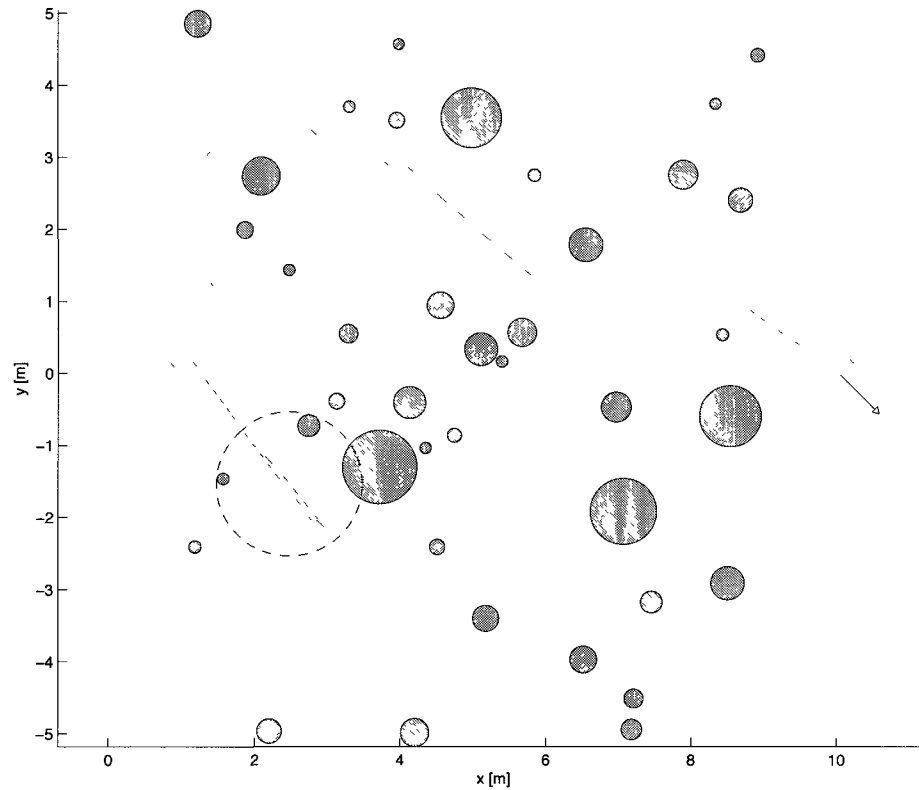


Figure 5.3 – An example of backtracking by the prime rover solo is shown in the dashed circle.

Another disadvantage that afflicted the prime rover solo scenario was the occurrence of *premature feature selection*. As the rover had no information about the features closer to the goal and no guarantee that these features would have high benefit scores, the rover was often forced to visit a feature early in its traverse only to find high-scoring features later. An example of premature feature selection is shown in Figure 5.4. It is conceivable that if the “one feature maximum” constraint that was specified in this thesis was lifted, future high-scoring features could be visited. However, if a large portion of the detour allowance is used to visit the first feature,

high-scoring future features may need to be bypassed to ensure the final goal can be reached.

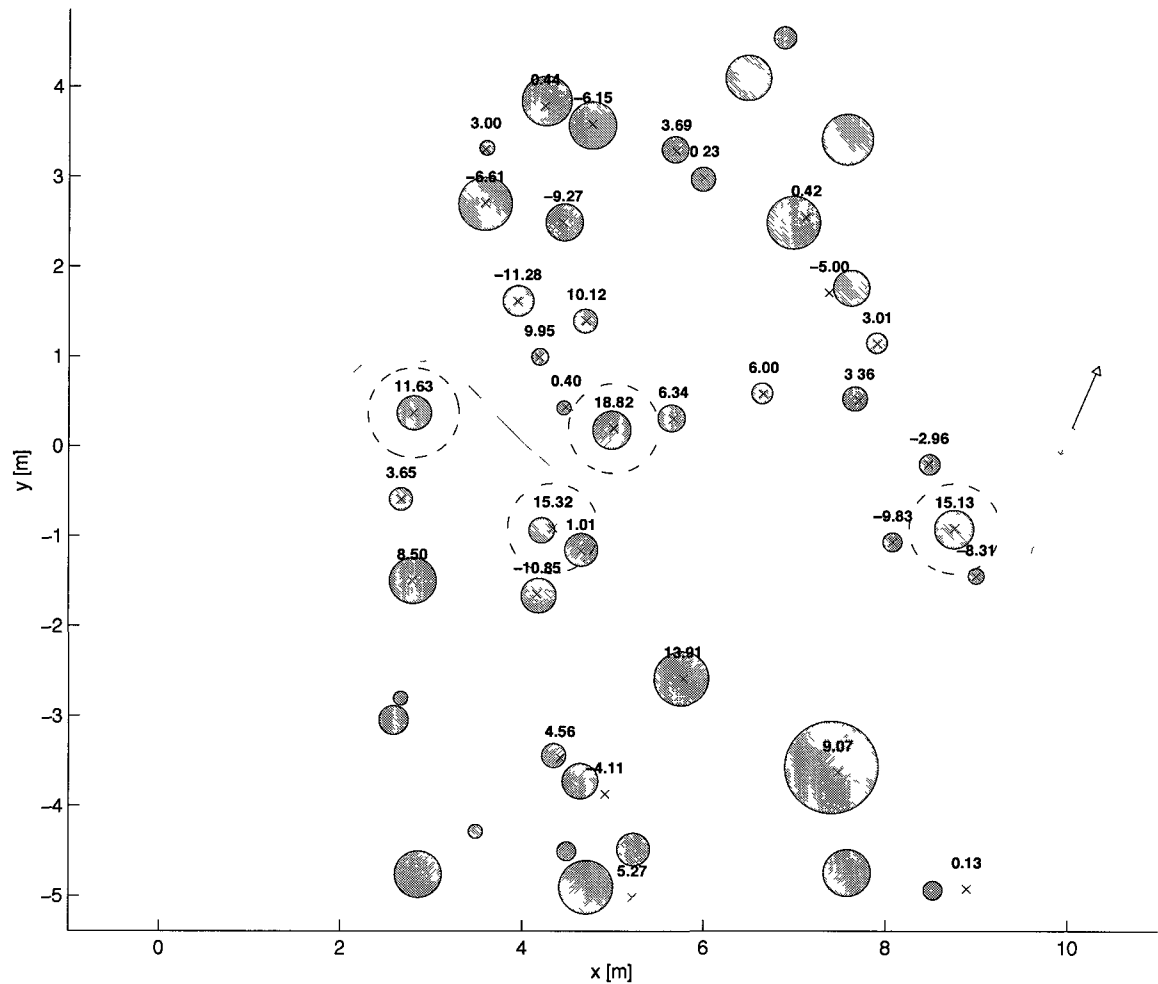


Figure 5.4 – An example of premature feature selection by the prime rover solo. Benefit scores are shown above the estimated locations of the observed features (each labelled with an “x”). The selected feature is in the dashed circle while three possibly better choices are shown in the dash-dotted circles.

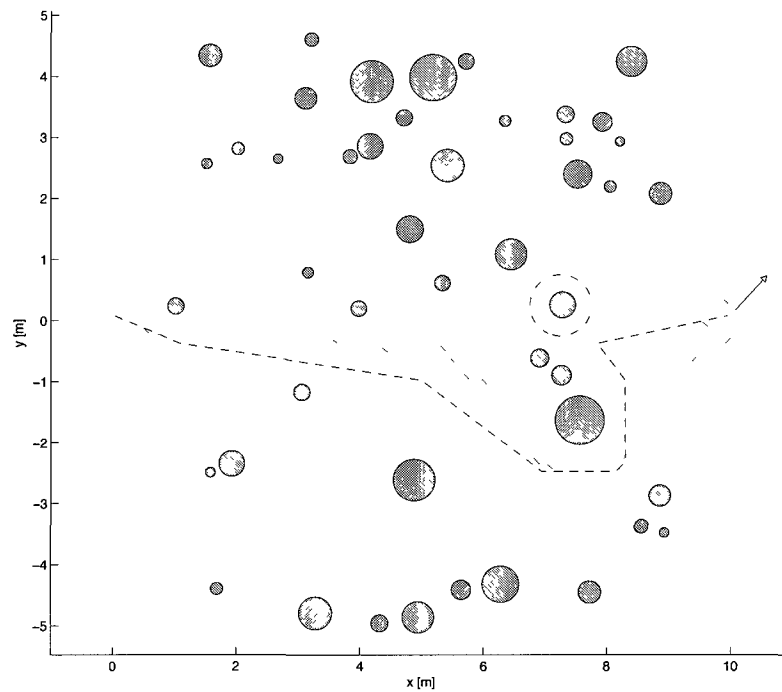
5.1.2 Prime/Scout Rover Pairing

The prime/scout rover pairing scenario is illustrated in Figure 3.2. This scenario consisted of two parts: the scout traverse and the prime traverse. During the scout traverse, the scout rover travels to the final goal while performing feature localization,

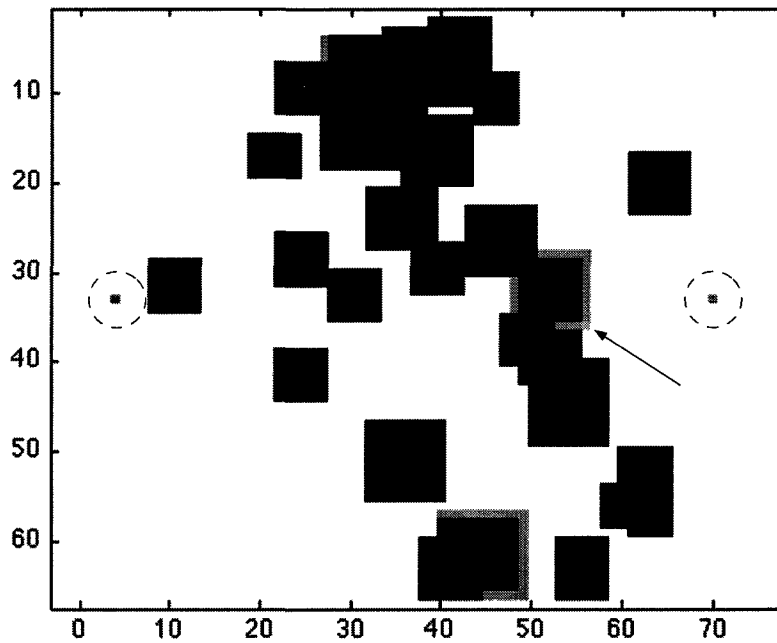
feature identification, and feature evaluation. The information gained from this traverse is given to the prime rover for the prime traverse, which uses it to visit a single feature for up-close observation along the way to the final goal. An example of the results of a scout rover traverse outlining the information that is relayed back to the prime rover is shown in Figure 5.5. This information includes the estimated location of all observed features, their mineral probabilities and their benefit scores. Also included is the feature with the greatest score-to-detour ratio and a recommended path (i.e., the path that was used in the detour calculation) for visiting that feature.

One of the issues that negatively affected the prime rover solo scenario was backtracking. Although backtracking is often not avoidable during the scout traverse (an example is shown in Figure 5.6), the prime rover can avoid this unnecessary detour using the information provided to it. The importance of this distinction was touched upon when discussing the motivation of using a scout rover in Section 1.1. Scout rovers are envisioned to be small, rugged, inexpensive and possibly expendable. As a result, they are designed to investigate dangerous areas, such as those where backtracking may be required to escape. By using a scout rover, sending the large, expensive prime rover into a dangerous area can be avoided while information about the scientific value and the dangers of that area can still be investigated.

In recommending the best feature to visit, the information provided to the prime rover by the scout rover has already considered the required detour to visit that feature. However, there can be cases where the recommended path is found to be unsuitable due to new observations by the prime rover. During execution of the recommended path, if the prime rover encounters obstacles previously unobserved by the scout rover, the path must be appropriately adjusted. An example of this situation is illustrated in Figure 5.7. Note that “positive” adjustments are also possible if improved estimates of feature locations open up a shorter path than what was previously possible. Either way, these adjustments were not accounted for in the original feature



(a) Feature map.



(b) Occupancy grid.

Figure 5.5 – Examples of paths, feature selection and an occupancy grid generated by the scout rover. In (a), the solid line is the scout path and the dashed line is selected detour path. The highest ranking feature is labelled by a dash-dotted circle. In (b), the start and goal vertices are labelled by dashed circles and the selected science vertex is labelled with an arrow. The empty vertices considered during feature selection are shaded gray.

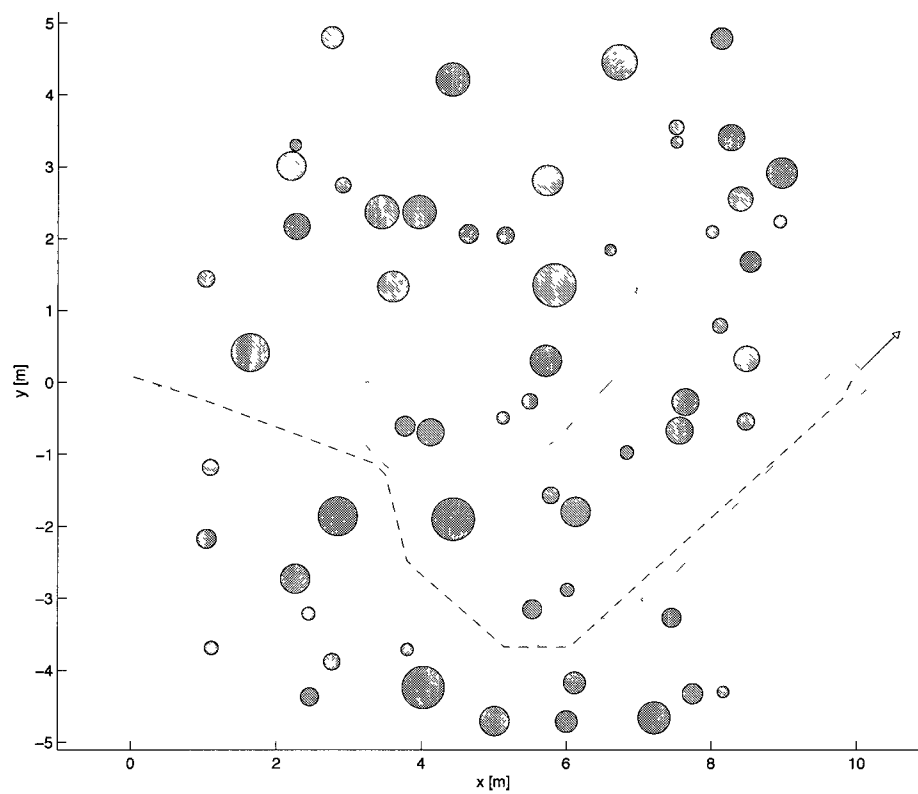


Figure 5.6 – The path executed by the scout rover (solid line) prevented backtracking in the suggested path for the prime rover (dashed line).

selection, which may negatively affect the score-to-detour ratio. One possible solution to this issue is setting boundaries on the areas of the map in which the original path can be planned, preventing a path to simply be planned “around” all the obstacles, when in fact it is just traversing through an unexplored area.

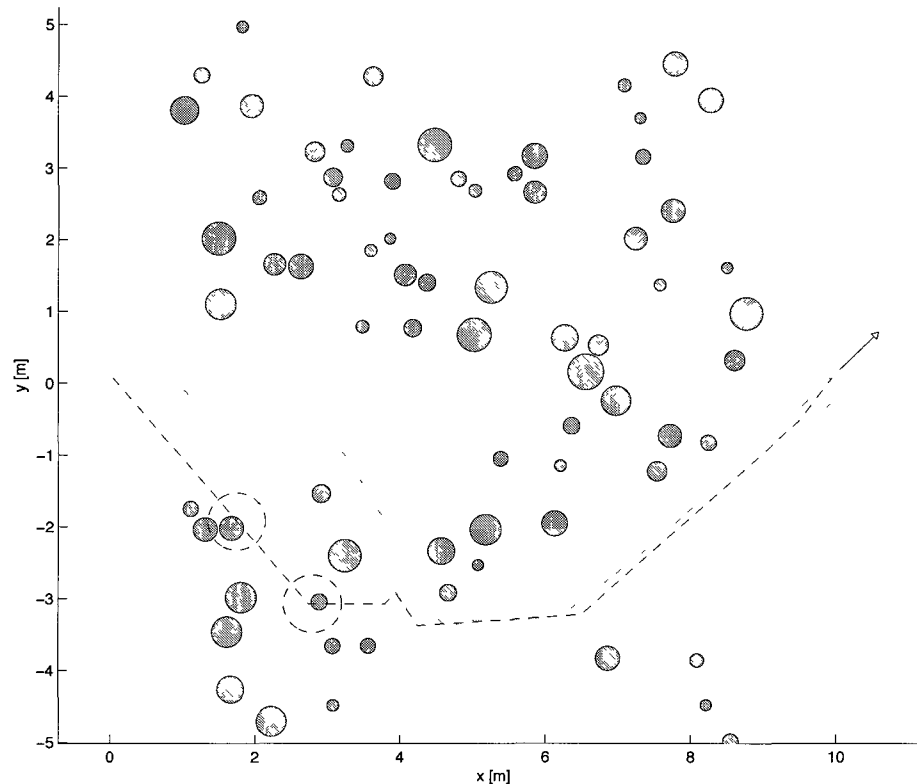


Figure 5.7 – The path executed by the prime rover (solid line) significantly deviating from the recommended path provided by the scout rover (dashed line) due to the previously unobserved obstructions in the original path (shown in the dashed circles).

5.1.3 Scenario Comparison

In each of the thirty trials, the total distance travelled by the prime rover, the benefit score of the feature observed up-close, the score-to-detour ratio and the percentage of features that were observed was recorded for both scenarios. The results for all the trials are provided in Appendix D. A quantitative comparison of the performance of the two scenarios requires the results of each individual trial to be compared, as the

random feature properties, feature distribution and importance weights changed in each trial. This meant the absolute highest possible score and the absolute shortest possible path was different in each trial. Therefore, score, distance and observed features comparisons were based on percent difference from the prime rover solo results; i.e.,

$$L_{D\text{compare}} = \frac{L_{D\text{prime+scout}} - L_{D\text{primeSolo}}}{L_{D\text{primeSolo}}} \times 100 \%, \quad (5.1a)$$

$$S_{b\text{compare}} = \frac{S_{b\text{prime+scout}} - S_{b\text{primeSolo}}}{S_{b\text{primeSolo}}} \times 100 \%, \quad (5.1b)$$

$$\left(\frac{S_b}{L_D}\right)_{\text{compare}} = \frac{\left(\frac{S_b}{L_D}\right)_{\text{prime+scout}} - \left(\frac{S_b}{L_D}\right)_{\text{primeSolo}}}{\left(\frac{S_b}{L_D}\right)_{\text{primeSolo}}} \times 100 \%, \quad (5.1c)$$

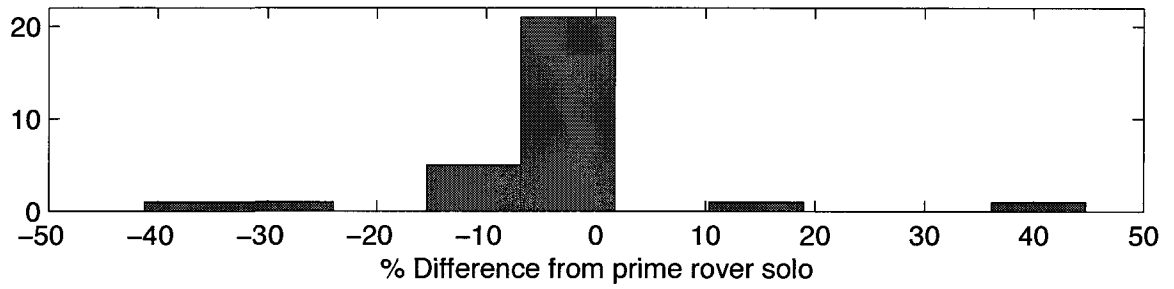
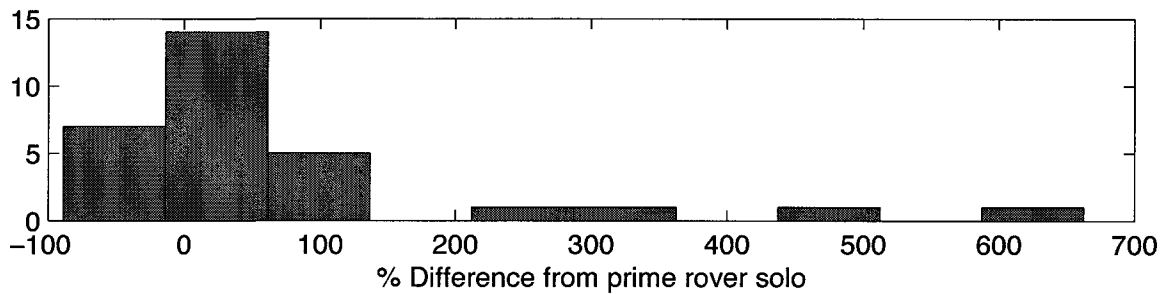
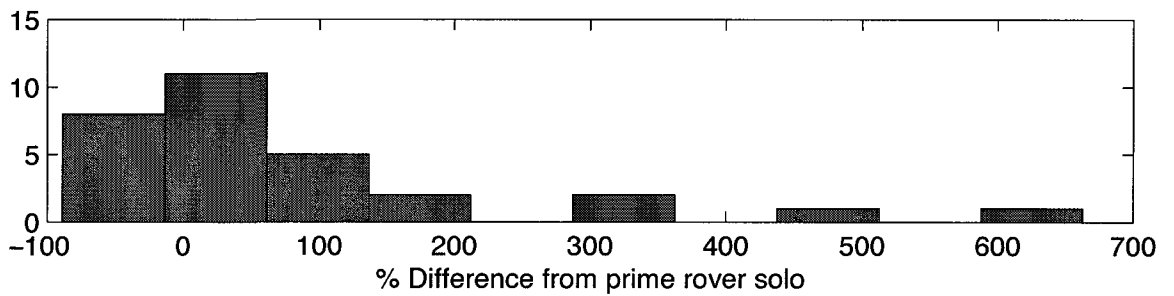
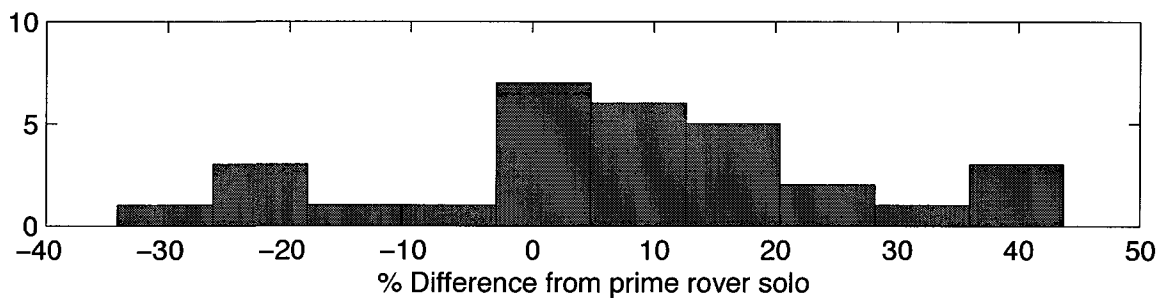
$$(\%FO)_{\text{compare}} = \frac{(\%FO)_{\text{prime+scout}} - (\%FO)_{\text{primeSolo}}}{(\%FO)_{\text{primeSolo}}} \times 100 \%, \quad (5.1d)$$

where the detour length (L_D), benefit scores (S_b) and percentage of randomly generated features that were observed ($\%FO$) comparisons are used to quantify the results. A histogram illustrating the distribution of the results is provided in Figure 5.8 and a summary of the results is presented in Table 5.1.

Table 5.1 – A quantitative comparison of the two test scenarios over the course of 30 trials.

| Result | Mean | Median | St. dev. | Minimum | Maximum |
|---|---------|---------|----------|----------|----------|
| $L_{D\text{compare}}$ | -2.30 % | -0.48 % | 13.15 % | -41.27 % | 44.76 % |
| $S_{b\text{compare}}$ | 78.38 % | 22.23 % | 162.87 % | -88.62 % | 662.82 % |
| $\left(\frac{S_b}{L_D}\right)_{\text{compare}}$ | 84.47 % | 22.66 % | 164.07 % | -88.57 % | 663.01 % |
| $(\%FO)_{\text{compare}}$ | 7.20 % | 8.33 % | 19.34 % | -34.09 % | 43.75 % |

Although not the case in every trial, the mean values suggest that the prime rover performed better when information from a scout rover was available. The most

(a) Detour length ($L_{D_{compare}}$).(b) Benefit score ($S_{b_{compare}}$).(c) Score-to-detour ratio $\left(\left(\frac{S_b}{L_D}\right)_{compare}\right)$.(d) Percentage of features observed $(\%FO)_{compare}$.**Figure 5.8** – Histograms illustrating the distribution of the simulator results.

important result, $\left(\frac{S_b}{L_D}\right)_{\text{compare}}$ (as this was the parameter the autonomous science algorithm was attempting to maximize), has the greatest mean and median improvement for the prime/scout rover pairing compared to the other results. The source of this advantage is apparent—the improvement in benefit score ($S_{b_{\text{compare}}}$) was much greater than the improvement in detour distance travelled ($L_{D_{\text{compare}}}$). This result indicates that the trials where the prime rover solo scenario was impaired by premature feature selection (e.g., Figure 5.4) were more common or had a greater impact than distance-based handicaps such as kinks or backtracking. The median (and maximum) results indicate the mean performance of the prime/scout rover pairing scenario was greatly influenced by trials where improvements over the prime rover solo were substantial. Once again, the improvement in these extreme trials was due to an exorbitant increase in benefit score, not in a reduction of detour distance. However, benefit score improvements must be taken with caution, as benefit scores are generated by the autonomous science algorithm itself (e.g., poor identification by the BN can lead to misleading benefit scores; this situation was observed in the laboratory tests in Section 5.2.3). However, assuming sound identification, the improvement in benefit score from using a scout rover can be considerable.

One of most noticeable results in Table 5.1 are the wide ranges of the result parameters as indicated by the standard deviation and the minimum and maximum values. Observations of the paths and feature maps produced by the two scenarios over the course of the 30 trials provides some insight on the causes of this result. Explanations of the handicaps endured by the prime rover solo scenario (e.g., kinks, backtracking and premature feature selection) have explained possibilities why its performance suffered in some trials (see Section 5.1.1). In these trials, the results tended towards the extreme, favouring the prime/scout rover pairing scenario. Conversely, problems with the prime/scout rover pairing scenario (e.g., deviations from the recommended path) often resulted in the opposite extreme, favouring the prime

rover solo. However, also common was the nominal case, where the two scenarios performed similarly. This case occurred most often when the most direct path to the final had only minimal obstructions, and a high-scoring feature was discovered close enough to the start position that premature feature selection by the prime rover solo was avoided. An example of such a case is shown in Figure 5.9. Histograms of the results (Figure 5.8) show that the nominal case was the most common, and the extreme cases that favoured the prime/scout rover pairing scenario occurred more frequently than those that favoured by the prime rover solo scenario. The distribution of the two extreme cases and the nominal case likely caused the wide range of results.

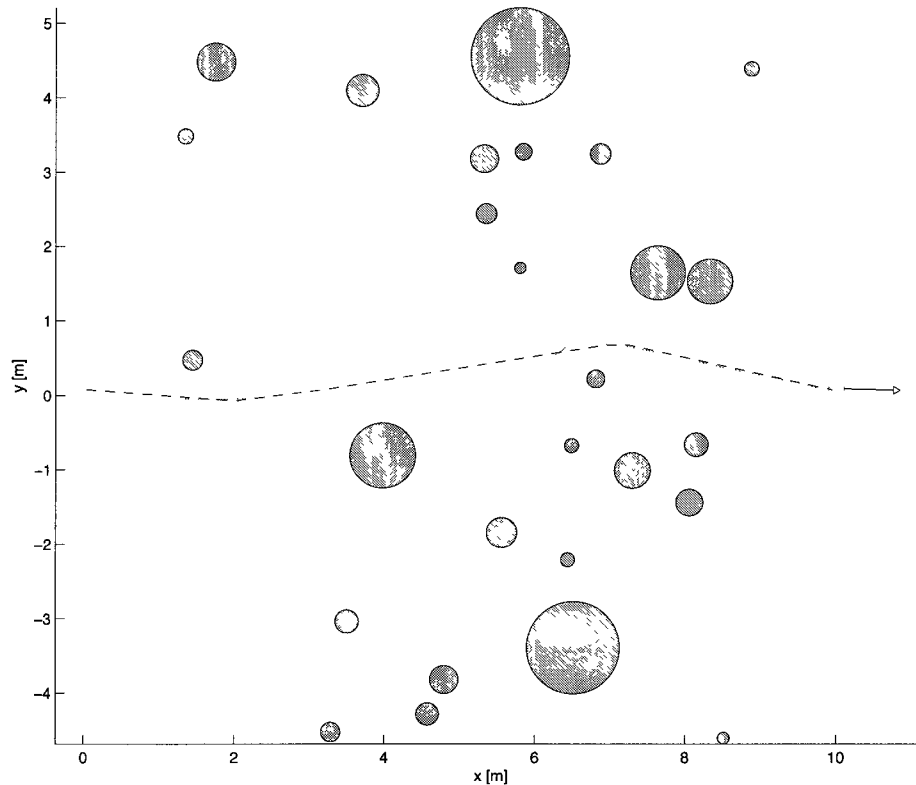


Figure 5.9 – An example of the nominal case where the result from the two scenarios is similar. The path of prime rover solo (solid line) and of the prime rover with scout information (dashed line) have only small deviations.

5.1.4 Possible Improvements

Observations of the behaviour of the autonomous science algorithm in the two different scenarios provide insight into how the algorithm could possibly be improved. Some improvements can be made to address the individual weaknesses of each scenario and others can improve the system overall. The following points specify some possible improvements:

Link path-planning and to realistic vehicle models

Kinks in the path often formed in the prime rover solo scenario due to indecisiveness in feature selection. The rover would make sudden turns if features with better score-to-detour ratios were found. However, the calculated detours considered only the path length, not the original configuration of the rover or the number/sharpness of turns. By linking the path-planning and the vehicle model, paths could be planned that consider the cost of executing the path based on the abilities of the rover. Furthermore, as the physical attributes and kinematic constraints of the scout and prime rover will likely be different, they should be modelled separately to improve realism.

Allow multiple up-close observations

The primary reason why the number of up-close observations is restricted to one in this thesis is computational requirements. To calculate the detour length, (3.15) requires that the lengths of two paths are calculated for every science vertex on the map (from the rover to the science vertex and from the science vertex to the final goal). However, if there is the possibility of visiting more than one science vertex, the number of paths that must be planned increases substantially. For example, if two features have ten science vertices each, 600 paths would need to be planned to account for all the possible routes the rover could take (compared to 40 if only one feature was visited). However, it is possible

that multiple up-close observations could be accommodated without substantially increasing the computational requirements. Paths could be planned that continue to only consider one feature, but once that feature is visited, the algorithm is free to visit additional features as long as no constraints are broken. If sufficient detour allowance remains after the first feature is investigated, the rover could continue to another. This could augment the science return in both scenarios, and would particularly help trials with premature feature selection in the rover prime solo scenario.

Constrain size of occupancy grid during feature selection

When performing feature selection, the rover calculates the necessary detour required to visit each science vertex. However, it is possible that some of these detour paths include vertices that have never been observed (which are considered empty until observations suggest that they are not). As a result, the calculated detour path length is erroneous if future observations reveal obstacles obstructing the path. One solution to this problem would be constraining the size of the occupancy grid during feature selection to only include observed vertices. However, this solution is not entirely suitable because a path to the final goal must be planned, even if the location of the goal has not yet been observed. One way to partially implement the observed vertices solution would be to apply this constraint only on the paths *to* the science vertices, but not on the paths from the science vertices to the goal.

5.2 Laboratory

The methods and equipment used in the laboratory tests are described in Section 4.2. The purpose of these tests were to assess the effectiveness the feature identification and evaluation algorithms in a more realistic scenario. Results concerning identification

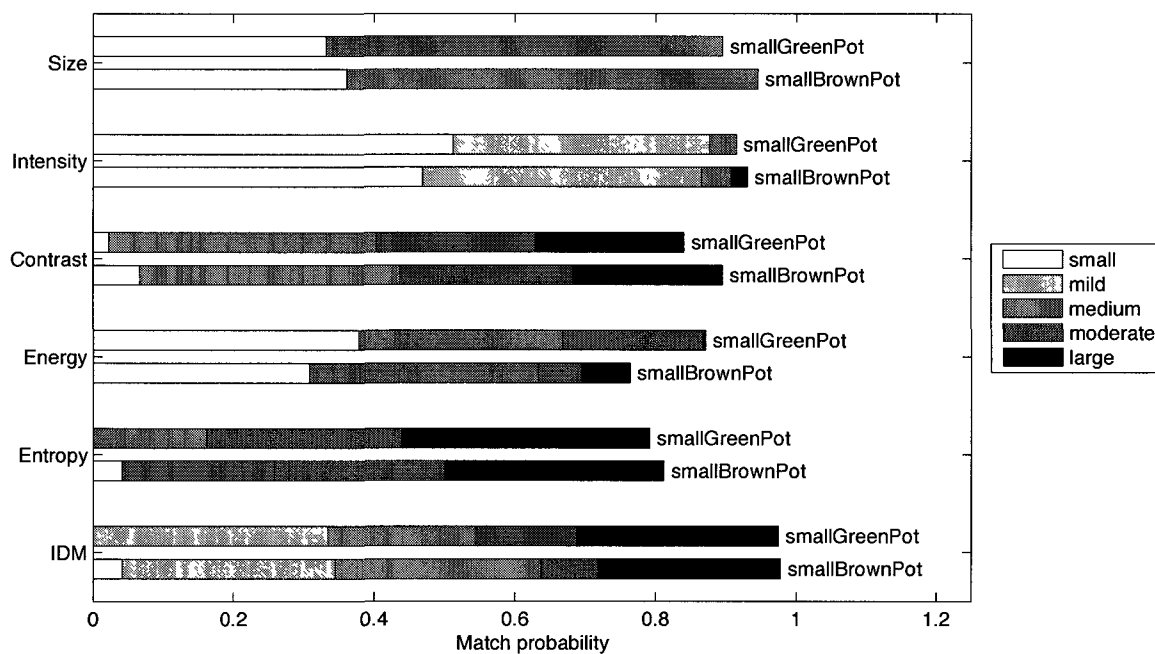
are presented and discussed for the training process in Section 5.2.1 and for the four test trials in Section 5.2.2. The results and discussion concerning the effectiveness of the evaluation algorithm is provided in Section 5.2.3. Finally, some discussion about possible general improvements to the developed algorithms based on the results is included in Section 5.2.4.

5.2.1 Training

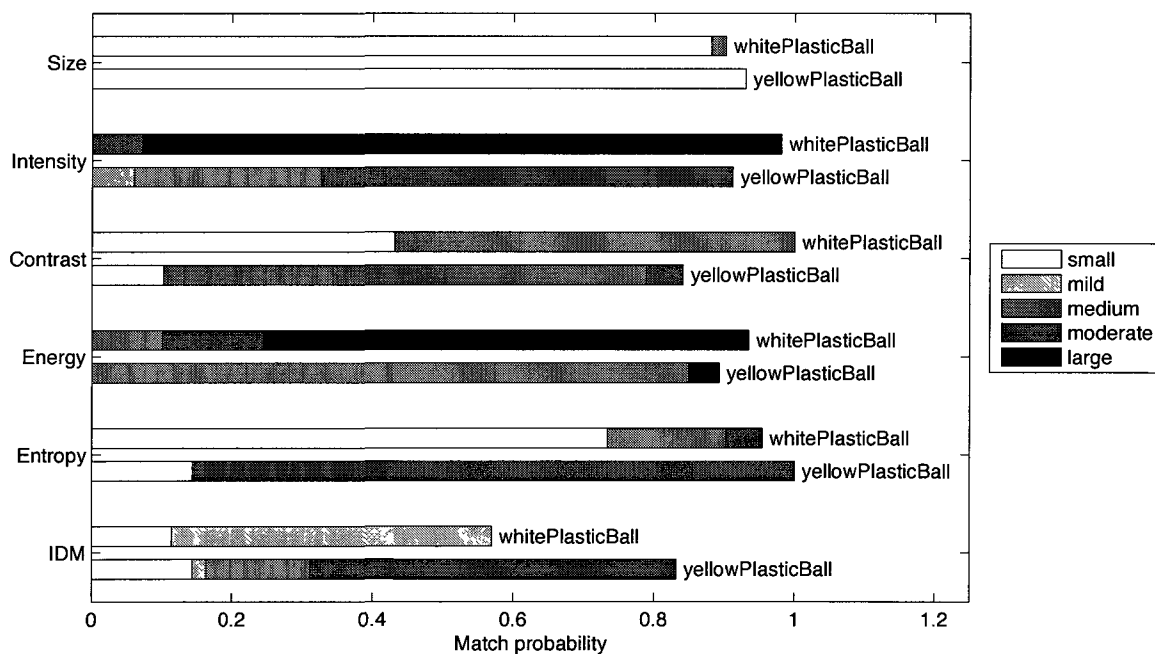
The BN was trained using the data set that was collected in the method described in Section 4.2.3. However, not every object was properly segmented and detected by the machine vision algorithms in every image. Overall, 171 objects were identified over the 30 images. Histograms of the continuous measurements for each attribute were used to specify the trapezoidal parameters of the fuzzy membership functions, using the process outlined in Section 2.3.2. The trapezoidal parameters for each state of each attribute are provided in Appendix C. As was indicated in Section 2.3.1, the resulting parameters are not sum normal, allowing for outlying measurements to be considered an “unknown” object.

The resulting attribute states memberships of each of the 13 objects after the training process are shown in Table E.1 in Appendix E. Similarities in the membership values among some of the objects provide insight into possible difficulties in differentiating objects during the test trials. If state memberships are too similar for most or all the attributes between two objects, the BN may split the probability between the similar objects. This is especially true if attributes that form the primary differences between two objects are not measured or are measured poorly. Objects that shared similar characteristics and were at risk of difficult differentiation are compared in Figure 5.10.

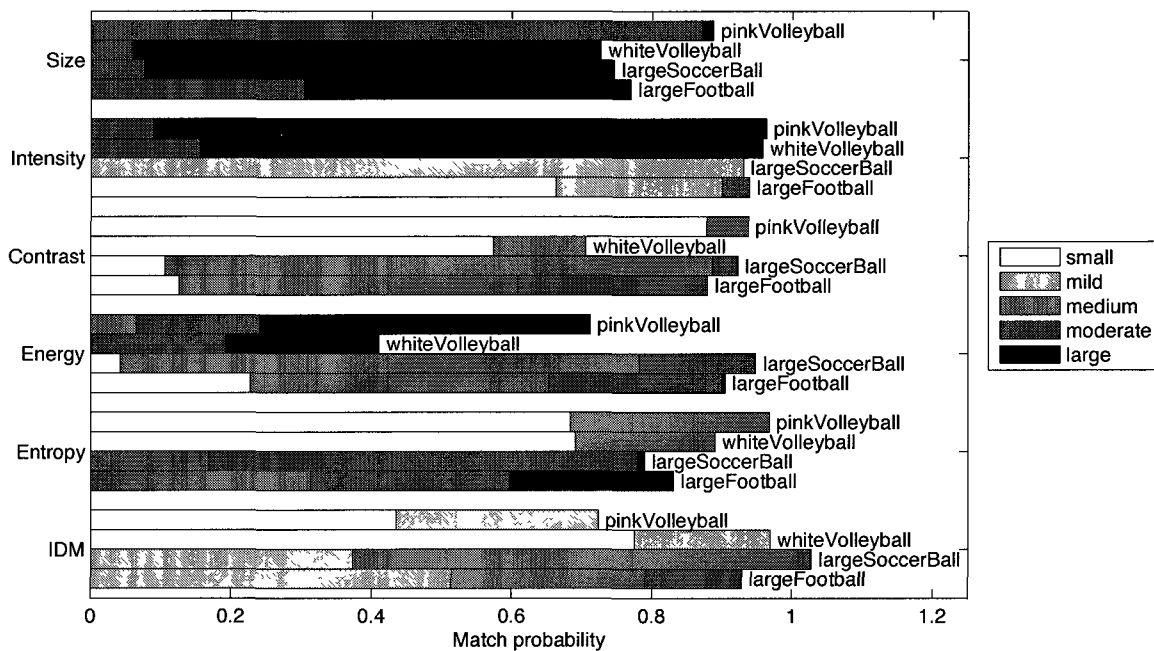
The omission of colour as an attribute (see Section 4.2.2 for explanation) can cause objects similar in size and texture have similar state memberships. Obvious objects



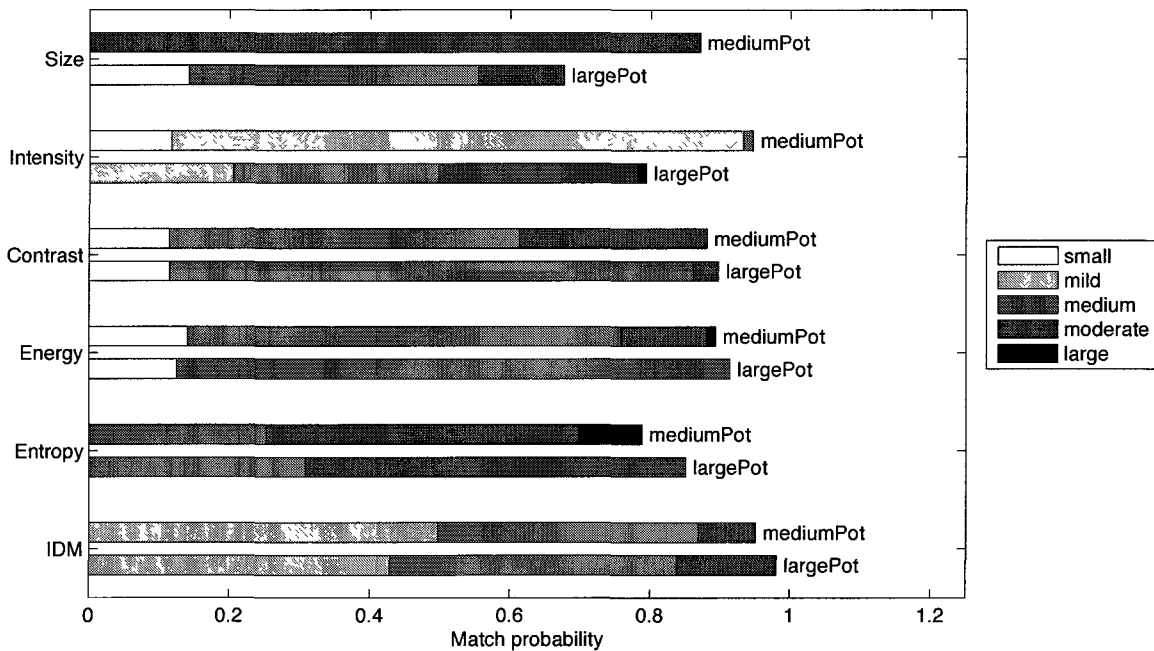
(a) Small green pot vs. small brown pot.



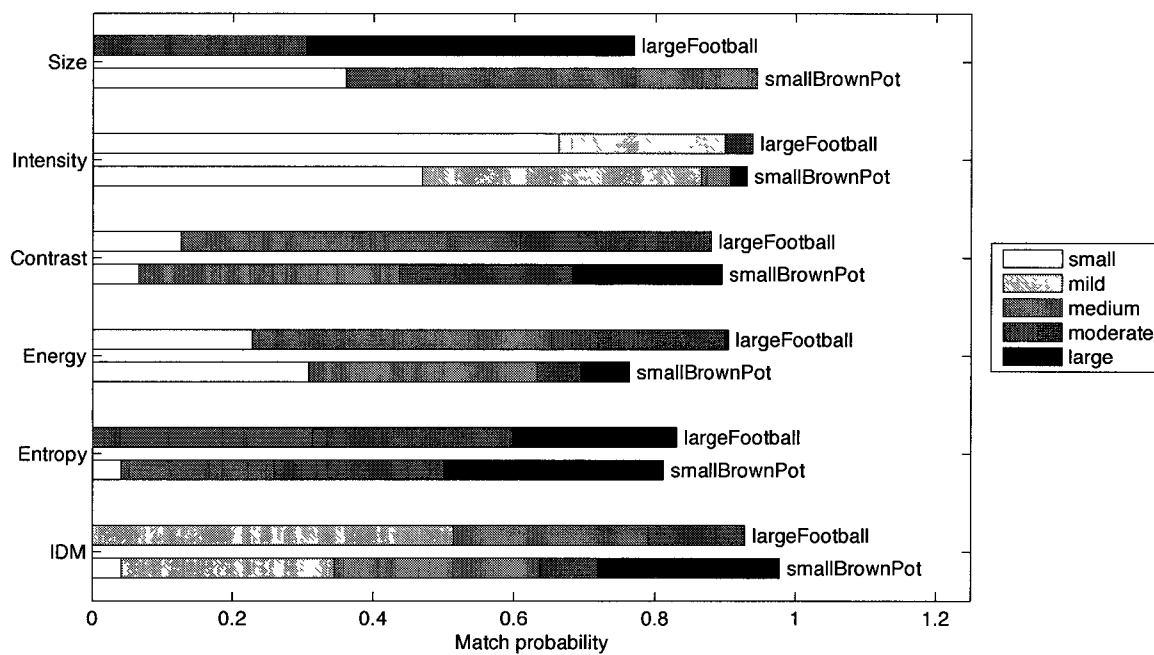
(b) White plastic ball vs. yellow plastic ball.



(c) Pink volleyball vs. white volleyball vs. large soccer ball vs. large football



(d) Medium pot vs. large pot.



(e) Large football vs. small brown pot.

Figure 5.10 – A comparison of the training results for groups of objects that shared similar characteristics. Note that the “mild” state only exists for intensity and IDM.

where this could be in issue were the small green pot and the small brown pot, the white plastic ball and the yellow plastic ball, and the four large balls—the pink volleyball, the white volleyball, large soccer ball and the large football (Figures 5.10(a), 5.10(b), and 5.10(c) respectively). The differences between the two small pots were minute for all six attributes. This leads to the prediction that upon observing either of these objects during the test trials, it will be impossible for the BN to differentiate between them and a split in probability is most likely. If differentiating these objects were critical, it is likely that at least one additional attribute is required (with colour being an obvious choice). The similarities between the two plastic balls were less profound. Although their size is nearly identical, the remaining five attributes have some stark differences. There is significant variation between the two objects in intensity, entropy and IDM. It is speculated that the reason for these differences is rooted in the performance of machine vision algorithms. Segmentation of the white plastic ball was much more reliable due to its contrast compared to the background, as seen in Figure 5.11. It is possible poor segmentation led to some of the differences in the state memberships. A comparison between the pink and white volleyballs shows similarities approaching the level of the two small pots. Interestingly, however, was that the greatest difference detected between these two balls was their size. In reality, their diameters are nearly equal. Once again, image segmentation was the likely source for this discrepancy. The pink volleyball had relatively poorer contrast compared to the background, and it's possible some of its figure was not segmented, resulting in its smaller perceived size. The texture of the large soccer ball differs somewhat from the two volleyballs, and its surface is more reflective. These differences were sufficient to be reflected in most of the attributes. However, there are some notable similarities between the large soccer ball and large football, particularly in size and energy.

A comparison of the training results for the large and medium pot yields some interesting results (Figure 5.10(d)). Perhaps the most interesting result is that the

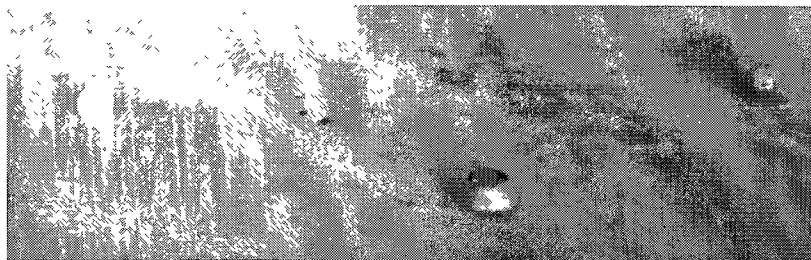


Figure 5.11 – A comparison of the white (far left) and yellow (far right) plastic balls as captured by the camera.

medium pot was perceived to be larger than the large pot. Although this is obviously an error, it is not an error in the sense that it will not necessarily negatively affect the identification algorithm. What is more important is consistency. If the large pot is consistently measured to be within the size membership values calculated during the training process, correct identification is still possible. Therefore, unsurprisingly, size is one of the prime attributes that will be used to differentiate between the two pots (along with intensity). However, surprisingly, a smaller size measurement will be indicative that the feature is the larger pot.

Finally, although not at the same level of similarity as some of the objects discussed above, there are some unexpected similarities between two seemingly unrelated objects: the large football and the small brown pot (Figure 5.10(e)). Obviously, a precise size measurement will vastly aid their differentiation; however, the state memberships for the remaining attributes are very similar. The colour-dependence of some of the attributes (colour in the grayscale sense) is likely to be the root of some of the similarities, as the true colour of the two objects is extremely similar. These observations, in conjunction to those made in the comparisons above, outline the importance of including attributes that are measurements of sufficiently different physical characteristics. It also highlights the importance of pursuing additional sensors alongside the stereo camera that can detect attributes not possible through

machine vision. For example, an infrared camera could be used to measure the specific heat capacity of rocks. Rocks that visually appear similar may have significantly different specific heat capacities depending on their composition [40].

5.2.2 Feature Identification Performance

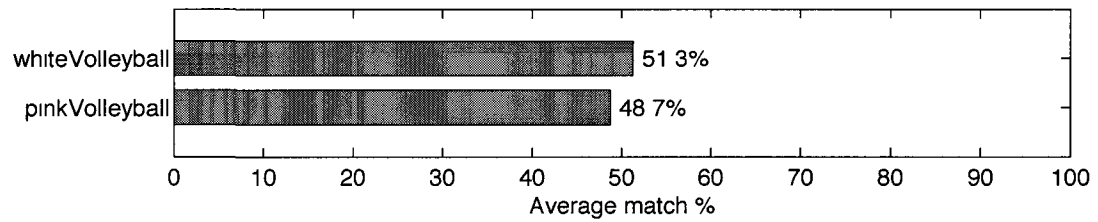
Over the course of the four test trials, 194 features were detected with machine vision and identified using the trained BN. It should be noted that false positives detected by the machine vision algorithm (e.g., an empty patch on the floor) are not included in this total. The purpose for this omission is that differentiation between features and benign terrain is primarily the responsibility of machine vision, which is outside the scope of this thesis. Theoretically, the BN could be also be used for this purpose by including non-features (e.g., sky, shadows, sand, etc.) in the training process. Because the fuzzy membership functions used in this thesis are not sum normal, false positives were often identified as “unknown”, but were also falsely identified as features if some of their attributes partially matched objects that the BN was trained to identify. Adding non-features to the training process would aide the detection process and add robustness to the system, but was not performed in this thesis.

Detailed identification results of all 13 objects are shown in Figure F.1 in Appendix F. Abbreviated identification results, showing only the two highest match percentages for each object, are shown in Figure 5.12. Overall, the BN identified the correct object as the most probable match in 9 of 13 cases. In two of the four mismatches, the second most probable match was the correct object. Among the best identified objects were the white volleyball, large football and white plastic ball (Figures 5.12(b), 5.12(g), and 5.12(h), respectively). Two factors that aide in the correct identification of objects are consistency and uniqueness. For consistency, the measured attributes of an object must be similar over several measurements and in different environments. This means test measurements will be more likely to resemble measurements taken

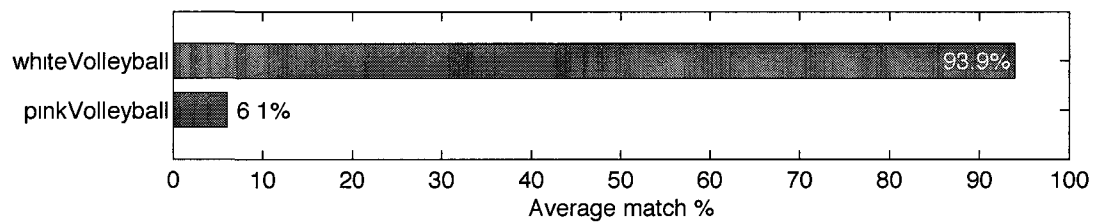
during the tuning process. Consistency is primarily a function of the quality of the sensors and algorithms used to determine attribute measurements. The measurements of the white volleyball and white plastic ball were quite consistent due to the ease of segmentation, as was discussed in Section 5.2.1. Inconsistencies can occur when objects are improperly segmented causing only part of the object to be measured, or a patch of the background is segmented as part of the object. Because these two objects were easily segmented due to their contrast compared to the background, the full object itself was usually detected and yielded consistent attribute measurements. For uniqueness, the state membership functions that describe an object must not closely resemble other objects. This reduces ambiguity between objects and allows for high probability matches. Although the large football had some similarities to the small brown pot (shown in Figure 5.10(e)), these were relatively weak similarities. The large football had the advantage of both consistency and uniqueness that contributed to its high identification probability (as shown in Table E.1(b), the intensity of the large football was particularly unique).

Among the poorest performing objects were the small football and the yellow plastic ball (Figures 5.12(f) and 5.12(i), respectively). The top two most probable matches were incorrect for both of these objects. In both cases, there were issues with consistency between the training data and the test data. To illustrate these inconsistencies, the training results are compared to the test results for a well identified object (the white volleyball) and a poorly identified object (the yellow plastic ball) in Figure 5.13. For the white volleyball, the training data and test data were fairly consistent; the same states were represented in every attribute. Conversely, there were some notable inconsistencies with the yellow plastic ball. Size, intensity, entropy and IDM all have states that occur in the training data but not the test data, or vice-versa.

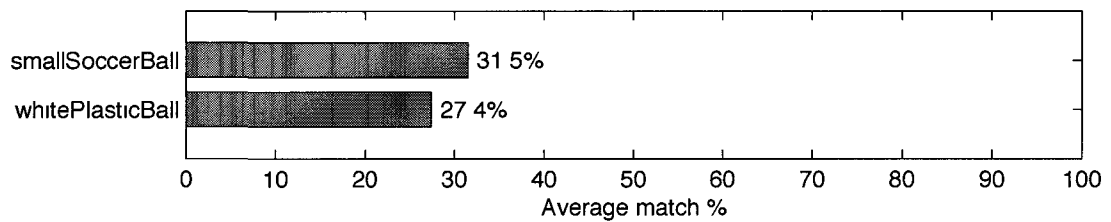
As was predicted when analyzing the training results (Section 5.2.1), there were



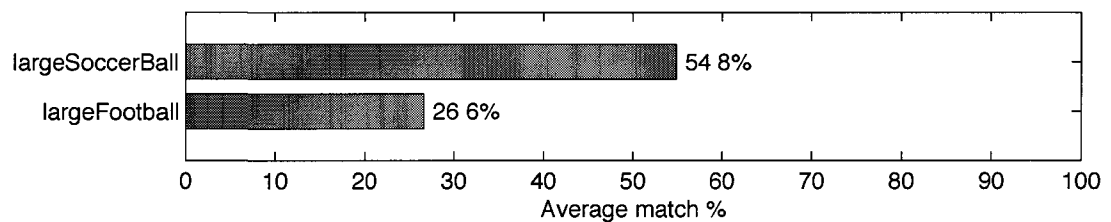
(a) Pink volleyball (4 detections).



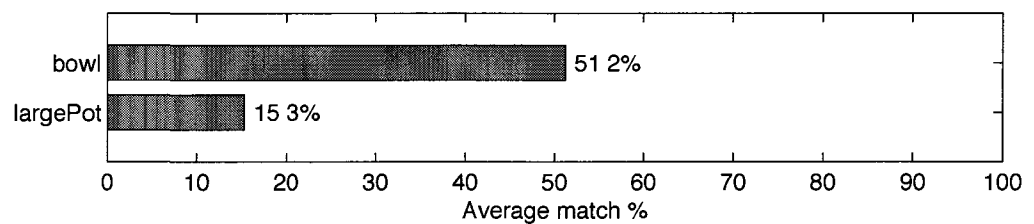
(b) White volleyball (3 detections).



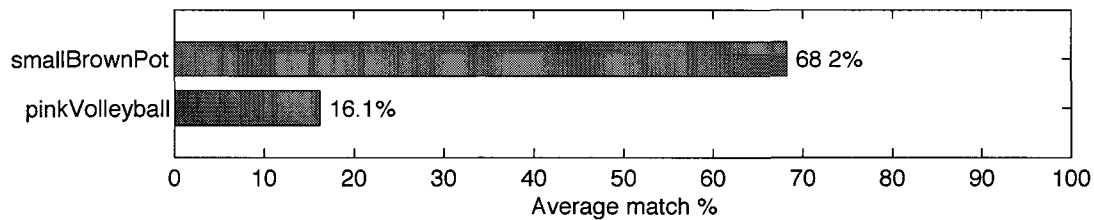
(c) Small soccer ball (3 detections).



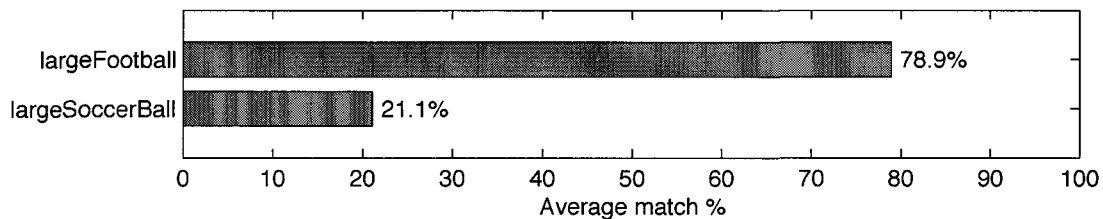
(d) Large soccer ball (11 detections).



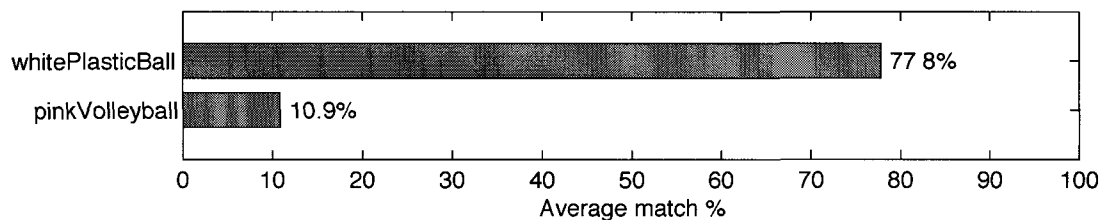
(e) Bowl (22 detections).



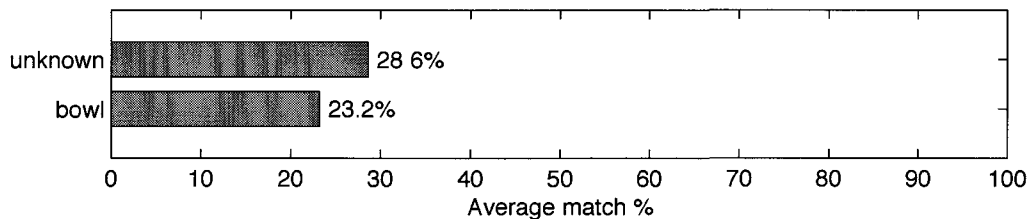
(f) Small football (4 detections).



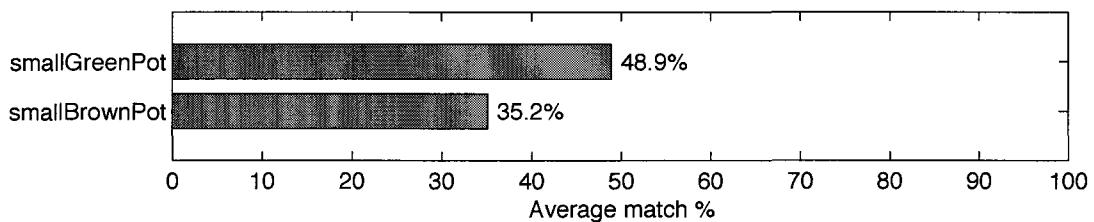
(g) Large football (5 detections).



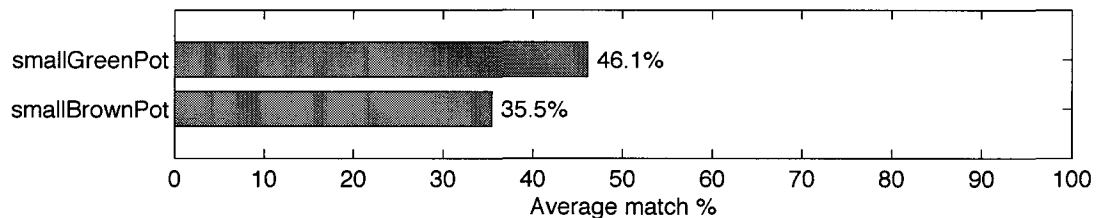
(h) White plastic ball (9 detections).



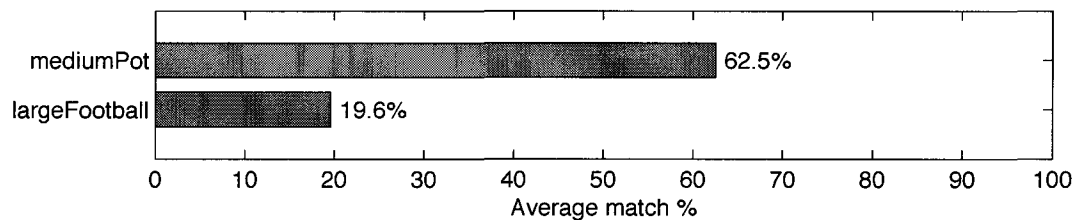
(i) Yellow plastic ball (7 detections).



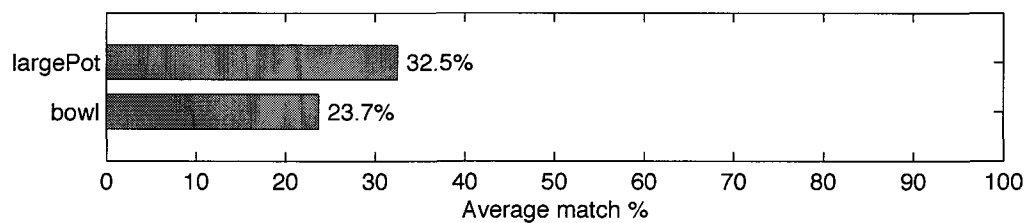
(j) Small green pot (51 detections).



(k) Small brown pot (51 detections).

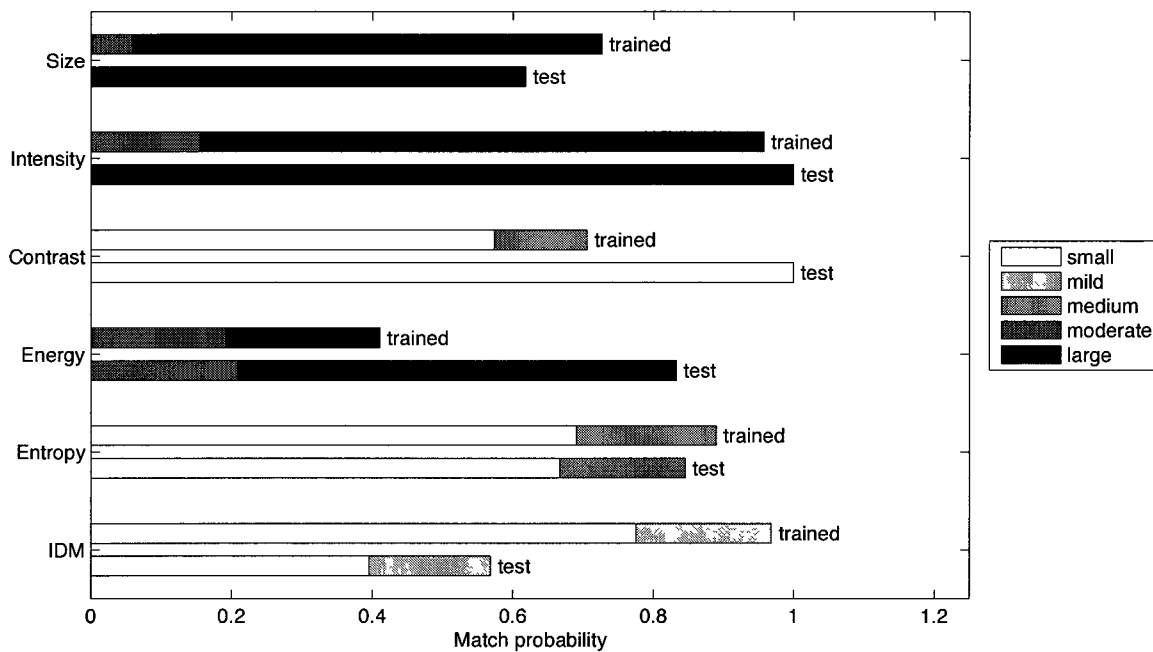


(l) Medium pot (12 detections).

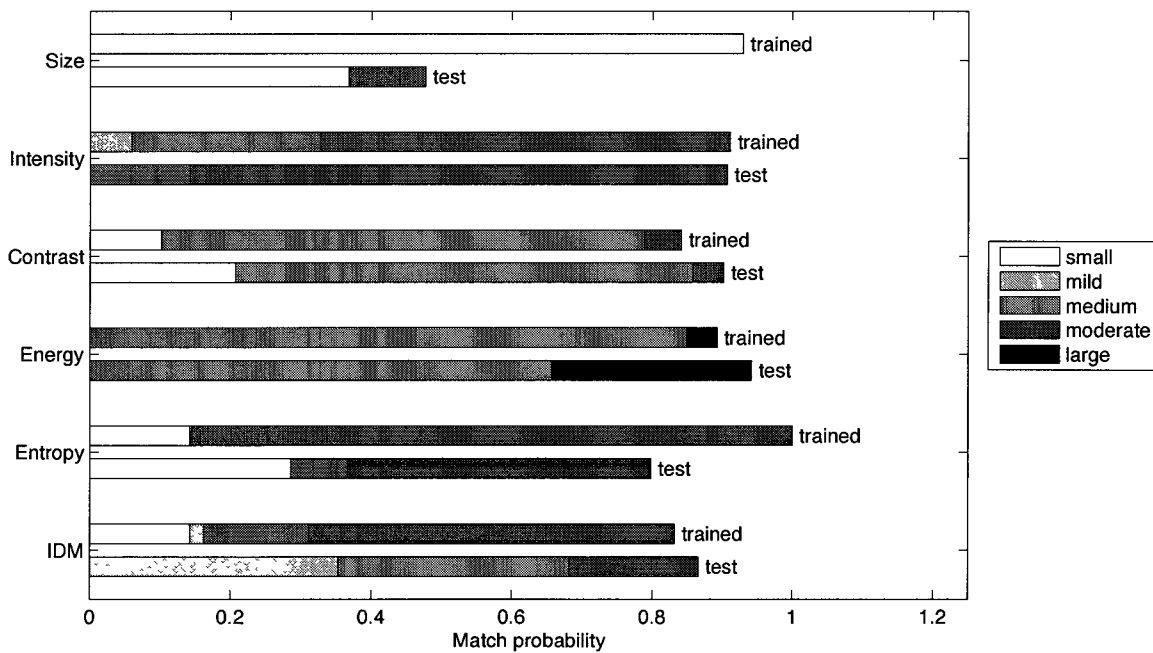


(m) Large pot (12 detections).

Figure 5.12 – The average match percentage by the Bayesian network for each of the 13 objects used in the laboratory tests. Results are abbreviated to show only the top two matches.



(a) White volleyball.



(b) Yellow plastic ball.

Figure 5.13 – A comparison of the training vs. test results for the white volleyball and the yellow plastic ball.

difficulties in distinguishing objects that had similar attribute state memberships (particularly the two small pots and the two volleyballs). As is shown in Figures 5.12(j) and 5.12(k), the small green pot and small brown pot were more-or-less not differentiable. In both cases, the average match probability of the small green pot was slightly higher than the small brown pot. A similar result occurred when identifying the pink volleyball and white volleyball (Figures 5.12(a) and 5.12(a), respectively). In this case, identifying the pink volleyball resulted in a near even match probability between the two. However, the white volleyball was correctly identified with an extremely high match percentage, with the pink volleyball making up the remainder. This was likely due to the exceptional consistency of the attribute measurements for the white volleyball, as was discussed above and illustrated in Figure 5.13(a).

Overall, it was found that uniqueness in attribute state membership and consistency of attribute measurements had a profound influence on the effectiveness of identifying objects with the BN. Uniqueness could be improved by using more attributes that describe additional physical characteristics of an object. This improvement may require additional sensors, such as cameras that can capture images outside of the visible spectrum. Consistency could be improved by enhancing the detection capabilities of the machine vision algorithms or using sensors that offer more repeatable measurements.

5.2.3 Feature Evaluation Performance

The results of the four test trials are shown in Figures 5.14 to 5.17. The poses of the rover from which images were captured are the arrowed boxes labelled in alphabetical order. Circles represents the locations features, with the name of the object that occupied each feature location printed above. Objects that were detected by the machine vision algorithm are marked with an “X” and have their benefit bias scores printed below. As it is outside the scope of this thesis, data association was

performed manually. When a feature was observed multiple times, the score it was assigned when its apparent size was greatest was used. This represented the occasion when the most visual information about the feature was available. The benefit bias equation that was used to calculate benefits scores used the general form reported in (3.11) using the importance weights outlined in Table 4.2. The purpose of this test was to evaluate the effectiveness of the benefit bias equation in determining the science value of observed features. As was indicated by the importance weights, the large soccer ball and white plastic ball had the greatest value. Moderate value was given to the white volleyball, small green pot and small brown pot. The pink volleyball, small soccer ball, small football, large football, yellow plastic ball and large pot were given moderately negative value, while the bowl and medium pot were considered the least valuable.

Trial 1

The results of the first trial are illustrated in Figure 5.14. The highest benefit score (3.78) was assigned to a large soccer ball, one of the most valued objects. The lowest benefit score (-5.36) was assigned to a small brown pot, which was assigned moderately positive value by the BBE. A detailed look at the top two and bottom two scored features is shown in Table 5.2. In this trial, the top scoring features correctly reflected objects assigned the highest importance weights. In addition, match percentages were exceptional, which in turn leads to a high normalized standard deviation. A concern is that the apparent size for the top scoring feature was relatively low, especially as it was one of the larger objects (large soccer ball). This either meant no up-close images were captured of the object or the feature was improperly segmented. However, the latter explanation is unlikely as a poor size measurement would have negatively affected the match percentage.

Table 5.2 – The top two and bottom two scored features from the first laboratory test trial.

| Score | True object | P | Σ | Top two matches | |
|-------|-------------|-------|----------|-----------------|--------------|
| | | | | First | Second |
| 3.78 | LSB | 0.061 | 0.865 | LSB (97.5 %) | LFB (2.5 %) |
| 1.58 | WPB | 0.136 | 0.962 | WPB (99.9 %) | WVB (0.1 %) |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| -1.49 | LPT | 0.157 | 0.978 | PVB (98.5 %) | WVB (1.5 %) |
| -5.36 | SBP | 0.516 | 0.503 | MPT (53.2 %) | LFB (46.8 %) |

It is interesting to note that both of the bottom two scoring features were misidentified. There were constant difficulties in identifying the large pot (as shown by the sporadic match percentage distribution in Figure F.1(m) in Appendix F), but the misidentification of the small brown pot was somewhat surprising. The generous match percentage assigned to the large football was speculated to be possible from the training results (a comparison is shown in Figure 5.10(e)). However, according to the accumulated average match percentages for all the test trials (Figure F.1(k)), this result was somewhat an anomaly.

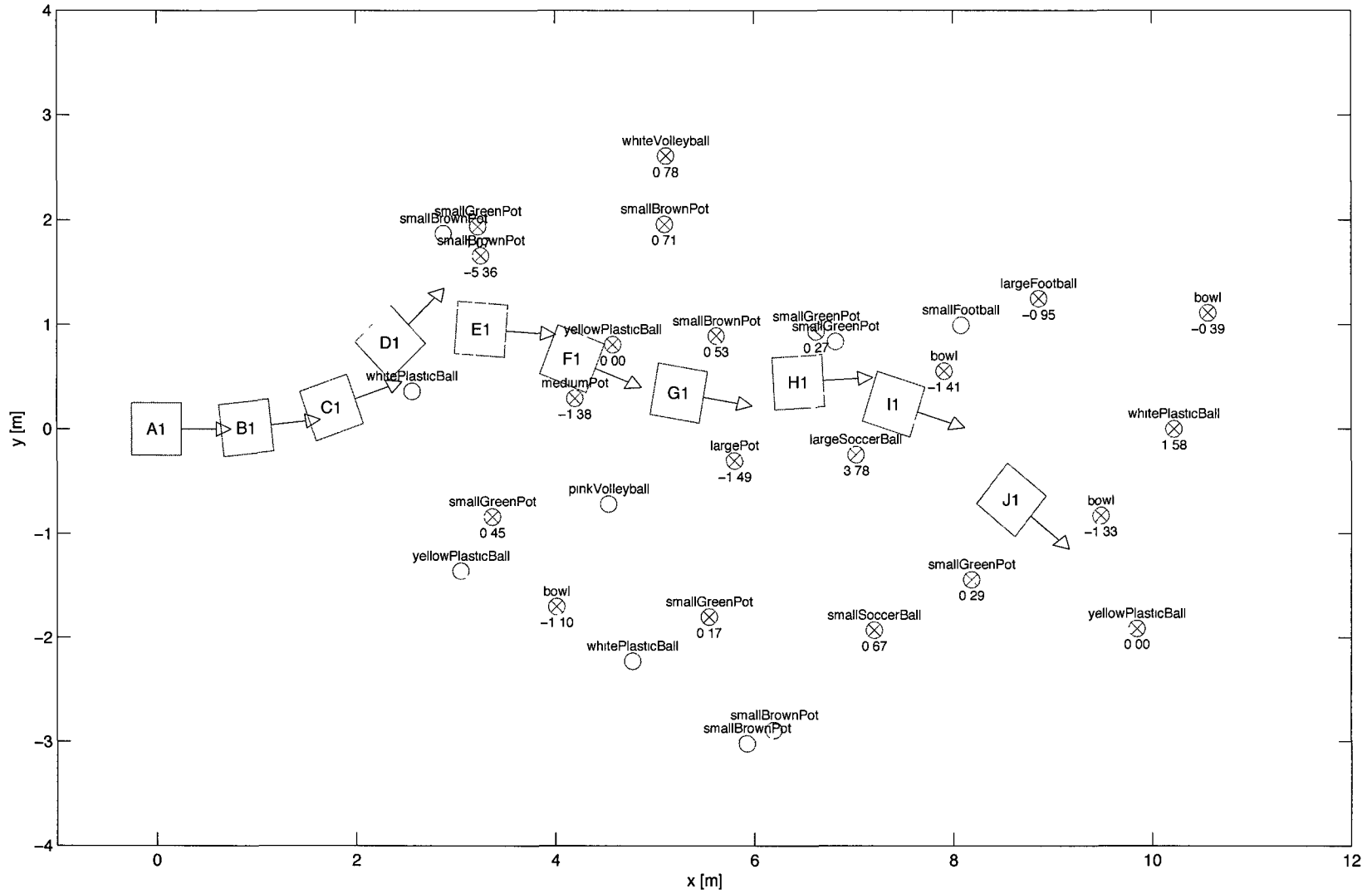


Figure 5.14 – Path and scoring results for the first laboratory trial.

Trial 2

The results of the second trial are illustrated in Figure 5.15. The highest benefit score (3.29) was once again assigned to a large soccer ball, one of the most valued objects. The lowest benefit score (-5.93) was assigned to a medium pot, which was considered one of the least valuable objects by the BBE. A detailed look at the top two and bottom two scored features is shown in Table 5.3. Much like the first trial,

Table 5.3 – The top two and bottom two scored features from the second laboratory test trial.

| Score | True object | P | Σ | Top two matches | |
|----------|-------------|----------|----------|-----------------|--------------|
| | | | | First | Second |
| 3.29 | LSB | 0.122 | 0.946 | LSB (96.3 %) | LFB (3.7 %) |
| 1.06 | SGP | 0.181 | 0.585 | SGP (67.5 %) | SBP (32.5 %) |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| -4.26 | YPB | 0.318 | 0.530 | BWL (77.3 %) | YPB (15.4 %) |
| -5.93 | MPT | 0.618 | 0.571 | LFB (65.9 %) | MPT (34.1 %) |

the top scoring feature was a highly-valued object and was assigned a high match percentage. The second best feature was a moderately-valued small green pot. It is interesting to note that this score was achieved despite a relatively low normalized standard deviation. As was shown in Figure 5.10(a), it would be difficult for the BN to distinguish between the two small pots, which was the case here. However, as both small pots were given moderate importance weights in the BBE, the inherently lower normalized standard deviation is the only handicap caused by their difficult differentiation. If they had drastically different importance weights, their assigned scores would be unreliable.

There were no surprises for lowest two scoring features, which both had the lowest importance weights. In both cases, the true object was assigned the second highest match percentage. However, the yellow plastic ball has potential to cause problems

with the scoring algorithm. As was discovered during the training process (Section 5.2.1), improper segmentation may negatively affect measuring its attributes. Its average match percentages all the tests were inconsistent (F.1(i)), which reflects this hypothesis. It is possible that a high match percentage for a valued object could cause a high score to be falsely assigned.

Trial 3

The results of the third trial are illustrated in Figure 5.16. The highest benefit score (16.63) was a medium pot, one of the least-valued objects. The lowest benefit score (−2.59) was assigned to a bowl, which was also one of the least valuable objects according to the BBE. A detailed look at the top two and bottom two scored features is shown in Table 5.4. The exceptionally high benefit score given to one of the lowest

Table 5.4 – The top two and bottom two scored features from the third laboratory test trial.

| Score | True object | P | Σ | Top two matches | |
|-------|-------------|-------|----------|-----------------|--------------|
| | | | | First | Second |
| 16.63 | MPT | 0.627 | 0.937 | LSB (95.8 %) | LFB (4.2 %) |
| 4.04 | WVB | 0.483 | 0.928 | WVB (95.1 %) | PVB (4.9 %) |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| -2.53 | SBP | 0.243 | 0.503 | MPT (53.3 %) | LFB (46.7 %) |
| -2.59 | BWL | 0.222 | 0.591 | MPT (77.2 %) | LSB (14.1 %) |

valued objects was caused by a number of factors. First, it was observed to have a relatively high apparent size and the match probabilities calculated by the BN yielded a relatively high normalized standard deviation. Next, an exceptionally high match percentage was assigned to a highly-valued object, the large soccer ball. This result underlines the importance of consistency, which was discussed in Section 5.2.2. This is especially relevant when an inconsistently measured object can be mistaken for an object with a drastically different importance weight. As a result, improvements to both the machine vision algorithms and the BN are required to improve consistency. The latter could possibly be improved via optimization, which is discussed in Section 5.2.4.

The bottom two scoring features were both misidentified. It is possible that the relatively low apparent sizes made it difficult for identification, as less information

was available to the machine vision algorithms. It should also be noted that these minimum scores were higher than the minimum scores from the other trials. This means that no low-valued object was observed up-close and positively identified, as the apparent size and normalized standard deviation tends to move benefit scores to zero.

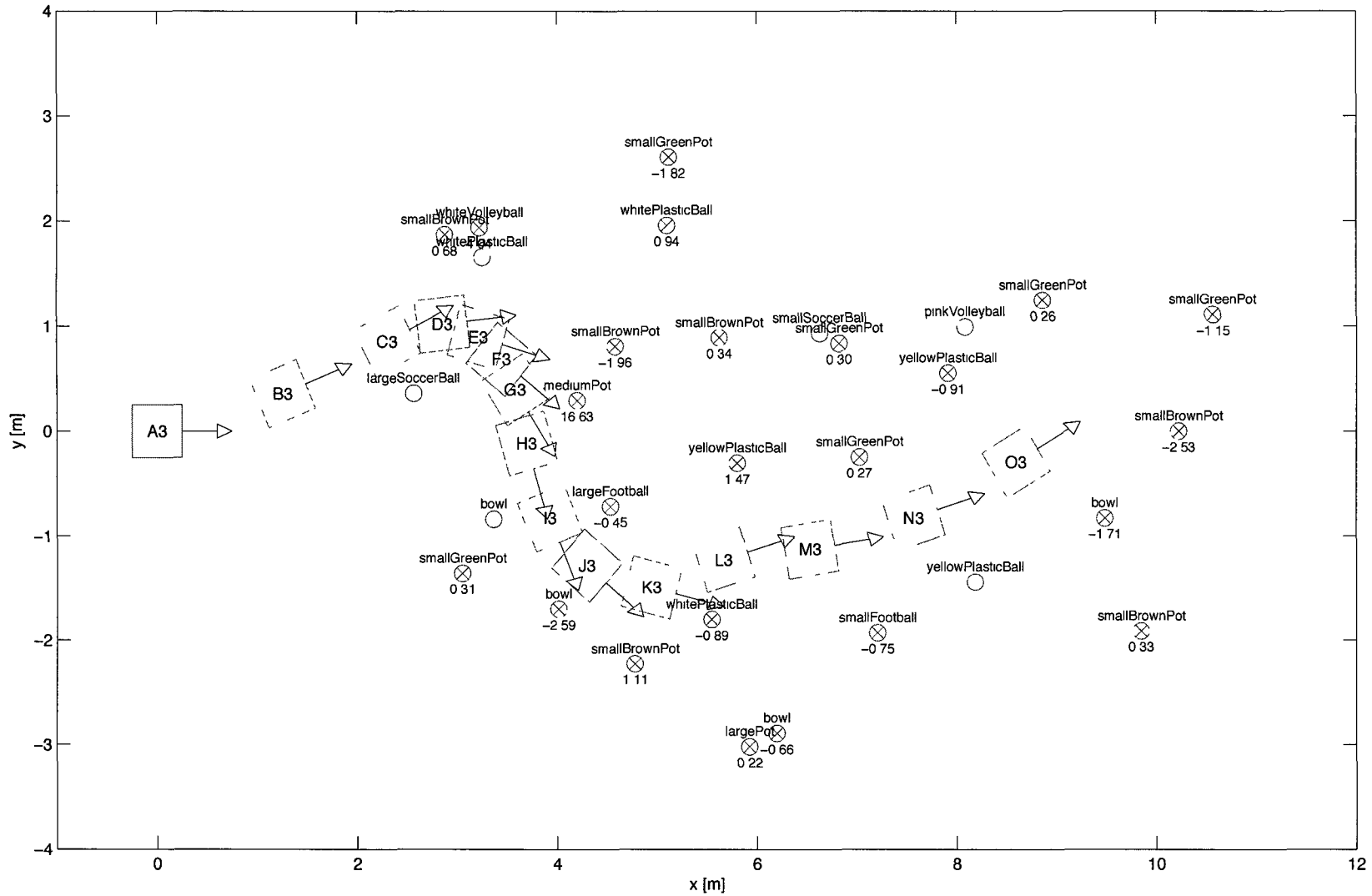


Figure 5.16 – Path and scoring results for the third laboratory trial.

Trial 4

The results of the fourth trial are illustrated in Figure 5.17. The highest benefit score (4.74) was a white plastic ball, one of the highest-valued objects. The lowest benefit score (−8.80) was assigned to a medium pot, one of the least valuable objects according to the BBE. A detailed look at the top two and bottom two scored features is shown in Table 5.5. The highest scoring feature was correctly identified with an

Table 5.5 – The top two and bottom two scored features from the fourth laboratory test trial.

| Score | True object | P | Σ | Top two matches | |
|-------|-------------|-------|----------|-----------------|--------------|
| | | | | First | Second |
| 4.74 | WPB | 0.222 | 0.771 | WPB (95.5 %) | BWL (2.4 %) |
| 4.01 | LFB | 0.142 | 0.967 | LSB (97.8 %) | LFB (2.2 %) |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| -5.16 | SGP | 0.513 | 0.500 | MPT (50.7 %) | LFB (49.3 %) |
| -8.80 | MPT | 0.752 | 0.576 | MPT (75.9 %) | LSB (12.2 %) |

exceptional match percentage. The second best feature was misidentified. A very low apparent size likely influenced this misidentification, much like the bottom two features in the third trial. It should be noted that the large football and large soccer ball were commonly misidentified as each other, as was shown in Figures 5.12(d) and 5.12(g). Similarities in some of their attribute state memberships are notable in the training results in Figure 5.10(c). However, unlike the two small pots, two significantly different importance weights were assigned to these objects, causing their assigned scores to be less reliable.

The lowest scoring feature provides a good example of a well-identified but low-valued object, in this case resulting in the lowest score of the four trials. The second lowest scoring feature was a common misidentification of a small pot; a similar misidentification occurred in the first and third trials. In all these cases, the highest

matching object was the medium pot, one of the lowest-valued objects.

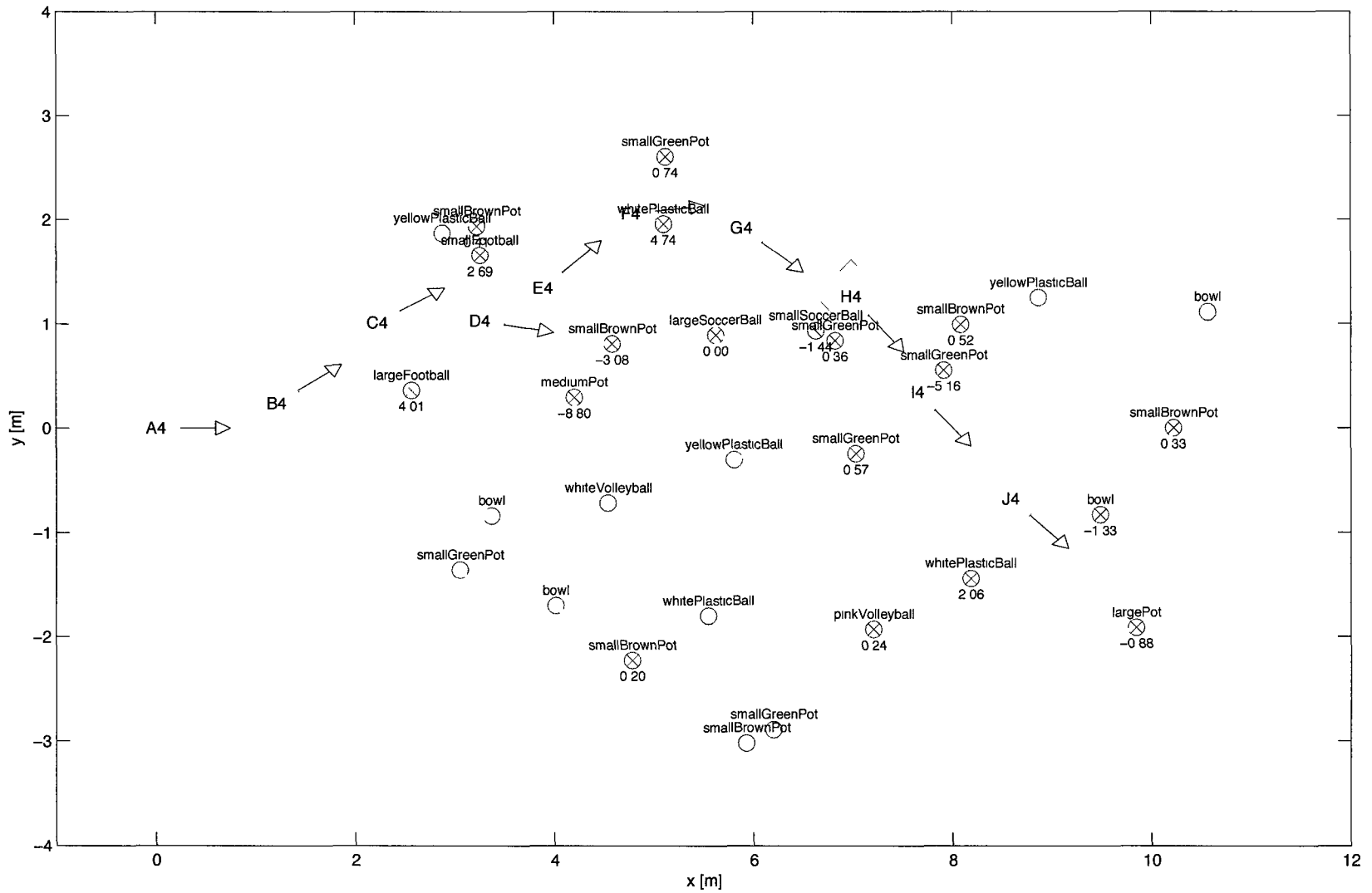


Figure 5.17 – Path and scoring results for the fourth laboratory trial.

5.2.4 Possible Improvements

By considering both the training and test results, some overall observations can be made concerning the effectiveness of using a BN for object identification and the validity of the scoring system. Although improvements are possible by simply improving the quality and consistency of the data provided to the BN by the sensors, there are changes to the methods and algorithms used in this thesis that may improve the results. The following points address these possible improvements:

Optimize the directed acyclic graph

Rather than using a directed acyclic graph (DAG) specified by the operator to represent causality (as was done in this thesis), such a graph can be generated with empirical evidence. The utility of such an approach becomes apparent once the number of nodes and relationships increase and become more complex. Methods have been developed to autonomously determine the causal relationships between variables using data sets [41]. Applying such an algorithm may reveal unforeseen relationships between variables thought to be independent. It is possible that additional causal relationships among the variables used in this thesis may improve the DAG that was used, and in turn, improve identification results.

Optimize the fuzzy state membership functions

As was outlined in Section 3.4.2, the fuzzy state membership functions were determined manually through observation of histograms of the results of the training data. A variety of methods have been shown to improve the performance of fuzzy logic through optimization of the membership function parameters. Simon improved sum normal membership functions through parameter optimization with a Kalman filter [21]. Figueiredo and Gomide achieved a similar result using neural networks [42]. A sensitivity study describing the magnitude

of influence changing the fuzzy parameters would have on the overall system may reveal the necessity for an optimization technique.

Improve the training procedure

At the time of data collection, it was unknown how effective the machine vision algorithms would be at detecting the test objects. As a result, many objects were left undetected in many of the training images, resulting in a smaller training set than anticipated. This also led to an uneven distribution as to the number of times each object was detected. Therefore, difficulties in identifying some objects were compounded by the fact that training data for that object was sparse. To remedy this issue, a larger training set could be used or detection could be performed manually.

Improve the benefit bias equation

The results of the test trials revealed some weaknesses of the current form of the benefit bias equation (BBE). In the third trial, a misidentification led to a high score being assigned to a lowly-valued object. A similar result occurred in the fourth trial. One thing that was common in these two cases was that the feature had a small apparent size. In the current form of the BBE, a linear relationship between apparent size and benefit score is used. However, it is unknown whether a non-linear relationship would be more effective in adjusting the score based on apparent size. A study could be performed on the relationship between apparent size and identification performance (which may be object dependent) to optimize the BBE.

5.2.5 Transitioning to Natural Features

The laboratory experiments used artificial objects to validate the identification and evaluation algorithms presented in Sections 3.4 and 3.5, respectively. However, for

true validation for use in an autonomous science system, natural features (especially those that are analogous to Martian rocks) must be used. The success of such a validation depends greatly on the selection of the attributes to be measured. The texture-based attributes used in the laboratory experiments would likely remain valuable in natural feature identification. Size, however, would be much less valuable as rocks of the same mineral type can sometimes drastically differ in size from one rock to the next. It is recommended that additional or more diverse texture measurements are included in the attribute set for natural feature detection. Additionally, measuring additional attributes such as albedo and shape (as was included by Castano *et al* in Section 2.5.2) would likely enhance the accuracy of identification.

Chapter 6

Conclusion

The over-arching goal of this thesis was to develop and test an autonomous science system. The development was broken down into four smaller goals, whose current statuses are discussed in Section 6.1. Achievements and contributions that were made in the field of study as a result of the work performed for this thesis are presented in Section 6.2. Finally, as the development of an autonomous science system is an ambitious undertaking that requires inputs from several fields, temporal constraints prevented some work from being performed. As a result, Section 6.3 provides suggestions for the direction of future research and development.

6.1 Status of Goals

In the onset of the research and development of work presented in this thesis, four goals were outlined which are presented in Section 1.3. In this section, the status of these goals is discussed; namely, the extent in which they were or were not achieved.

“The rover should be capable of identifying (i.e., not just sorting) several different objects via the machine vision, image processing and machine learning techniques.”

There is an important distinction between identifying objects and identifying attributes. Identifying objects implies that attributes have been identified and measured, and that information is further used to identify an object *based* on the attributes. The purpose of this goal was to perform identification with a machine learning technique. To an extent, the results in Section 5.2 achieved this result. Issues with the identification algorithm were identified and improvements that address these issues were suggested.

“The machine learning technique should be expandable and tuneable to include new feature types.”

The nature of the Bayesian network training process that was used for the laboratory tests in this thesis (Section 3.4.3) allows for expansion to include additional features. If the identification of new features is desired, a data set of the attributes of that feature can be used to train the identification algorithm.

“A map of identified features should be built with sufficient accuracy that a rover can safely plan efficient paths from one point in the map to another.”

The framework necessary to achieve this goal was either developed (e.g., feature localization) or identified (e.g., SLAM). Successful mapping of the features was performed in simulation, but assumptions were made concerning localization of the rover. As a result, this goal was only partially fulfilled and the work required for total fulfilment is outlined in Section 6.3.

“The rover should be able to identify features, plan paths, and drive to goals without any intervention from an operator.”

Identifying features (Section 3.4), path-planning (Section 2.1) and driving to goals (Section 2.4.3) were integrated into a single autonomous system and successfully demonstrated in simulation. Additionally, identifying features was performed in the

laboratory tests. Achieving the full autonomous system outside of simulation would require SLAM and integration of the implemented controller on a real robotic platform.

6.2 Achievements and Contributions

Autonomous science is a relatively new, but active, field of study. A literature review of the most recent developments (Section 2.5) revealed that there remains many undeveloped or underdeveloped methods left for investigation. Some of these areas are examined in this thesis. The autonomous science algorithm that was developed is rooted in using machine learning for the identification and evaluation of objects rather than classification by attributes alone. Although using probabilistic classification to evaluate observations is not novel, the evaluation algorithm used in this thesis also considered the quality of the observation and the certainty of the classification. Also, feature evaluations were tied directly into the path-planning of the rover, allowing for detour considerations when determining the scientific value of a feature. To summarize, this work performed for this thesis has made the following achievements or contributions:

- (i) The development, implementation and experimental validation of an object identification technique combining machine vision and machine learning in the form of a BN. The specific combination of the attributes derived through image processing and the identification of objects with a BN has shown to be effective in identifying objects and providing a quantitative confidence in this identification.
- (ii) An investigation into the effectiveness of using a scout rover to enhance the capabilities of a prime rover using autonomous science. The utility of using a

scout rover to provide *a priori* information to a better-equipped prime rover was found to positively effect the science return per unit distance travelled in simulation. To date, no known previous efforts have combined the prospect of performing autonomous science with a scout rover. The framework was established to continue this research with field tests.

- (iii) A novel feature evaluation approach that combines the probabilistic identification of a feature with the quality of observations, the certainty of identification, and the reachability of the feature. Possible improvements to this technique were outlined to further enhance its effectiveness.
- (iv) The creation of a modular, generalized simulation platform to test autonomous science. The simulator facilitates modularity of many of the algorithms required for autonomous science such as: rover localization, feature localization, path-planning, and rover control. In addition, the structure of BNs is easily configurable, as is the training process from real or simulated data.

6.3 Future Work

This section outlines some possible directions for future work that could extend the usability of the existing architecture. Specific algorithmic improvements are summarized based on the results of the simulator (Section 5.1.4) and the laboratory tests (Section 5.2.4). As a result, this section focuses on the work required to transform the current algorithms into a full-fledged autonomous science system. The following steps are suggested:

- (i) The path-planning algorithm used in this thesis employed occupancy grids—a simplification of cost maps. Rather than simply determining the occupancy of vertices, assigning the connections between vertices a traversal cost would add

detail that can be used by the path-planner to create better paths. For example, costs could be functions of the height of rocks, the depth of crevasses, or the grade of slopes. As was described in Section 2.1, most graph-based path-planners are already equipped to handle this enhancement. In an unrelated project, a graph-based path-planning algorithm was tested in a laboratory environment that constructed a cost map using a laser scanner. Although not essential, laser scanners can provide accurate raw measurements of the local environment of the rover and an investigation their usefulness as part of the autonomous science system is recommended.

- (ii) As was noted in Section 1.2.1, rover localization was assumed throughout this thesis. Obviously, a true implementation of an autonomous science system would require accurate rover localization to create and plan paths in feature maps. The feature localization that is performed in this thesis forms one half of SLAM, which includes rover localization. It is recommended that a full SLAM algorithm is used as part of the autonomous science algorithm. Another requirement is data association, which ensures repeated measurements of a common feature can be identified and combined. Traditionally, data association uses the estimated location of a feature and compares it to a set of measurements. However, as the autonomous science algorithm provides much more information about many of the observed features, it is conceivable that an augmented science-influenced data association could be developed. This algorithm could use the output of the autonomous science algorithm to aide in the association of features.
- (iii) A collection of artificial objects were used as the feature set in the laboratory tests for this thesis. If the final goal of the autonomous science system is to be useful for a planetary mission, natural features should be used, particularly those whose physical characteristics are not unique to Earth. Natural features

(e.g., rocks) are likely a greater challenge to detect and identify, as they have fewer visually distinguishing characteristics and the same rock types are less consistent from rock to rock. Additionally, there is the possibility that dust or other effects of the environment will make identification more difficult. Here, the importance of measuring a wide array of attributes (possibly with multiple sensors) becomes more important such that uniqueness among different rock types can be established.

List of References

- [1] D. Thompson, S. Niekum, T. Smith, and D. Wettergreen, "Automatic detection and classification of features of geologic interest," in *IEEE Aerospace Conference*, pp. 366–377, 2005.
- [2] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 2, pp. 99–110, 2006.
- [3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II state of the art," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [4] J. Guo and L. Liu, "A study of improvement of D* algorithms for mobile robot path planning in partial unknown environments," *Kybernetes*, vol. 39, pp. 935–945, 2010.
- [5] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global path planning on board the Mars Exploration Rovers," in *2007 IEEE Aerospace Conference*, pp. 1–11, 2007.
- [6] M. Baerg, J. Beck, S. Hulme, S. Lievense, E. Tozzi, M. Viotti, and G. Webster, "Jet Propulsion Laboratory: Mars Science Laboratory." Retrieved April 2011 from <http://mars.jpl.nasa.gov/msl/mission/technology/insituexploration/planetarymobility/>.
- [7] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [8] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," in *Proceedings of the AAAI Conference of Artificial Intelligence*, pp. 1177–1183, 2007.

- [9] Jet Propulsion Laboratory, "Photojournal." Retrieved April 2011 from <http://photojournal.jpl.nasa.gov/catalog/PIA02406>.
- [10] D. Ferguson and A. Stentz, "The Field D* algorithm for improved path planning and replanning in uniform and non-uniform cost environments," Tech. Rep. CMU-RI-TR-05-19, Robotics Institute, Carnegie Mellon University, June 2005.
- [11] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [12] A. Nash, S. Koenig, and M. Likhachev, "Incremental Phi*: incremental any-angle path planning on grids," in *Proceedings of the 21st international joint conference on Artificial intelligence*, (San Francisco, CA, USA), pp. 1824–1830, Morgan Kaufmann Publishers Inc., 2009.
- [13] A. Onisko, P. J. F. Lucas, and M. J. Druzdzal, "Comparison of rule-based and Bayesian network approaches in medical diagnostic systems," in *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, AIME '01, pp. 283–292, Springer-Verlag, 2001.
- [14] B. Bornstein, R. Castano, M. Gilmore, M. Merrill, and J. Greenwood, "Creation and testing of an artificial neural network based carbonate detector for Mars rovers," in *2005 IEEE Aerospace Conference*, pp. 378–384, 2005.
- [15] R. Zhang and A. J. Bivens, "Comparing the use of Bayesian networks and neural networks in response time modeling for service-oriented systems," in *Proceedings of the 2007 workshop on Service-oriented computing performance: aspects, issues, and approaches*, SOCP '07, (New York, NY, USA), pp. 67–74, ACM, 2007.
- [16] A. Darwiche, *Modeling and reasoning with Bayesian networks*. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2009.
- [17] D. J. Spiegelhalter, R. C. G. Franklin, and K. Bull, "Assessment, criticism and improvement of imprecise subjective probabilities for a medical expert system," in *Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '89, pp. 285–294, North-Holland Publishing Co., 1990.
- [18] L. A. Zadeh, "Fuzzy sets," *Information Control*, vol. 8, pp. 338–353, 1965.
- [19] G. Shafer, "Perspectives on the theory and practice of belief functions," *International Journal of Approximate Reasoning*, vol. 4, pp. 323–362, 1990.

- [20] D. Koks and S. Challa, “An introduction to bayesian and dempster-shafer data fusion,” tech. rep., Defence Science and Technology Organisation, 2003.
- [21] D. Simon, “Sum normal optimization of fuzzy membership functions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, pp. 363–384, August 2002.
- [22] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, “The analysis of a simple k-means clustering algorithm,” Tech. Rep. CAR-TR-937, Center for Automation Research, University of Maryland, January 2000.
- [23] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, pp. 610–621, November 1973.
- [24] M. Bak, N. K. Poulsen, and O. Ravn, “Path following mobile robot in the presence of velocity constraints,” tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, 2001.
- [25] Guy Webster, “NASA Mars Rover Getting Smarter as it Gets Older.” Retrieved May 2011 from <http://www.jpl.nasa.gov/news/news.cfm?release=2010-094>.
- [26] T. Smith, S. Niekum, D. Thompson, and D. Wettergreen, “Concepts for science autonomy during robotic traverse and survey,” in *IEEE Aerospace Conference Proceedings*, pp. 347–355, 2005.
- [27] R. Castano, T. Estlin, R. C. Anderson, D. M. Gaines, A. Castano, B. Bornstein, C. Chouinard, and M. Judd, “Oasis: Onboard autonomous science investigation system for opportunistic rover science,” *Journal of Field Robotics*, vol. 24, pp. 379–397, May 2007.
- [28] M. Woods, A. Shaw, D. Barnes, D. Price, D. Long, and D. Pullan, “Autonomous science for an exomars rover-like mission,” *Journal of Field Robotics*, vol. 26, no. 4, pp. 358–390, 2009.
- [29] D. Pullan, *Analogue Studies for In Situ Surface Planetary Exploration*. PhD thesis, University of Leicester, 2008.
- [30] D. Barnes, S. Pugh, and L. Tyler, “Autonomous science target identification and acquisition (ASTIA) for planetary exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3329–3335, October 2009.

- [31] P. R. Gazis and T. Roush, "Autonomous identification of carbonates using near-ir reflectance spectra during the february 1999 marsokhod field tests," *Journal of Geophysical Research*, vol. 106, no. E4, pp. 7765–7773, 2001.
- [32] M. Wagner, D. Apostolopoulos, K. Shillcutt, B. Shamah, R. Simmons, and W. Whittaker, "The science autonomy system of the nomad robot," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1742–1749, 2001.
- [33] R. E. Arvidson, C. Acton, D. Blaney, J. Bowman, S. Kim, G. Klingelhofer, J. Marshall, C. Niebur, J. Plescia, R. S. Saunders, and C. T. Ulmer, "Rocky 7 prototype mars rover field geology experiments 1. lavic lake and sunshine volcanic field, california," *Journal of Geophysical Research*, vol. 103, pp. 22671–22688, September 1998.
- [34] L. Pedersen, "Autonomous characterization of unknown environments," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 277–284, 2001.
- [35] T. Roush, "Essential autonomous science inference on rovers (EASIR)," in *IEEE Aerospace Conference Proceedings*, vol. 2, pp. 790–800, 2004.
- [36] Jet Propulsion Laboratory, "Mars Pathfinder Frequently Asked Questions." Retrieved August 2011 from http://mars.jpl.nasa.gov/MPF/rover/faqs_sojourner.html.
- [37] M. Montemerlo and S. Thrun, *FastSLAM*, vol. 27. Springer Tracts in Advanced Robotics, 2007.
- [38] S. Portigliotti, M. Dumontel, M. Capuano, and L. Lorenzoni, "Landing site targeting and constraints for EXOMARS 2016 mission," in *International Planetary Probe Workshop*, 2010.
- [39] Point Grey Research, Inc., "Stereo Vision Camera Catalogue." Retrieved May 2011 from http://www.ptgrey.com/products/Point_Grey_stereo_catalog.pdf.
- [40] D. Waples and J. Waples, "A review and evaluation of specific heat capacities of rocks, minerals, and subsurface fluids. part 1: Minerals and nonporous rocks," *Natural Resources Research*, vol. 13, pp. 97–122, 2004.
- [41] G. Rebane and J. Pearl, "The recovery of causal poly-trees from statistical data," in *3rd Workshop on Uncertainty in AI*, pp. 175–182, 1987.

- [42] M. Figueiredo and F. Gomide, "Design of fuzzy systems using neurofuzzy networks," *IEEE Transactions on Neural Networks*, vol. 10, no. 4, pp. 815–827, 1999.

Appendix A

Simulation Parameters

The simulation parameters are outlined in Table A.1. These parameters were kept constant for each trial of the simulated tests. One exception is the importance weights, which were each assigned a random value in the uniform distribution $U(-1, 1)$.

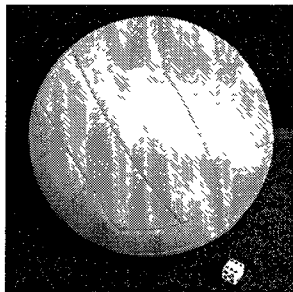
Table A.1 – The simulation parameters.

| Parameter | Value | Section |
|--|----------------------------|---------|
| Camera field-of-view | 66° | 3.2.2 |
| Camera range | 5 m | 3.2.2 |
| Desired forward velocity | 2.2 cm/s (80 m/h) | 2.4.3 |
| Detour coefficient (D) | 1.5 | 3.5.2 |
| Distance between waypoints | 0.25 m | 3.1 |
| Dynamic benefit score threshold multiplier (m_D) | $\frac{\mu + \sigma}{\mu}$ | 3.5.2 |
| Importance weights (a, b, c, d) | $U(-1, 1)$ | 3.5.1 |
| Maximum dimension of rover footprint (h) | 0.5 m | 3.3.1 |
| Maximum rover angular velocity | 3.63 °/s | 2.4.3 |
| Maximum rover forward velocity | 2.2 cm/s (80 m/h) | 2.4.3 |
| Minimum rover angular velocity | -3.63 °/s | 2.4.3 |
| Minimum rover forward velocity | 0 cm/s | 2.4.3 |
| Occupancy grid resolution | 0.15 m | 3.3.1 |
| Scale constant (K) | 100 | 3.5.1 |
| Science sensor radius (r_{sen}) | 0.50 m | 3.3.2 |
| Static benefit score threshold ($S_{b,stat}$) | 0 | 3.5.2 |
| Time step (T) | 0.25 s | 4.1.1 |

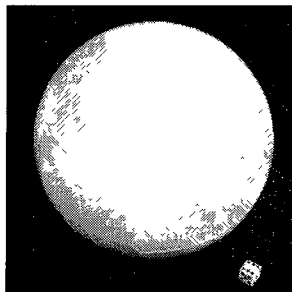
Appendix B

Laboratory Test Features

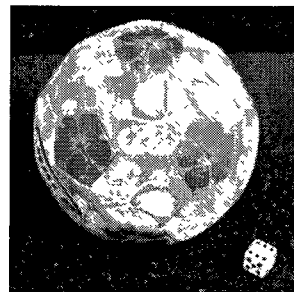
Figure B.1 shows the 13 unique objects that were used in the laboratory tests. Note that the white plastic ball and the yellow plastic ball (Figures B.1(h) and B.1(i), respectively) differ only in colour, as do the small green pot and small brown pot (Figures B.1(j) and B.1(k), respectively). The standard die in the figures has an edge length of 16 mm.



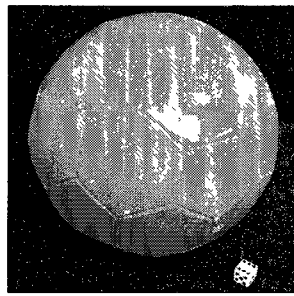
(a) Pink volleyball.



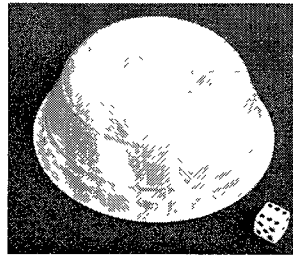
(b) White volleyball.



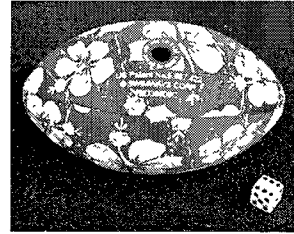
(c) Small soccer ball.



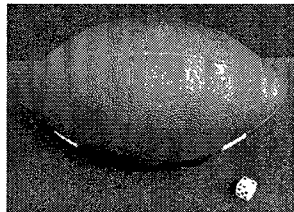
(d) Large soccer ball.



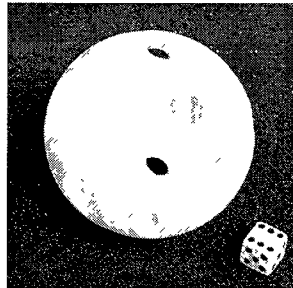
(e) Bowl.



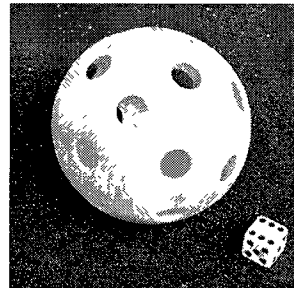
(f) Small football.



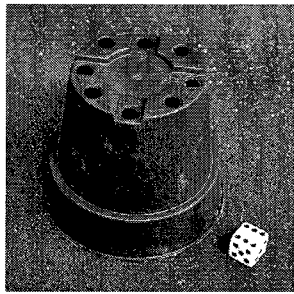
(g) Large football.



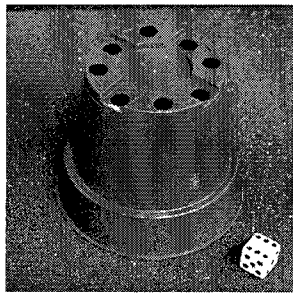
(h) White plastic ball.



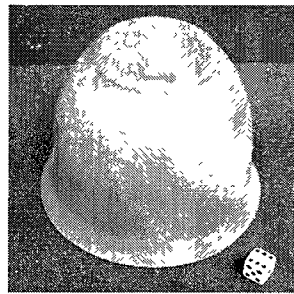
(i) Yellow plastic ball.



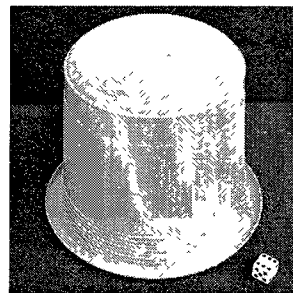
(j) Small green pot.



(k) Small brown pot.



(l) Medium pot.



(m) Large pot.

Figure B.1 – The objects used for the laboratory tests.

Appendix C

Laboratory Attribute Parameters

The name and trapezoidal parameters of the states of each of the six attributes is shown in Table C.1. These parameters were selected via observation of the training data histograms for each attribute, as was described in Section 3.4.2.

Table C.1 – The trapezoidal parameters and the names of the states of the attributes for the laboratory tests.

| Attribute | State | Trapezoidal Parameters |
|-----------|----------|------------------------------|
| Size | Small | {20.0, 48.8, 63.8, 73.9} |
| | Medium | {63.8, 78.9, 88.9, 98.9} |
| | Moderate | {93.9, 134.0, 169.0, 194.0} |
| | Large | {184.0, 224.0, 239.0, 300.0} |
| Intensity | Low | {0.0, 49.9, 67.9, 81.5} |
| | Mild | {72.5, 82.0, 109.0, 136.0} |
| | Medium | {118.0, 136.0, 149.0, 158.0} |
| | Moderate | {149.0, 172.0, 181.0, 212.0} |
| | High | {194.0, 230.0, 275.0, 300.0} |
| Contrast | Low | {0.000, 0.024, 0.071, 0.119} |
| | Medium | {0.071, 0.119, 0.221, 0.261} |
| | Moderate | {0.238, 0.356, 0.689, 0.760} |
| | High | {0.713, 0.832, 1.160, 1.400} |
| Energy | Low | {0.000, 0.130, 0.241, 0.315} |
| | Medium | {0.223, 0.352, 0.426, 0.556} |
| | Moderate | {0.482, 0.519, 0.556, 0.722} |
| | High | {0.648, 0.815, 0.852, 1.000} |
| Entropy | Low | {0.00, 0.10, 0.52, 0.70} |
| | Medium | {0.58, 0.82, 1.05, 1.22} |
| | Moderate | {1.11, 1.28, 1.50, 1.80} |
| | High | {1.71, 2.00, 2.63, 3.00} |
| IDM | Low | {0.000, 0.008, 0.024, 0.048} |
| | Mild | {0.032, 0.089, 0.121, 0.153} |
| | Medium | {0.121, 0.140, 0.170, 0.198} |
| | Moderate | {0.177, 0.213, 0.265, 0.290} |
| | High | {0.273, 0.298, 0.330, 0.450} |

Appendix D

Simulation Test Results

The full detailed results of the simulation tests are shown in Table D.1. Each trial used the same map and simulation parameters for the two scenarios. Listed are the number of features that were observed during the trial, the detour distance (L_D), the benefit score of the feature that was visited for up-close observation (S_b), and the score-to-detour ratio for that trial $\left(\frac{S_b}{L_D}\right)$.

Table D.1 – The full detailed results of the simulations tests comparing the prime rover solo (P) to the prime/scout rover pairing (P+S).

| Trial | Feat. found | | L_D [m] | | S_b | | $\frac{S_b}{L_D}$ | |
|-------|-------------|-----|-----------|-------|-------|-------|-------------------|------|
| | P | P+S | P | P+S | P | P+S | P | P+S |
| 1 | 35 | 27 | 11.09 | 10.75 | 3.93 | 4.42 | 0.35 | 0.41 |
| 2 | 22 | 24 | 10.62 | 10.37 | 30.73 | 22.68 | 2.89 | 2.19 |
| 3 | 11 | 13 | 10.00 | 9.92 | 3.41 | 7.85 | 0.34 | 0.79 |
| 4 | 21 | 24 | 11.46 | 10.53 | 17.65 | 39.30 | 1.54 | 3.73 |
| 5 | 34 | 38 | 14.02 | 13.95 | 19.64 | 15.80 | 1.40 | 1.13 |
| 6 | 25 | 25 | 11.07 | 10.78 | 2.70 | 6.39 | 0.24 | 0.59 |
| 7 | 23 | 25 | 11.47 | 11.59 | 6.07 | 14.27 | 0.53 | 1.23 |
| 8 | 15 | 18 | 10.74 | 10.78 | 10.45 | 12.47 | 0.97 | 1.16 |
| 9 | 44 | 29 | 19.31 | 11.34 | 11.85 | 15.78 | 0.61 | 1.39 |
| 10 | 15 | 21 | 10.87 | 10.91 | 18.23 | 16.36 | 1.68 | 1.50 |
| 11 | 28 | 29 | 12.09 | 13.88 | 6.11 | 9.75 | 0.51 | 0.70 |
| 12 | 13 | 13 | 10.23 | 10.24 | 8.21 | 16.68 | 0.80 | 1.63 |
| 13 | 15 | 18 | 11.51 | 10.66 | 24.42 | 9.60 | 2.12 | 0.90 |
| 14 | 12 | 9 | 9.89 | 9.85 | 23.45 | 2.67 | 2.37 | 0.27 |
| 15 | 16 | 23 | 12.22 | 11.39 | 4.17 | 15.32 | 0.34 | 1.35 |
| 16 | 39 | 38 | 14.34 | 12.25 | 21.81 | 22.94 | 1.52 | 1.87 |
| 17 | 18 | 19 | 10.03 | 10.02 | 16.65 | 11.73 | 1.66 | 1.17 |
| 18 | 12 | 13 | 10.17 | 10.16 | 1.77 | 13.53 | 0.17 | 1.33 |
| 19 | 16 | 21 | 10.29 | 9.96 | 21.42 | 15.02 | 2.08 | 1.51 |
| 20 | 39 | 32 | 14.70 | 10.84 | 3.96 | 5.79 | 0.27 | 0.53 |
| 21 | 15 | 15 | 9.97 | 9.99 | 27.06 | 33.15 | 2.72 | 3.32 |
| 22 | 20 | 20 | 11.86 | 10.48 | 23.01 | 24.35 | 1.94 | 2.32 |
| 23 | 15 | 19 | 10.19 | 10.15 | 5.74 | 2.80 | 0.56 | 0.28 |
| 24 | 23 | 27 | 11.02 | 11.15 | 6.25 | 35.03 | 0.57 | 3.14 |
| 25 | 26 | 27 | 10.07 | 10.02 | 8.04 | 7.52 | 0.80 | 0.75 |
| 26 | 12 | 13 | 10.15 | 10.08 | 21.47 | 27.10 | 2.11 | 2.69 |
| 27 | 15 | 14 | 10.09 | 9.95 | 3.93 | 17.75 | 0.39 | 1.78 |
| 28 | 34 | 41 | 12.96 | 18.77 | 19.38 | 23.64 | 1.50 | 1.26 |
| 29 | 16 | 22 | 11.47 | 11.47 | 15.49 | 20.69 | 1.35 | 1.80 |
| 30 | 17 | 13 | 10.57 | 10.63 | 21.81 | 25.79 | 2.06 | 2.43 |

Appendix E

Laboratory Training Results

The training results for each of the six attributes are shown in Table E.1. These results were obtained using the Bayesian network training technique described in Section 3.4.3.

Table E.1 – The Bayesian network training results for the six attributes in the laboratory tests.

| (a) Size | | | | |
|---------------------|--------|--------|----------|--------|
| Object | Size | | | |
| | Small | Medium | Moderate | Large |
| Pink volleyball | 0 | 0 | 0.8701 | 0.0173 |
| White volleyball | 0 | 0 | 0.0593 | 0.6670 |
| Small soccer ball | 0.6836 | 0 | 0 | 0 |
| Large soccer ball | 0 | 0 | 0.0761 | 0.6696 |
| Bowl | 0.9192 | 0.0137 | 0 | 0 |
| Small football | 0.6113 | 0 | 0 | 0 |
| Large football | 0 | 0 | 0.3040 | 0.4653 |
| White plastic ball | 0.8806 | 0.0213 | 0 | 0 |
| Yellow plastic ball | 0.9299 | 0 | 0 | 0 |
| Small green pot | 0.3313 | 0.5640 | 0 | 0 |
| Small brown pot | 0.3606 | 0.5849 | 0 | 0 |
| Medium pot | 0 | 0 | 0.8705 | 0 |
| Large pot | 0.1418 | 0.4119 | 0.1225 | 0 |

(b) Intensity

| Object | Intensity | | | | |
|---------------------|-----------|--------|--------|----------|--------|
| | Small | Mild | Medium | Moderate | Large |
| Pink volleyball | 0 | 0 | 0 | 0.0890 | 0.8743 |
| White volleyball | 0 | 0 | 0 | 0.1556 | 0.8021 |
| Small soccer ball | 0 | 0 | 0 | 0.1456 | 0.7310 |
| Large soccer ball | 0 | 0.9304 | 0 | 0 | 0 |
| Bowl | 0 | 0.2727 | 0.4545 | 0.1727 | 0.0445 |
| Small football | 0 | 0 | 0.0877 | 0.1396 | 0.7077 |
| Large football | 0.6621 | 0.2374 | 0 | 0.0392 | 0 |
| White plastic ball | 0 | 0 | 0 | 0.0720 | 0.9090 |
| Yellow plastic ball | 0 | 0.0601 | 0.2667 | 0.5848 | 0 |
| Small green pot | 0.5116 | 0.3652 | 0.0380 | 0 | 0 |
| Small brown pot | 0.4682 | 0.3969 | 0.0417 | 0 | 0.0240 |
| Medium pot | 0.1163 | 0.8162 | 0.0142 | 0 | 0 |
| Large pot | 0 | 0.2049 | 0.2916 | 0.2843 | 0.0126 |

(c) Contrast

| Object | Contrast | | | |
|---------------------|----------|--------|----------|--------|
| | Small | Medium | Moderate | Large |
| Pink volleyball | 0.8779 | 0.0602 | 0 | 0 |
| White volleyball | 0.5729 | 0.1325 | 0 | 0 |
| Small soccer ball | 0.0395 | 0.2462 | 0.7069 | 0 |
| Large soccer ball | 0.1060 | 0.7803 | 0.0367 | 0 |
| Bowl | 0.1981 | 0.7110 | 0.0272 | 0 |
| Small football | 0 | 0.5000 | 0.4107 | 0 |
| Large football | 0.1258 | 0.4823 | 0.2716 | 0 |
| White plastic ball | 0.4302 | 0.5698 | 0 | 0 |
| Yellow plastic ball | 0.1022 | 0.6861 | 0.0526 | 0 |
| Small green pot | 0.0224 | 0.3811 | 0.2246 | 0.2124 |
| Small brown pot | 0.0663 | 0.3702 | 0.2456 | 0.2135 |
| Medium pot | 0.1139 | 0.4992 | 0.2678 | 0 |
| Large pot | 0.1144 | 0.7466 | 0.0369 | 0 |

(d) Energy

| Object | Energy | | | |
|---------------------|--------|--------|----------|--------|
| | Small | Medium | Moderate | Large |
| Pink volleyball | 0 | 0.0647 | 0.1765 | 0.4705 |
| White volleyball | 0 | 0.0023 | 0.1896 | 0.2190 |
| Small soccer ball | 0.2341 | 0.4602 | 0.0093 | 0.1969 |
| Large soccer ball | 0.0425 | 0.7392 | 0.1667 | 0 |
| Bowl | 0.0330 | 0.7223 | 0.1711 | 0 |
| Small football | 0.3750 | 0.3654 | 0.1455 | 0 |
| Large football | 0.2269 | 0.4255 | 0.2477 | 0.0047 |
| White plastic ball | 0 | 0.1003 | 0.1436 | 0.6908 |
| Yellow plastic ball | 0 | 0.8494 | 0 | 0.0431 |
| Small green pot | 0.3777 | 0.2888 | 0.2015 | 0.0026 |
| Small brown pot | 0.3077 | 0.3248 | 0.0622 | 0.0685 |
| Medium pot | 0.1400 | 0.6178 | 0.1228 | 0.0130 |
| Large pot | 0.1248 | 0.7897 | 0 | 0 |

(e) Entropy

| Object | Entropy | | | |
|---------------------|---------|--------|----------|--------|
| | Small | Medium | Moderate | Large |
| Pink volleyball | 0.6825 | 0.2854 | 0 | 0 |
| White volleyball | 0.6900 | 0.2000 | 0 | 0 |
| Small soccer ball | 0.2532 | 0 | 0.3865 | 0.2390 |
| Large soccer ball | 0 | 0.1673 | 0.6119 | 0.0100 |
| Bowl | 0 | 0.2669 | 0.6542 | 0 |
| Small football | 0 | 0.4652 | 0.1250 | 0.3214 |
| Large football | 0 | 0.3127 | 0.2836 | 0.2341 |
| White plastic ball | 0.7336 | 0.1691 | 0.0515 | 0 |
| Yellow plastic ball | 0.1429 | 0 | 0.8571 | 0 |
| Small green pot | 0 | 0.1604 | 0.2769 | 0.3541 |
| Small brown pot | 0.0417 | 0.2169 | 0.2411 | 0.3121 |
| Medium pot | 0 | 0.2528 | 0.4447 | 0.0907 |
| Large pot | 0 | 0.3081 | 0.5428 | 0 |

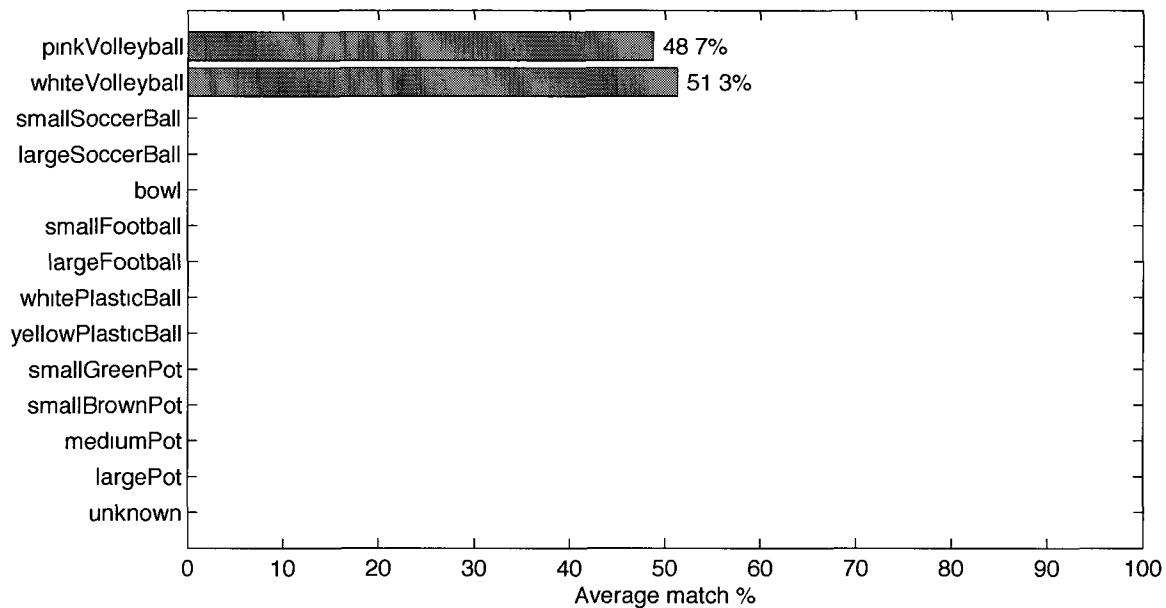
(f) IDM

| Object | IDM | | | | |
|---------------------|--------|--------|--------|----------|--------|
| | Small | Mild | Medium | Moderate | Large |
| Pink volleyball | 0.4342 | 0.2885 | 0 | 0 | 0 |
| White volleyball | 0.7748 | 0.1939 | 0 | 0 | 0 |
| Small soccer ball | 0 | 0.2356 | 0.2744 | 0.3712 | 0 |
| Large soccer ball | 0 | 0.3721 | 0.6526 | 0.0029 | 0 |
| Bowl | 0 | 0.3302 | 0.4922 | 0.1169 | 0 |
| Small football | 0 | 0.1002 | 0.3625 | 0.1212 | 0.3259 |
| Large football | 0 | 0.5134 | 0.2765 | 0.1388 | 0 |
| White plastic ball | 0.1136 | 0.4557 | 0 | 0 | 0 |
| Yellow plastic ball | 0.1429 | 0.0198 | 0.1482 | 0.5210 | 0 |
| Small green pot | 0 | 0.3342 | 0.2093 | 0.1441 | 0.2872 |
| Small brown pot | 0.0417 | 0.3025 | 0.2927 | 0.0820 | 0.2587 |
| Medium pot | 0 | 0.4965 | 0.3719 | 0.0828 | 0 |
| Large pot | 0 | 0.4280 | 0.4107 | 0.1428 | 0 |

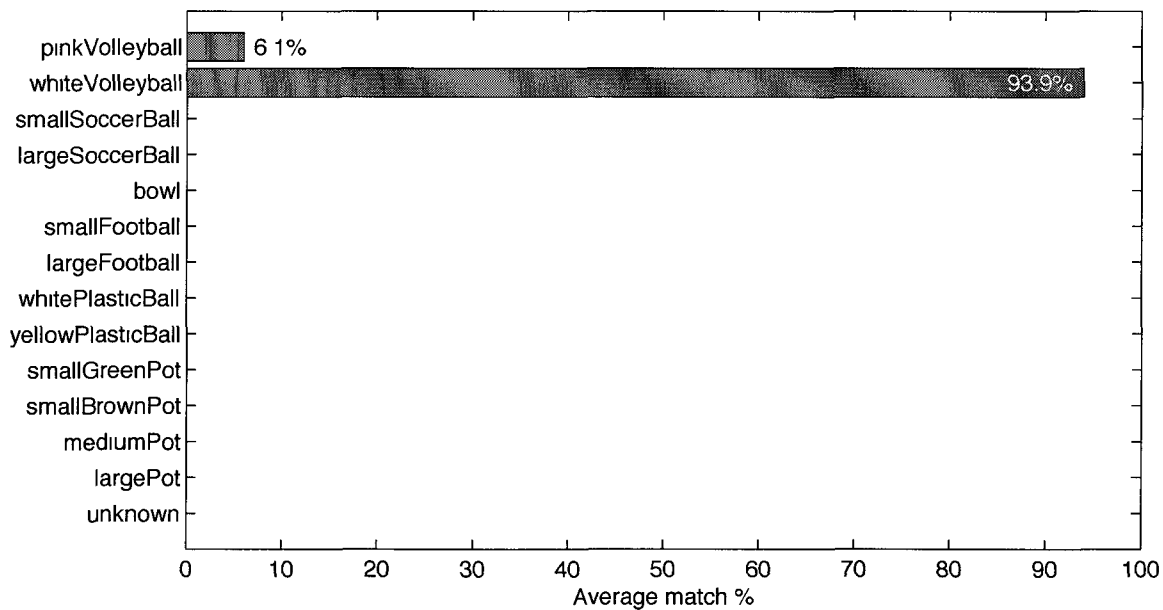
Appendix F

Laboratory Identification Results

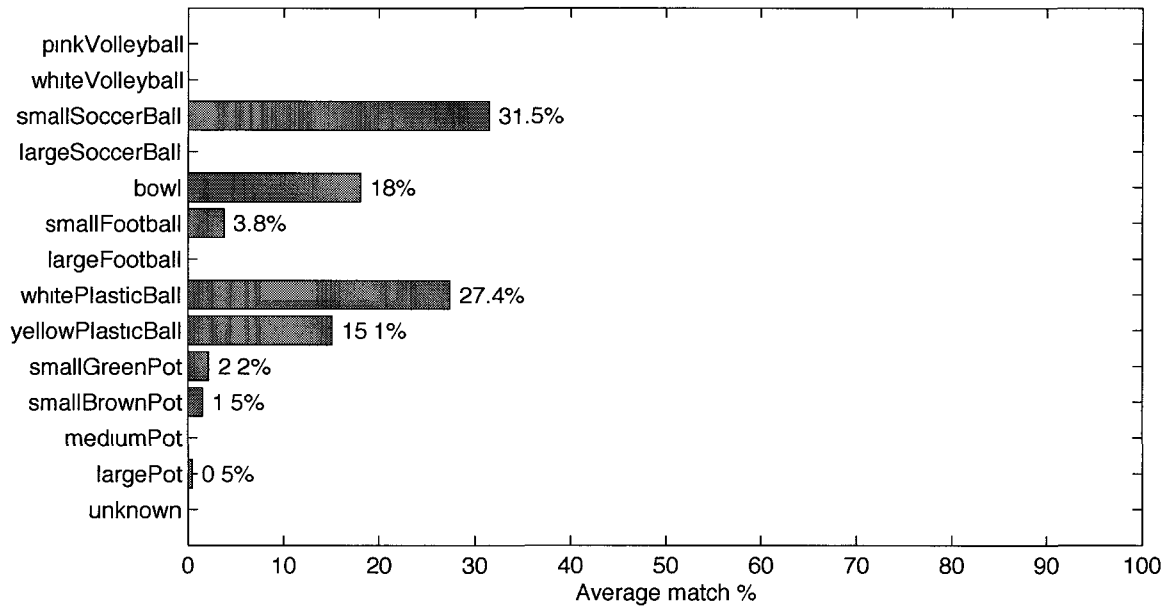
Detailed identification results from the laboratory tests are shown in Figure F.1. The subfigures illustrate the average match percentages for each of the 13 objects.



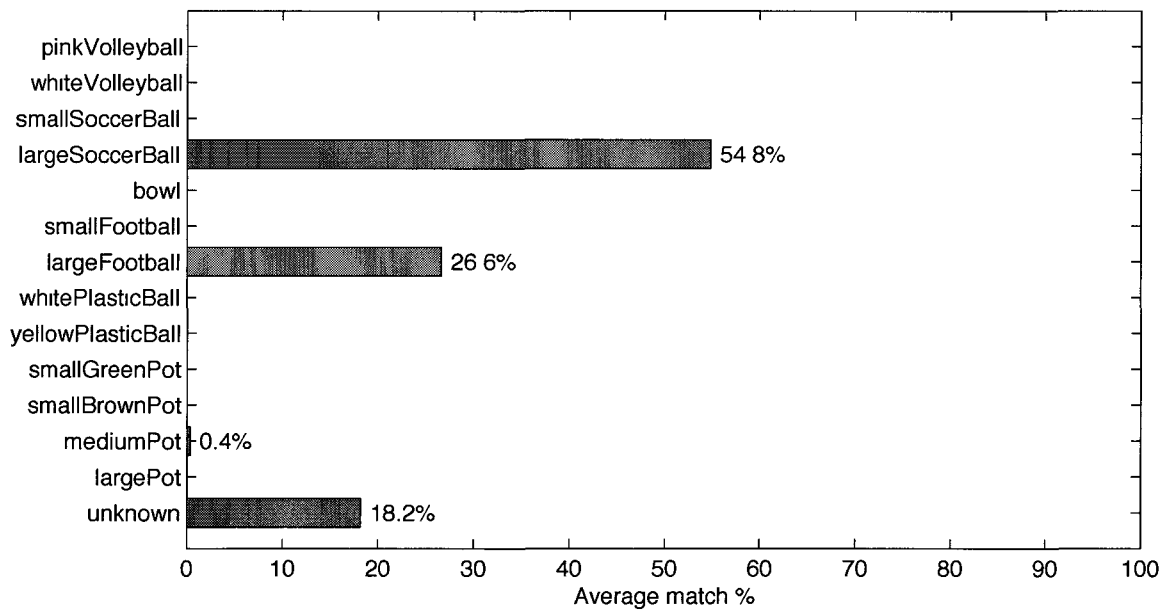
(a) Pink volleyball (4 detections).



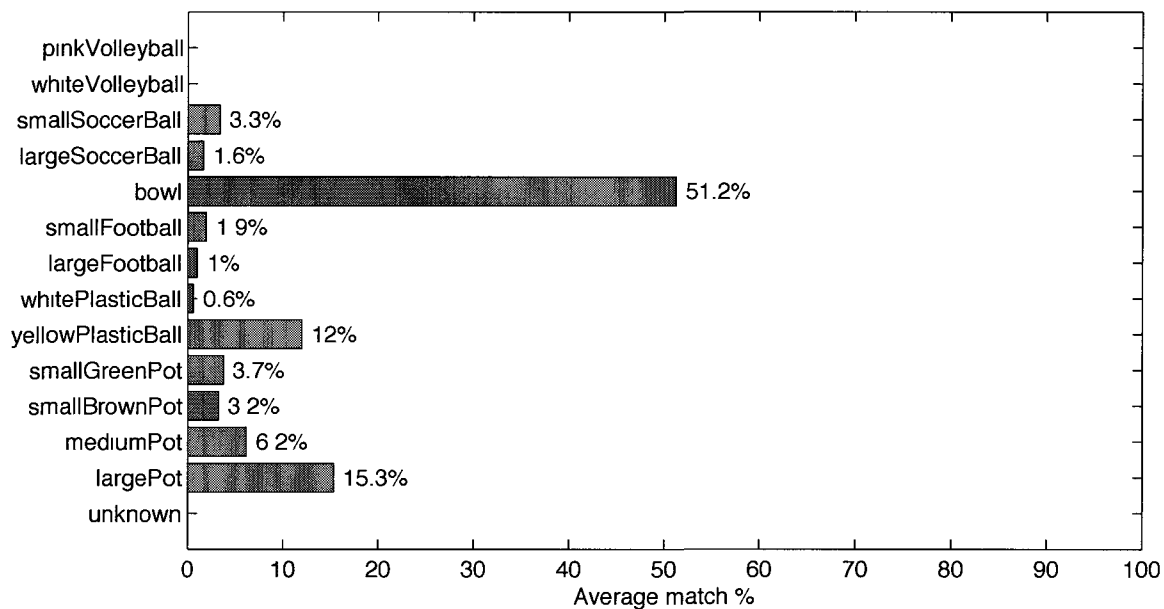
(b) White volleyball (3 detections).



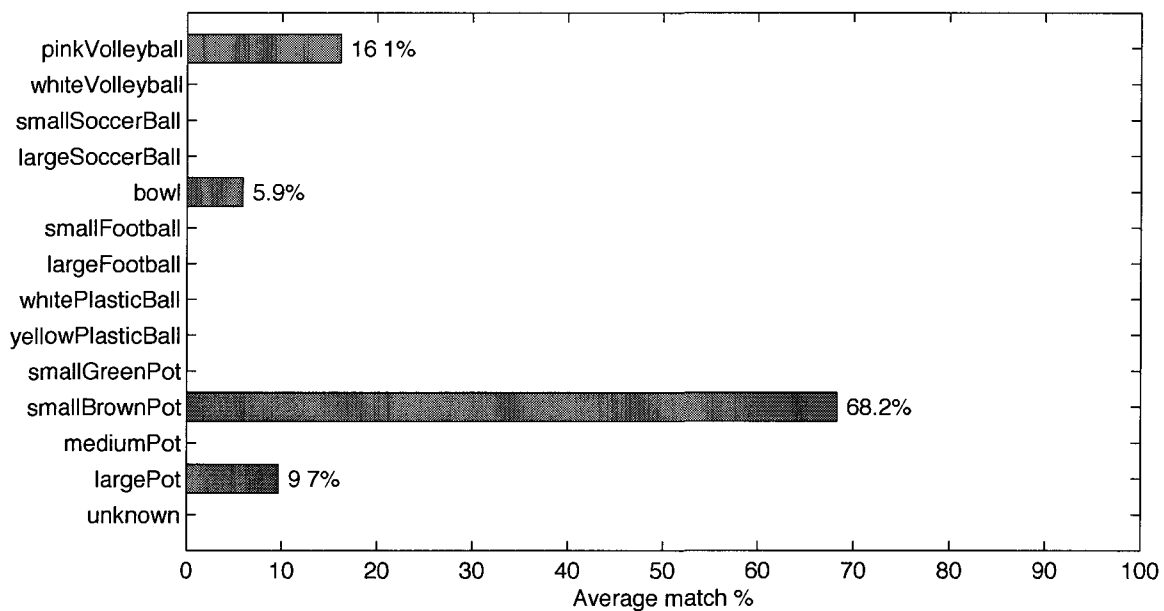
(c) Small soccer ball (3 detections).



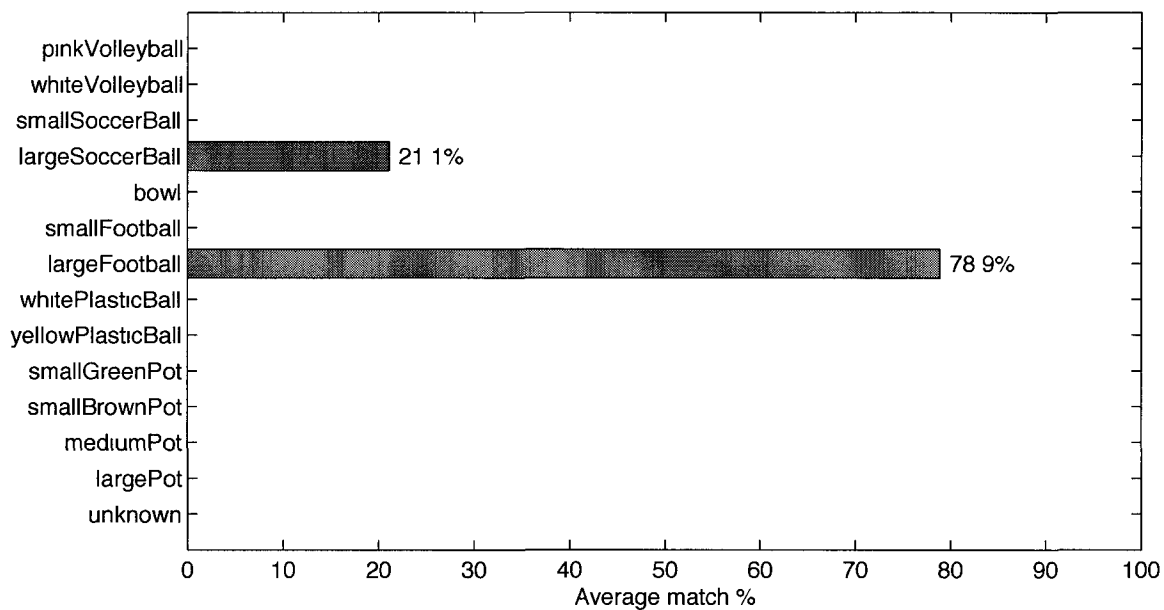
(d) Large soccer ball (11 detections).



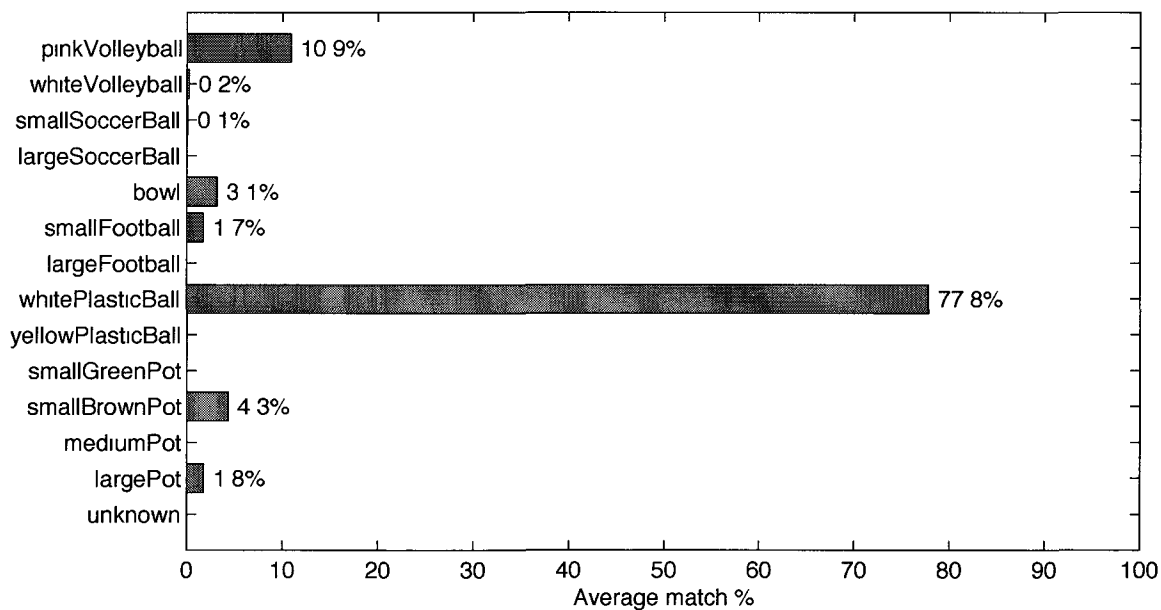
(e) Bowl (22 detections).



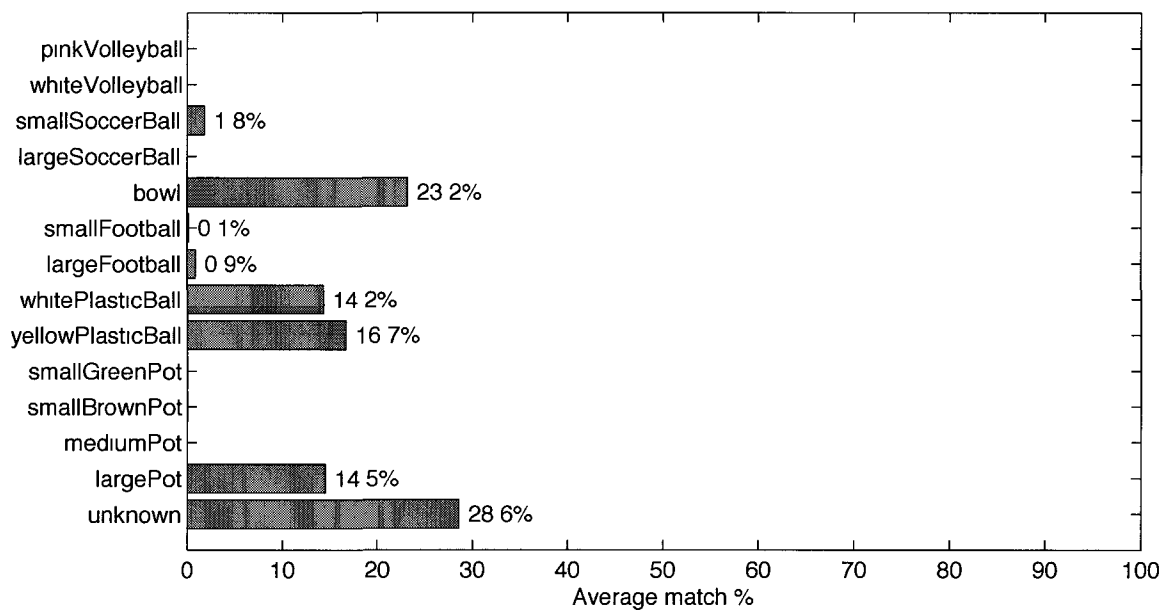
(f) Small football (4 detections).



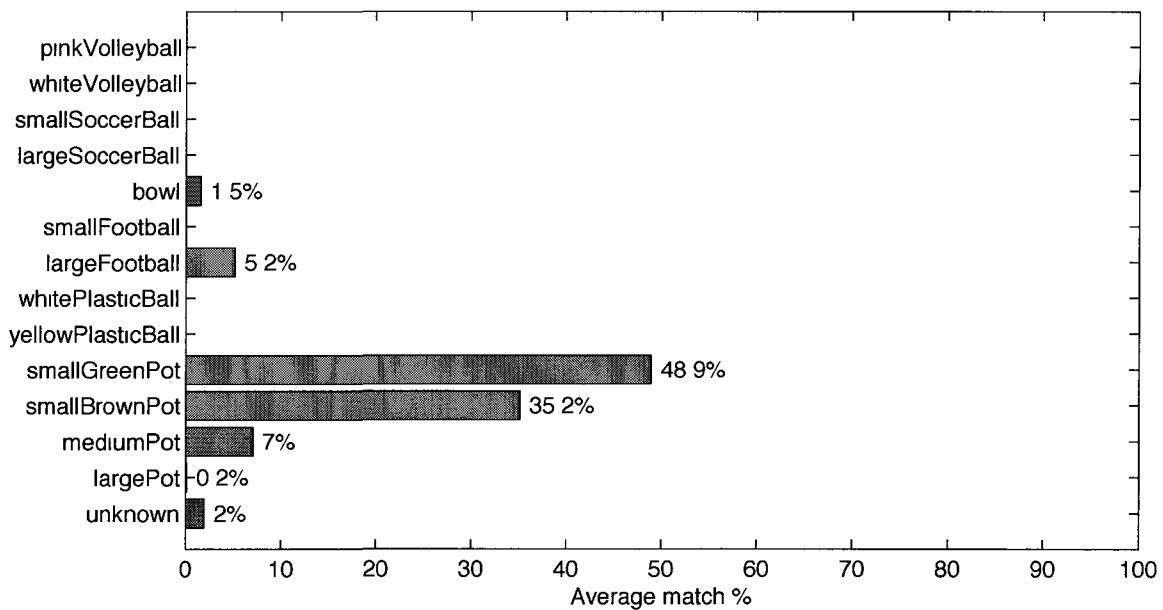
(g) Large football (5 detections).



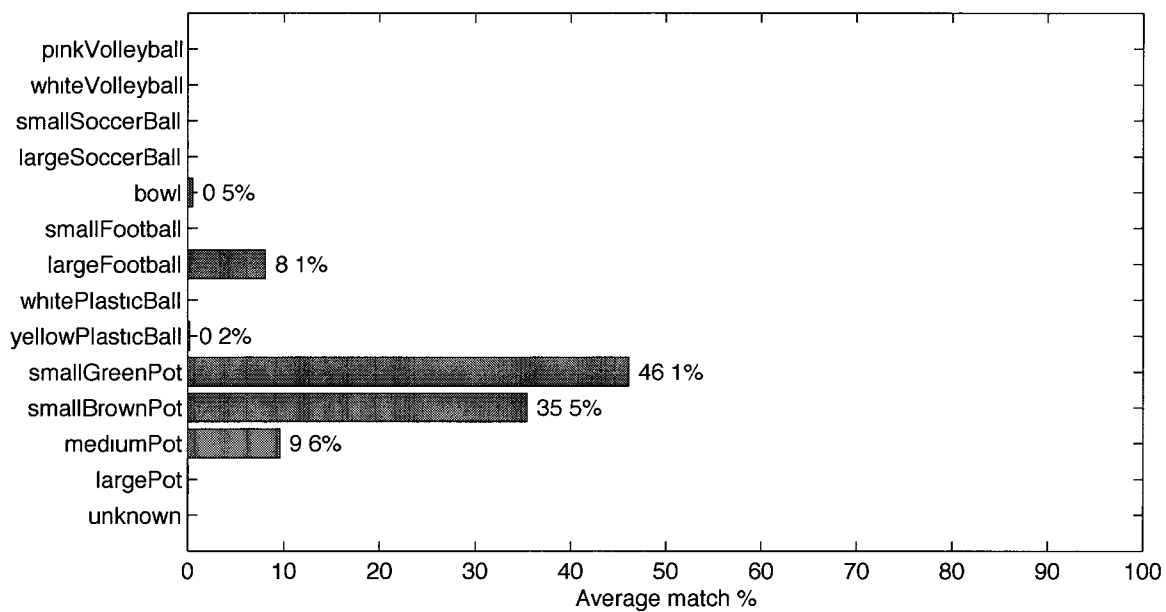
(h) White plastic ball (9 detections).



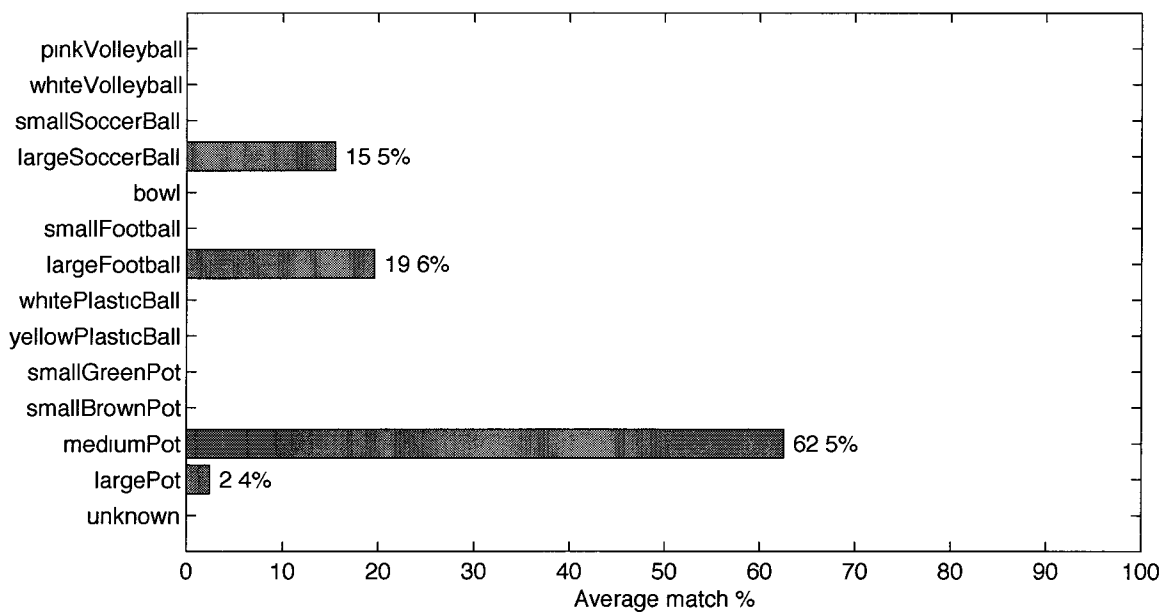
(i) Yellow plastic ball (7 detections).



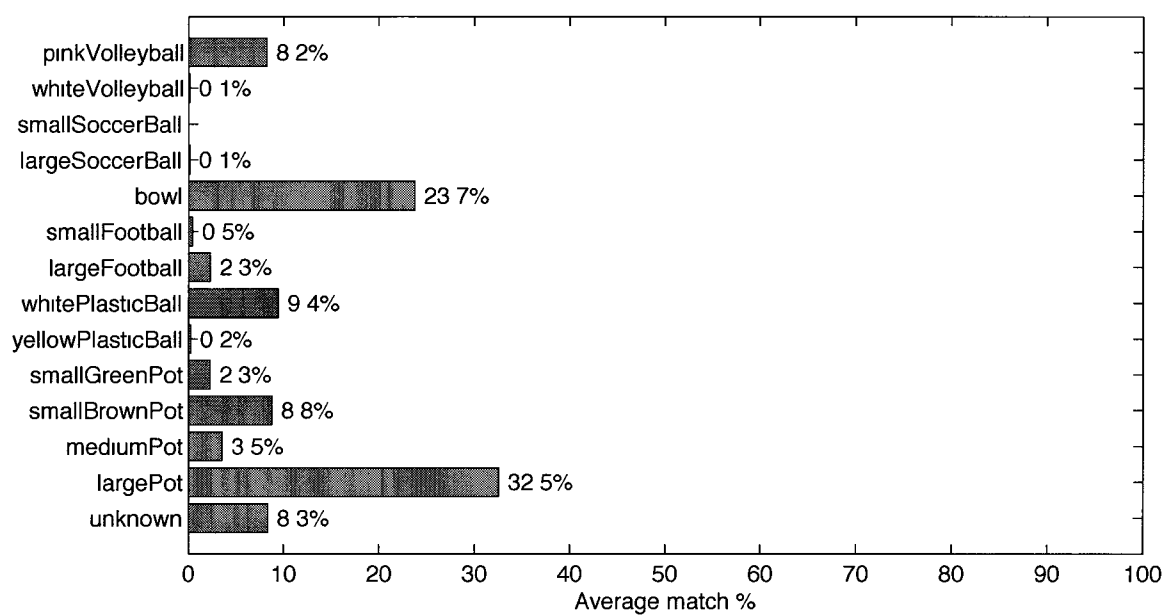
(j) Small green pot (51 detections).



(k) Small brown pot (51 detections).



(l) Medium pot (12 detections).



(m) Large pot (12 detections).

Figure F.1 – The average match percentage by the Bayesian network for each of the 13 objects used in the laboratory tests.