UNIVERSITY OF TECHNOLOGY, SYDNEY

Autonomous exploration and mapping of complex 3D environments by means of a 6DOF manipulator

by

Gavin David Paul

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and IT
Intelligent Mechatronic Systems Group

January 2010

Declaration of Authorship

I, Gavin David Paul , declare that this thesis titled, 'Autonomous exploration and mapping of complex 3D environments by means of a 6DOF manipulator' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF TECHNOLOGY, SYDNEY

Abstract

Faculty of Engineering and IT
Intelligent Mechatronic Systems Group

Doctor of Philosophy

by Gavin David Paul

The futuristic vision of industrial robotic systems that operate in complex, unstructured and diverse environments is beginning to become a reality due to the advances in computing, sensing and control. Automatically acquiring the structure and the properties of an environment in a timely manner is one of the key tasks that need to be accomplished in many field robotics applications. This thesis presents a novel and efficient approach to the exploration of three-dimensional (3D) environments using an industrial robot manipulator. The approach presented combines the objectives of 3D map building and surface material-type identification. The manipulator is manoeuvred through a sequence of viewpoints that are selected to maximise the quality of the map generated, minimise the time taken for the exploration, as well as minimise the uncertainty of the surface material type estimation, all whilst avoiding potential collisions between the manipulator and the environment.

The thesis first focuses on acquiring the geometry of surfaces in the environment while exploring the industrial robot manipulator's collision-free configuration space. Ellipsoidal virtual bounding fields are positioned around the manipulator's links so that distance queries can be performed and collisions with obstacles in the environment or unexplored space are avoided. Information theory is used to measure the information remaining on the geometric map and the manipulator's configuration space. A sampling strategy is used to select candidate viewpoints which are predicted to reduce the information remaining to measure. Each viewpoint enables the manipulator to position and orientate a sensor so that environment data can be gathered. The candidate viewpoint solutions can then be ranked based upon the exploration objectives. The collected sensor data is fused into a map. The map is then segmented into groups of Scale-Like Discs (SLDs), which are generated via principal component analysis.

Once the surface geometry becomes available, a strategy is required to maximise the accuracy of the surface material-type identification. Surface material-type identification is made possible through intensity measurements, which indicate the reflectivity of the surface when illuminated by an infra-red laser. Thus, identification is significantly influenced by the relative geometry between the sensor and the surface to be identified. Information theory is used again to determine surfaces which have not had their surface material-type identified. Appropriate viewpoints facilitating accurate identification are selected by solving an optimisation problem using the Levenberg-Marquardt algorithm.

This two-stage exploration approach is shown to successfully determine viewpoints enabling an accurate environmental map to be generated. The proposed algorithms and approaches are integrated into the system, Autonomous eXploration to Build A Map (AXBAM). Extensive experimental studies have been conducted on a complex steel bridge structure using a Denso industrial robot that has been equipped with a laser range finding sensor. These experimental studies demonstrate the efficacy of the AXBAM system.

Acknowledgements

I would like to thank my supervisor Associate Professor Dikai Liu for continually assisting me throughout the course of this project and providing hours of intelligent receptive interactions which have ultimately led to a more complete, quality thesis.

Thanks to the rest of the team at CAS, especially my co-supervisor, Professor Dissanayake for his helpful insights, patience and passion for robotics.

Thanks to my friends Nathan Kirchner and Tarek Taha for showing me how exciting research can be.

Thanks to my RTA project mentors Stephen Webb and Ngai (Raymond) Ming Kwok for research guidance and surreptitious pacification. Thanks to the Engineering Faculty at UTS who provided computers for me to work on and robotics equipment to experiment with.

Special thanks goes to my family in particular my mum - always my teacher. A very special thanks to my wife Mariko who constantly believed in me and encouraged me. My extended family's support undoubtedly led to the creation of this thesis.

Contents

Declaration of Authorship **i**

Abstract **ii**

Acknowledgements **v**

List of Figures **x**

List of Tables **xv**

Abbreviations **xvi**

Nomenclature **xvii**

Glossary of Terms **xxi**

1 Introduction **1**
- 1.1 Background 3
- 1.2 Motivation 4
- 1.3 Scope 6
- 1.4 Contributions 7
- 1.5 Publications 8
  - 1.5.1 Directly Related Publications 8
  - 1.5.2 Related Publications 9
  - 1.5.3 Open Source Project Authorship 10
- 1.6 Thesis Outline 10

2 Review of Related Work **12**
- 2.1 Object and Environment Mapping 12
  - 2.1.1 Sensor Selection for Exploration and Mapping 13
  - 2.1.2 Environment Representation 14
  - 2.1.3 2D Mapping 16
  - 2.1.4 3D Mapping 17
- 2.2 Exploration for Different Purposes 19

2.2.1 Environment Exploration 20
2.2.2 Inspection of Objects Surfaces 21
2.2.3 Exploration of a Manipulator's Configuration Space 24
2.3 Efficient Collision-free Motion Planning 26
2.3.1 Maintaining Robot Safety in Planning 26
2.3.2 Efficient Movement Planning 27
2.4 Conclusion 29

3 Exploration of Unknown 3D Complex Environments: A Two-stage Approach 30
3.1 Architecture of the Two-stage Exploration Approach 30
3.2 Stage One: Geometric Exploration 32
3.2.1 Objective Functions 33
3.2.1.1 Geometric Information Reduction 33
3.2.1.2 Configuration Space Information Reduction 33
3.2.1.3 Manipulator Movement Minimisation 34
3.2.1.4 Safe Manipulator Pose Selection 35
3.2.2 Multi-objective Optimisation 37
3.2.3 Viewpoint Solution Space Sampling 38
3.2.4 Stage One Termination 40
3.3 Stage Two: Material-type Exploration 41
3.3.1 Cost Functions 44
3.3.1.1 Viewpoint-to-Surface Orientation 46
3.3.1.2 Viewpoint-to-Surface Range 46
3.3.1.3 Viewpoint-to-Targeted Surface Centre Accuracy 47
3.3.1.4 Safe Manipulator Pose Selection 47
3.3.1.5 Field-of-View Sensing Constraint 48
3.3.1.6 Manipulator Joint Limitations 48
3.3.2 Non-linear Optimisation 49
3.4 Conclusions 50

4 Information Measurement-based Exploration 51
4.1 Measurement of Remaining Information 52
4.1.1 Two-Dimensional Exploration Demonstration 55
4.2 Geometric Information Model 57
4.2.1 Regions of Interest 60
4.3 Configuration Space Information Model 63
4.4 Material-Type Information Model 69
4.4.1 Surface Material-type Belief Update 70
4.4.2 Surface Material-type Identification Demonstration 72
4.5 Conclusions 75

5 Surface Representation and Map Segmentation 76
5.1 Surface Representation Problem 77
5.1.1 Data Collection Process Formalisation 78

- 5.2 Scale-Like Disc Generation . . . 80
  - 5.2.1 SLD Parameter Tuning and Simulation Results . . . 85
- 5.3 Map Segmentation . . . 90
  - 5.3.1 Segmentation Formulation . . . 91
    - 5.3.1.1 Angle Between Surface Normals . . . 91
    - 5.3.1.2 Distance Between Centres of SLD . . . 92
    - 5.3.1.3 SLD Centre-to-Plane Distance . . . 93
    - 5.3.1.4 Difference Between Manipulator Poses . . . 94
  - 5.3.2 Map Segmentation by Means of SLD Clustering . . . 94
  - 5.3.3 Map Segmentation Demonstration . . . 96
- 5.4 Discussion and Conclusions . . . 100

6 Case Study: Exploration and Map Building of Complex Steel Bridge Structural Environments 101

- 6.1 Experiment Setup . . . 102
  - 6.1.1 The Environment and The Robot . . . 102
  - 6.1.2 AXBAM System Implementation . . . 103
- 6.2 Evaluation Criteria . . . 105
  - 6.2.1 Stage One: Geometric Mapping . . . 105
  - 6.2.2 Stage Two: Material-type Identification . . . 106
- 6.3 Experiment One: Exploration in a Simple Environment . . . 108
  - 6.3.1 Stage One of Exploration: Geometric Mapping . . . 109
  - 6.3.2 Stage Two of Exploration: Material-type Identification . . . 112
- 6.4 Experiment Two: Exploration in Complex Environment . . . 116
  - 6.4.1 Stage One of Exploration: Geometric Mapping . . . 117
  - 6.4.2 Stage Two of Exploration: Material-type Identification . . . 120
- 6.5 Experiment Three: Exploration in a Second Complex Environment . . . 124
  - 6.5.1 Stage One of Exploration: Geometric Mapping . . . 125
  - 6.5.2 Stage Two of Exploration: Material-type Identification . . . 128
- 6.6 Map Utilisation Discussion . . . 129

7 Conclusions 132

- 7.1 Summary of Contributions . . . 133
  - 7.1.1 A Novel Two-stage Manipulator-based Exploration Approach . . . 133
  - 7.1.2 Pose Selection Including Sensing Constraints . . . 133
  - 7.1.3 Information Theory-based Measurement Models . . . 133
  - 7.1.4 Surface Representation and Map Segmentation . . . 134
  - 7.1.5 Practical Contribution . . . 134
- 7.2 Discussion of Limitations . . . 135
- 7.3 Future Work . . . 136

Appendices 137

A Grit-blasting Pose Selection and Planning **138**
A.1 Manipulator Pose Selection for Grit-blasting 139
A.2 GA-based Optimisation of Target Point Sequencing 141

B Laser Range Classifier for Material-type Identification **144**

C A Robotic System for Steel Bridge Maintenance **147**
C.1 Hardware 148
  C.1.1 Industrial Manipulator 148
  C.1.2 Laser Range Finding Sensor 149
  C.1.3 Computing Hardware 149

D SLD Generation: Additional Parameter Tuning Results **150**

Bibliography **152**

List of Figures

1.1 Cluttered and obstacle-rich bridge maintenance environment recreated with a variety of typical surface material-types and lighting conditions. 2

1.2 *a*) A bridge spanning water; *b*) Structure underneath the bridge which must be maintained (paint/ rust removal and recoating). 4

2.1 3D mapping with a mobile robot [1]. *a*) Mine mapping cart with four laser range finders; *b*) Volumetric surface map visualisation using scan matching techniques. 17

2.2 *a*) The RoboScan system [2] acquiring the "Deposed Christ" high-relief (real size: 2.5 by 1.5 metres); *b*) Corresponding high resolution digital model produced. 19

2.3 C-space exploration [3] reduces the uncertainty in C-space and enables the manipulator a greater freedom of movement, so as to traverse C-space and achieve the final *goal* configuration. Initially, all but the C-space of the start pose was unknown. 25

3.1 Outline of the two-stage exploration approach showing the categories which each of the objectives falls into, and the named objective functions. 31

3.2 The predicted coverage of information remaining, assuming all n_Q contending viewpoints are used. An exponential curve is fitted to the data so as to observe convergence characteristics. 40

3.3 Pose selection parameters in surface material-type identification (distance to SLD targeted, d, surface centre point, $\hat{P}_i$, actual sensor's centre ray surface intersection point, $\mathbf{p}_a$, and angle, ϕ, to surface normal $\vec{n}_i$). A 3D Field-Of-View (FOV) is attained by tilting the 2D laser range finder with the 5th joint of the manipulator. 44

3.4 Sigmoid functions for a cost function, $g_c(\theta)$. *a*) Cost function for a maximum constraint e.g. $\theta < \theta_{max}$. *b*) Cost function for minimum and maximum constraints e.g. $\theta_{min} < \theta < \theta_{max}$. 45

4.1 2D exploration demonstration. The sensor has a 360° FOV and no obstacles exist besides the boundaries. The selection of next viewpoints is arbitrary. Sensing begins from the top left corner in *(a)*, then a viewpoint position is selected out of the data collected *(b)*, and finally sensing continues into the unexplored areas in *(c)* and *(d)*. 56

4.2 *a)* First 15 viewpoint sensing iterations from the 2D exploration example; *b)* Graphs of the information remaining after each sensing iteration. The more sensing that is performed, the less information remains in the environment. 56
4.3 Ray casting occurs for each ray from the sensor. If a ray passes through a voxel it is determined to be free, voxels which contain a SLD are occupied and voxels which have not had a ray cast through them at any stage have unknown occupancy state. 60
4.4 Application specific requirements are included. In this case the approximate workspace regions of interest $\{A_1, A_2, A_3\}$ can assist the exploration approach to select viewpoints. An enlarged voxel (volumetric pixel) visualisation is also shown to the right. 60
4.5 Simulated occlusions. *a)* Occluded regions of high uncertainty occur where there are large ray length differentials; *b)* Tilting the laser sensor through α so as to get a 3D FOV. Fig. 4.6 graphs the data from this simulation. . . 62
4.6 Range measurements, $\mathbf{R}_{\theta,\alpha}$, from Fig. 4.5*b* as θ_i changes on the scan plane and the sensor is tilted through α about the axis of tilt rotation. Large differentials in the range measurements exist which correspond to edges in the geometry of Fig. 4.5. 63
4.7 Usage of C-space nodes. *a*) The likelihood, $L(c_i)$, that the jth node is used when moving between any two poses, calculated by a count over all nodes used. Some nodes in C-space are traversed with greater likelihood than others (up to five orders of magnitude on the logarithmic vertical likelihood axis in some cases). The horizontal axis is the index of nodes; *b)* Likelihood of usage of C-space nodes for the first 3DOF can be visualised in a 3D plot with the joints as axes. The likelihood of usage is indicated by the size and colour of the spheres. 66
4.8 *a*) Histogram of the poses interfered with by voxel index. *b*) Importance of the voxels in Euclidean space with colour and size based on the count of poses which they interfere with. The manipulator is also shown for scale. . 67
4.9 As the belief about the state of each area changes with each viewpoint, the surface material-type is identified and the information remaining is reduced. *a*) At the 2nd iteration, there is a great deal of information; *b*) By the 13th iteration the information has been slightly reduced; *c*) After the 26th iteration, in the close range inspection areas there is very little material-type information remaining which is possible to sense. 73
4.10 Remaining material-type information marked with $(\cdot)$ and predicted remaining information marked with (*) while exploring to determine the material-type. 74

5.1 Grit-blasting maintenance can utilise overlapping disc-shaped targets. . . . 77
5.2 A grid of 3D points in the robot base coordinate frame is produced from manipulator configuration and laser sensor range data. 79
5.3 Point cloud, P, consisting of $n_p = 80977$ vertices and obtained from laser scans gathered by a robot manipulator tilting the scanner. A surface mesh (not shown) is made up of sets of 3 connected vertices which form faces. . . 80
5.4 Flowchart of the SLD generation algorithm. 81

5.5 *a*) Home point of the centre voxel; *b*) Calculate distance to points in centre voxel and in the surrounding $3^3 - 1 = 26$ voxels; *c*) Perform PCA on points within μ of home point to determine a plane; *d*) Register points within $\pm\mu$ of the plane to the SLD. 82
5.6 Single SLD: centre and surface normal, $\{\hat{\mathsf{P}}, \vec{n}\}$, represents a sub-point cloud. 84
5.7 With respect to changing μ(mm). *a*) Time taken to generate surfaces; *b*) Number of small SLDs created n_{ss}; *c*) Percentage (%) of point registered. . 86
5.8 With respect to changing n_{min}. *a*) Time taken to generate surfaces; *b*) Number of small SLDs created n_{ss}; *c*) Percentage (%) of point registered. . 87
5.9 Results of SLD generation when increasing the number of points in the point cloud, n_p. *a*) Time taken (red ·) and the trend line (black line); *b*) Number (no.) of planes created (blue *). 88
5.10 *a*) Photo of example structure; *b*) Fused raw data point cloud; *c*) Bottom right view; *d*) Front view of planes created; *e*) Expanded view of the corner; *f*) Expanded view of the front - showing SLDs and the associated normals (red lines) 89
5.11 Data reduction: Number of raw data points collected (.); Mesh vertices, n_p, (+); Number of SLDs, n_{ss}, (*). 90
5.12 Five SLDs with normal vectors shown, represent a curved surface in the environment. The connectivity of the SLDs must be determined so they can be clustered into segments for ease of use by the manipulator path planner. 91
5.13 Angle difference, $\delta\theta_{ij}$, between normals on the ith and jth SLDs. 92
5.14 Distance between centres of the ith and jth SLDs is $\delta d_{ij} = \|\hat{\mathsf{P}}_i - \hat{\mathsf{P}}_j\|$. . . . 92
5.15 Showing the point of intersection between the normal of the ith SLD and the jth SLD's plane, $\hat{\mathsf{P}}_{i\to s_j}$, and the point of intersection between the jth SLD's normal and the ith SLD's plane, $\hat{\mathsf{P}}_{j\to s_i}$. 93
5.16 The point cloud, P, with $n_p = 10637$ vertices represented by $n_{ss} = 433$ small SLDs. These SLDs are overlayed on the original point cloud as discs. The manipulator is also shown for scale. 96
5.17 Binary connectivity graphs for the connectivity of SLDs: *a*) Angle between normals, ${}^{\delta\theta}\mathcal{B}$; *b*) Distance between centres, ${}^{\delta d}\mathcal{B}$; *c*) Average distance from an SLD centre to another SLD plane, ${}^{\delta p}\mathcal{B}$; *d*) Manipulator pose C-space distance, ${}^{\delta q}\mathcal{B}$. 97
5.18 Overall connectivity graph between SLDs, **B**. The graph is binary (black = connection, white = no connection), symmetric, and nodes clusters are visible. 98
5.19 *a*) The clustered connectivity for the first segment; *b*) The first segment in relation to the manipulator; *c*) Angular histogram for each joint of the poses for the first segment. This shows that manipulator poses are similar. *d*) The connectivity for the second segment; *e*) The second segment in relation to the manipulator; *f*) Angular histogram for each manipulator joint. Note that the angular histograms are different to (*c*). 99

6.1 Environment set-up with several materials highlighted. *a*) The "Unwins Bridge" channel; *b*) Various material-types in the bridge. 102

6.2 *a*) The replica bridge maintenance environment and; *b*) The prototype grit-blasting robot and platform. 102
6.3 Flow Chart of the AXBAM System. *left*) Stage One exploration: Geometric Mapping; *right*) Stage Two exploration: Material-type identification. 104
6.4 The environment to be explored in the first experiment. 108
6.5 *a-f*)Viewpoints 2 to 7 for exploration. 109
6.6 Stage One objective functions: *a*) to *d*) for (f_1, f_2, f_3, f_4). Each viewpoint has up to 1000 possible candidates, (red $\cdot$) with one selected (blue *). . . . 110
6.7 The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations. 111
6.8 *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow. 112
6.9 *a*) Photo of the robot in the second viewpoint of Stage Two; *b*) The second viewpoint indicating the targeted SLD which lies on the edge of the containment plastic and the silver tape which is misidentified as painted metal; *c*) The third viewpoint (i.e. after the second viewpoint identified the material-type of the surface). 113
6.10 Stage Two viewpoint 2 constraints and objective functions: *a*) angle to surface normal $g_1(\vec{Q})$; *b*) distance to surface $g_2(\vec{Q})$; *c*) accuracy $g_3(\vec{Q})$; *d*) manipulator safety $g_4(\vec{Q})$; *e*) 3D field of view $g_5(\vec{Q})$; *f*) manipulator's joint limits $g_6(\vec{Q})$. 114
6.11 *a*) The sum of the squared objective functions, $\mathbf{g}(\vec{Q})$, for viewpoint 2. *b*) The manipulator joints over the 20 LM optimisation iterations so as to determine the second material-type identification viewpoint in Stage Two. . 115
6.12 Material-type information remaining after each viewpoint, $\mathbf{H}_3(\mathbf{M})$. 116
6.13 The environment for the second experiment. 116
6.14 The first six viewpoints used for Stage One exploration. 117
6.15 Viewpoints with all contending possibilities, (red $\cdot$) and the viewpoint selected (blue *). *a*) to *d*) The Stage One objective functions, 1 to 4. 118
6.16 The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations. 119
6.17 *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow. 120
6.18 *a*) and *b*) Stage Two viewpoint 12; *c*) and *d*) Viewpoint 13; and *e*) Viewpoint 14 with material-type maps. 121
6.19 Determining the viewpoint 13 in Stage Two of exploration including the constraints and objective functions: *a*) angle to surface normal $g_1(\vec{Q})$; *b*) distance to surface $g_2(\vec{Q})$; *c*) accuracy $g_3(\vec{Q})$; *d*) manipulator safety $g_4(\vec{Q})$; *e*) 3D field of view $g_5(\vec{Q})$; *f*) manipulator's joint limits $g_6(\vec{Q})$. 122
6.20 *a*) The sum of the squared objective functions, $\mathbf{g}(\vec{Q})$ for each viewpoint in Stage Two; *b*) The manipulator joints over 27 LM optimisation iterations so as to determine material-type identification viewpoint 13 in Stage Two. . 123
6.21 Material-type information remaining after each Stage Two exploration viewpoint, $\mathbf{H}_3(\mathbf{M})$. 124

6.22 The environment in the third experiment. 124
6.23 Exploration viewpoints 1 to 6. In this experiment eleven viewpoints are required, although after six viewpoints, the information reduction is not significant. 125
6.24 Viewpoints with all contending possibilities, (red ·) and the viewpoint selected (blue *). *a*) to *d*) The Stage One objective functions 1 to 4. 126
6.25 The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations. 127
6.26 *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow 127
6.27 Material-type information remaining after each Stage Two exploration viewpoint, $\mathbf{H}_3(\mathbf{M})$. 128
6.28 *a*) The target environment; *b*) Material-type map after the 18th viewpoint. 129
6.29 *a*) Grit-blasting plan created from a SLD-based segmented map. *b*) Actual trajectory of the grit-blast spot over the roof of the bridge environment using a laser pointer and a long exposure camera. 130
6.30 A manipulator conducting grit-blasting of a metal I-beam surface based on a map autonomously generated by AXBAM. 130

A.1 Parameters which must be optimised to achieve successful grit-blasting maintenance results. 139
A.2 Optimisation by GA with normalised parameters. 141

B.1 *a*) and *b*) The effect of surface roughness (ψ) and reflectance (C_r) on the *reflected light's intensity* illustrating how different parameter combinations produce uniquely shaped return intensity curves. *a)* Via Hancock model [4]; and *b)* via experiment [5]. *c) and d)* The effect of ψ and C_r on the *range measurement error* with the *c*) Hancock model; and *d*) experimental data. . 145

D.1 With respect to changing n_{min} and μ *a*) The time taken to make surfaces; *b*) Number of small SLDs created n_{ss}; *c*) The percentage (%) of points registered. 151

List of Tables

4.1 Environment Information Measurement Variables 54

5.1 SLD Generation Test Case Parameters 89
5.2 Map Segmentation Demonstration Constraint Parameters 96

6.1 Experiment Parameters 105

C.1 Denavit and Hartenberg [6] parameters, joint types and limits for the Denso VM-6083 manipulator arm, *Note 1*): all joint types are revolute, *Note 2*) the rotation angle parameter θ of all joints is 0. 148
C.2 Hokuyo URG-04LX Scanning Laser Range Finder relevant specifications. The sensor is lightweight, low power, short range with high accuracy and high resolution. 149

Abbreviations

ARC	Australian Research Council
AXBAM	Autonomous eXploration to Build A Map, system
CAS	ARC Center of Excellence for Autonomous Systems
CAM	Computer Aided Manufacturing
C-space	Configuration Space of manipulator
DOF	Degrees of Freedom
EM	Expectation Maximisation algorithm
FOV	Field Of View
GA	Genetic algorithm
GUI	Graphical User Interface
HRI	Human-Robot Interaction
LRC	Laser Range Classifier [5]
MER	Maximum Entropy Reduction
MLE	Maximum Likelihood Estimate
MSE	Mean-Squared Error
NSW	New South Wales (Australia)
OG	Occupancy Grid
PCA	Principal Component Analysis
RTA	Roads and Traffic Authority of New South Wales
SLD	Scale-Like Disc
UTS	University of Technology, Sydney

Nomenclature

	General Formatting Style
$f(\cdots)$	A scalar valued function
$\mathbf{f}(\cdots)$	A vector valued function
$[\cdots]^T$	Transpose
$\|\cdot\|$	Absolute value
$\|\cdot\|$	Vector length and normalised vector
C	Covariance matrix
D	A diagonalised matrix
$H(Y)$	Entropy of a random variable Y
$\{n, m\}$	Independent variables signifying the last index of a set or to refer to a count
$P(x_i)$	Probability of discrete state x_i
$P(x\|z)$	Conditional probability of x, given evidence z
$[\cdot]./[\cdot]$	Piecewise division i.e. ith vector element divided by ith element of another vector
	Specific Symbol Usage
A_i	The ith geometric region of interest
c_{wi}	Weighting co-efficient of the ith region of interest
$dist(\vec{Q})$	The minimum algebraic distance to all unsafe points in an environment for all of a manipulator's encompassing ellipsoids, as a function of the manipulator pose, $\vec{Q}$
${}^{j}\mathbf{c}$	C-space node set interfered with by voxel j

$f_i(\vec{Q})$	ith Stage One objective function
$g_i(\vec{Q})$	ith Stage Two cost function, optimal value, $\hat{g}_i = 0$
$\mathbf{g}(\vec{Q})$	Sum of squared cost functions
$\mathbf{H}_1(\mathbf{X})$	Geometric information remaining (i.e. uncertainty) of state of geometric environment $\mathbf{X}$
$\mathbf{H}_2(\mathbf{C})$	C-space information remaining in all voxels (in Euclidean space) because of uncertainty in C-space $\mathbf{C}$
$\mathbf{H}_3(\mathbf{M})$	Material-type information remaining about state of voxels containing surfaces in an environment $\mathbf{M}$
j	Voxel's index
k	Sensing viewpoints iterator during exploration
$L(c_i)$	Likelihood of traversing a C-space node, c_i, (i.e. a manipulator pose) during a random trajectory
n_p	Number of points in a point cloud or vertices in a mesh map
n_Q	Feasible poses (i.e. nodes) sampled from C-space solution space $\mathbf{Q}$
${}^j n_Q$	Count of C-space nodes interfered with by voxel j
n_m	Number of surface voxels requiring material-type identification
n_s	Number of map segments the environment is divided up into
n_{ss}	Number of small SLDs
n_t	Number of surface material-type states
n_u	Number of unknown voxels in an environment
n_v	Number of voxels in an environment
$\mathbf{p}$	A vector position variable (point or vertex) $[\mathsf{x}, \mathsf{y}, \mathsf{z}]^T$
P	A vector (or set) of 3D points or vertices
$\mathbf{p}_a$	Point where end-effector tool (sensor or maintenance) is directed
$P({}^jX = {}^jx_o)$	Probability that the jth voxel's occupancy state variable ${}^jX$, is in a possible occupancy state
$P({}^jM = {}^jx_m)$	Probability of jth voxel state variable, ${}^jM$ being a possible material-type

$P({}^j x_c)$	Probability of jth geometric voxel interfering with a path through C-space while taking into consideration the relative likelihood of traversing each C-space node
q_i	Individual manipulator joint for 6DOF case $i \in \{1, \ldots 6\}$
$\vec{Q}$	Industrial manipulator's joint vector, $[q_1 \ldots q_6]$, also a sensing viewpoint
$\mathbf{Q}$	Viewpoint solution space for manipulator pose joint vectors
$R_{\theta,\alpha}$	Matrix of distance range values from viewpoint, when tilting scanner through angle, α, where elements are scaler range values $r_{i,j}$
s_i	ith SLD
${}^0T_f(\vec{Q})$	Homogenous end-effector robotic transformation matrix at pose $\vec{Q}$ in base coordinate frame
${}^{i-1}T_i(q_i)$	Homogenous transformation matrix from link $i-1$ to i based on the joint, q_i
${}^fT_s(\vec{Q})$	Homogenous transformation matrix between the end-effector and sensor. Combined with ${}^0T_f(\vec{Q})$ to describe viewpoint transforms
$V_r(\vec{Q})$	Approximate volume that the robot currently occupies at pose $\vec{Q}$
$V_{new}(\vec{Q})$	New volume of geometric space sensed from the latest viewpoint

${}^j x_o$	Occupancy states of jth voxel (i.e. freespace, unknown, occupied)
${}^j x_c$	C-space interference states for the jth geometric voxel
${}^j x_m$	Material-type states of the jth voxel (estimated as $\hat{x}_m$)
α	Tilting the planar laser sensor through α results in a 3D FOV
ϕ_{max}	Sensing angular constraint for LRC to identify material-type
ρ_{min}	Sensing accuracy constraint for LRC

Combinations of Variables

$\{d_{min}, d_{max}\}$	Sensing range constraints where d must be for the LRC
$\{\hat{P}_i, \vec{n}_i, \mu\}$	ith SLD centre 'home point', $\hat{P}_i$, normal vector $\vec{n}_i$ and radius μ

$\mathbf{p}_{c,i}$	Manipulator's collision avoidance ith ellipsoid centre vector, $\mathbf{p}_{c,i} = [\mathsf{x}_{c,i}, \mathsf{y}_{c,i}, \mathsf{z}_{c,i}]^T$
$\mathsf{a}_{e,i}, \mathsf{b}_{e,i}, \mathsf{c}_{e,i}$	The ellipsoid parameters for equatorial radii $[\mathsf{a}_{e,i}, \mathsf{b}_{e,i}]$ and polar radius, $\mathsf{c}_{e,i}$, that encase the ith manipulator joint for collision avoidance
$\{q_{i,max}, q_{i,min}\}$	Set of maximum and minimum physical angular limits on each joint. $q_{t,max}$ related to the tilting joint for 3D sensing
$\{v_i, \lambda_i\}$	ith eigenvector and corresponding eigenvalue of sub-point cloud
$\{\delta\theta_{ij}, \delta d_{ij}, \delta p_{ij}, \delta q_{ij}\}$	For surfaces $\mathbf{s}_i$ and $\mathbf{s}_j$: the angular difference, distance between the centres and the planes, pose joint difference

Glossary of Terms

Blasting	Grit blasting maintenance operations on certain surfaces.
Environment	A complex 3D unstructured place in which a manipulator is positioned. Assumed to have some structural characteristics such as planar surfaces.
Freespace	Areas in the environmental model or map that are known to be free of objects, obstacles and surfaces.
Grid	A type of representation based on OGs used to divide a space into discrete grid cells. For 3D geometry this becomes voxels, and for C-space this becomes nodes.
Iteration	A single step or viewpoint which is determined by optimisation, or in the case of Levenberg-Marquardt optimisation, one iteration of the least squared optimiser.
Manipulator	In this thesis, this is a six-degree of freedom Denso industrial robotic manipulator, with either a laser range scanner or a grit-blasting tool mounted on the end-effector.
Map	Model of the geometry and material-type of surfaces in the surrounding environment.
Node	Manipulator pose in 6D C-space.
Obstacle	An object within the manipulator's workspace which a manipulator can collide with.
Occlusion	Not visible from a viewpoint due to an obstruction.
Obstruction	A surface within sensing range which causes an occlusion.
Platform	The movable platform on which the robot manipulator is fixed.

Planning	The act of generating a path (and motion) course which the robot can then follow to get between two poses.
Scaffolding	Temporary structure built under and around the bridge to allow maintenance by humans or robots.
Sensor	Generally refers to a laser range finder which returns range values to objects in an environment.
Scale-Like Disc	Small disc-shaped targets arranged in a scale-like overlapping pattern to form a representation of surfaces.
Solution Space	All possible solutions to an optimisation problem. In this case, it is within the physical bounds of the industrial robot manipulator's movements
Structural	Mainly consisting of planar surfaces in a man-made fixture such as a bridge. This type of environment can be unstructured with regards to a robot if it is not set up specifically for the robot.
Surface	This is the face of an object in the environment. The geometric and material-type properties must be determined.
Surface Normal	A 3D vector perpendicular to a surface.
Material-type	The type of material on an object's surface. Includes painted steel, rusted steel, timber, plastic and concrete
Unstructured	A Real-world environment that cannot be set up to facilitate ease of actuator movements. There are no limitations on the geometry of the environment, although it is generally assumed to consist of relatively smooth or planar surfaces.
Viewpoint	A position in space and an orientation of a sensor that a corresponding manipulator pose can achieve, can be expressed by the homogeneous transformation matrix, ${}^{0}T_s(\vec{Q})$, or manipulator joint vector, $\vec{Q}$.
Voxel	Volumetric Pixel which represents a 3D cube-like volume in Euclidean space.

Chapter 1

Introduction

Typical applications of industrial robots in manufacturing require significant attention to 'structure' the environment so that the robot is able to perform the necessary tasks. When a robot is placed in more complex and unstructured environments, as in many field robotic applications, autonomously acquiring knowledge about the environment is essential. Recent advances in sensing and mapping have made it feasible to address this problem, thereby enhancing the potential for robotic systems to be used in tasks such as civil engineering infrastructure maintenance. This thesis focuses on algorithms and approaches for solving the problem of acquiring and representing the geometry and material-type properties of surfaces in unknown or partially known three-dimensional (3D) environments using an infra-red laser scanning sensor mounted on a six-degree-of-freedom (6DOF) industrial robot manipulator. An approach is developed that allows a manipulator equipped with the sensor to explore and map unknown environments (i.e. acquire the geometry and material-type of surfaces in the environment), while discovering and avoiding obstacles. Furthermore, the exploration approach aims to efficiently gather information about the surrounding environment. This thesis presents techniques for: (a) positioning and orientating the sensor so that information it gathers contributes significantly to the geometric map and the robot manipulator's freedom of movement (i.e. configuration space); (b) representing a general 3D environment using a number of small Scale-Like Discs (SLDs) and clustering the SLDs into task-specific map segments; (c) considering safe and efficient robot manipulator movements when positioning the sensor in a partially known environment, in order to avoid potential collisions with known objects or objects that may be present in currently unknown regions; and (d) positioning and orientating the sensor to identify surface material-type properties in an environment by taking the sensor's constraints and other limiting factors into consideration. The exploration and mapping approach is developed and implemented on a Denso VM-6083 industrial robot equipped with a Hokuyo URG-04LX laser range finder on its end-effector. The outcomes of the thesis, including the exploration approach, algorithms, methods and a deployable system, are evaluated using a full-scale replica of a steel bridge structure (Fig. 1.1) - originally built to demonstrate the feasibility of an autonomous robotic system for steel bridge maintenance.

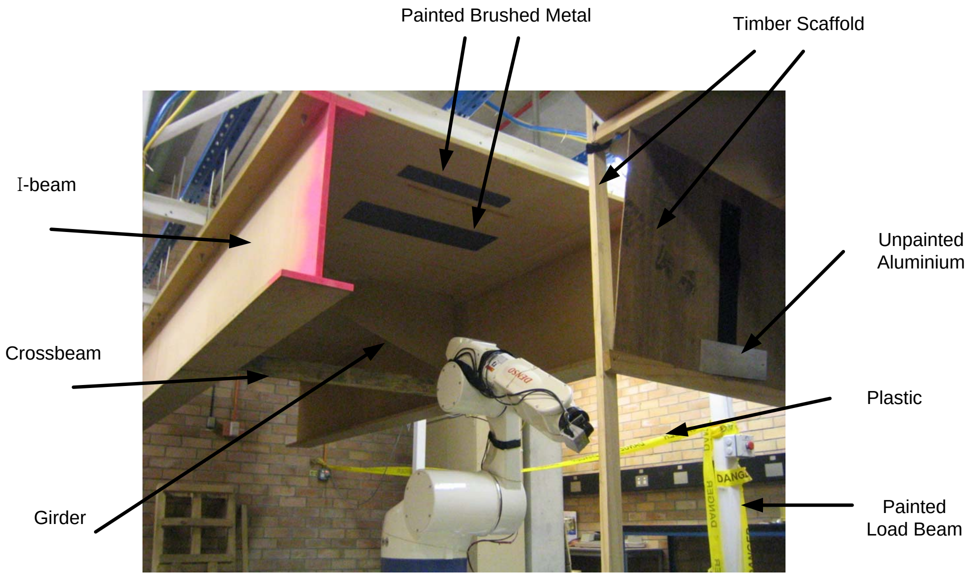


Figure 1.1: Cluttered and obstacle-rich bridge maintenance environment recreated with a variety of typical surface material-types and lighting conditions.

This chapter introduces the research work presented in the thesis. It commences with a background of the target application scenario of steel bridge maintenance and details key research issues. The remaining sections of this chapter describe the thesis scope and its main contributions and provide the outline of the thesis.

1.1 Background

Bridges are an essential element of transport infrastructure worldwide. There are over 30,000 road and rail bridges across Australia. Bridge maintenance and replacement form one of the largest expenditure items associated with traffic infrastructure management. For example, the State Government of Queensland has allocated $350 million towards replacing approximately 100 old and obsolete road bridges in regional Queensland for the five year period between 2006 and 2010 [7].

Corrosion is the primary cause of failure in steel bridges [8], however corrosion can be minimised by painting the steel structure with a protective coating. Until the 1980s, paints used on most steel bridges in Australia contained red lead, and in some cases asbestos or Chromium-6. Periodic inspection and maintenance of these bridges is an expensive undertaking, with significant associated environmental and employee health and safety issues.

The maintenance of steel bridges consists of two main procedures: rust/paint removal and repainting. An effective and efficient method of large scale paint stripping is grit-blasting, but this method has inherent problems. Grit-blasting is extremely labour intensive and hazardous [9], and is the most expensive operation performed during steel bridge maintenance. Workers must spend long periods of time handling forces of around 100N [10]. Also, dust particles from the paint and the grit used to remove the paint, have been shown to cause asbestosis, lead poisoning and silicosis [11] - a debilitating form of occupational lung disease caused by inhalation of crystalline silica dust. Some robotic systems exist for grit-blasting railway cars, ships [12], aircraft [13], tele-operated cleaning [14][15], and maintenance [16]. However, these systems are often pre-programmed or have the operator in direct control of the system with visual feedback on the progress. Hence, these systems do not operate autonomously. An autonomous robotic system has the potential to greatly increase productivity and reduce the risk exposure of workers.

This thesis originated as part of an ongoing project conducted at the University of Technology, Sydney (UTS) to develop an autonomous robotic system for stripping paint and rust from steel bridges (Fig. 1.2). The project has strong support from the Roads and

Traffic Authority of New South Wales (RTA). The ultimate objectives of the project are to reduce human exposure to hazardous and dangerous debris (containing rust, paint particles, lead, asbestos and/or Chromium-6), to relieve workers from labour intensive tasks, to reduce costs associated with bridge maintenance and to improve the safety of bridge maintenance operations. A 6DOF Denso VM-6083 industrial robot manipulator is used in the current prototype system. The desired outcome of this project is to develop a robust robotic system that performs bridge maintenance autonomously.

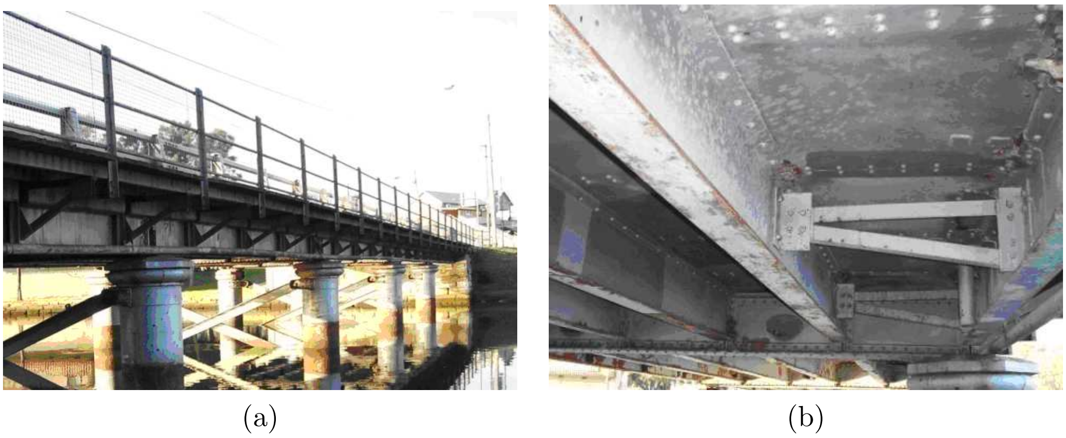

(a) (b)

FIGURE 1.2: *a*) A bridge spanning water; *b*) Structure underneath the bridge which must be maintained (paint/ rust removal and recoating).

1.2 Motivation

To enable a manipulator-based robotic system to conduct maintenance operations autonomously, control tasks such as positioning, orientating and moving the manipulator in relation to surfaces, needs to be done efficiently. Planning these control tasks for a manipulator affixed with a grit-blasting nozzle directing high-velocity garnet particles requires serious consideration. In order to correctly plan these tasks, the robotic system must have access to, or be able to generate, a high quality and immediately usable map of the surrounding environment. Due to the complexity of the environment, it would be particularly time consuming and tedious for even a skilled operator to create a complete environment map capturing the detailed surface geometry and the surface material-type (e.g. concrete, rusted/ painted metal, timber). Therefore, the autonomous robotic system must be able to efficiently and accurately position and orientate a sensor by means of a manipulator so as to map the geometry and material-type of surfaces in the environment. Henceforth, one position and orientation of the sensor will be referred to as a “viewpoint” in this thesis.

Viewpoints dictate the completeness and the quality of the environment map and the time taken for the mapping process. An exploration approach is used to determine worthwhile viewpoints. With the objective of building an accurate and complete map of a complex 3D environment, the exploration approach must discover both freespace (i.e. the area in the environment that is free of objects) and surfaces. Surfaces of objects that are inside the workspace of the manipulator will be referred to as “obstacles” since their presence can obstruct manipulator movements. Moving the manipulator in areas that are unknown (because they have not yet been explored) could lead to collisions with as yet unknown obstacles. Therefore, by discovering areas of freespace inside the manipulator’s workspace, the manipulator can be allowed a greater freedom of movement. Additionally, by discovering surfaces in the environment, it will be possible to determine surface completeness, predict the information which can be gained from a viewpoint with greater accuracy, and finally ascertain the geometric surfaces which require their material-type to be identified.

Complete exploration of a complex 3D environment requires a number of viewpoints. Between each viewpoint the manipulator must be moved along a collision-free trajectory through the discovered freespace. In order to select the next viewpoint from a set of contending viewpoints, the complexity and length of movements required to position the manipulator at a viewpoint must be considered by a motion planner. Efficient and safe motion planning of the 6DOF manipulator in a complex 3D environment containing limited freespace is challenging. The motion planning process must factor in a margin of error in relation to known obstacles and unknown areas in the manipulator’s workspace.

Exploring a complex 3D environment in order to quickly generate an accurate, complete and exploitable map, immediately prior to maintenance operations is clearly a necessary and yet challenging task. Efficient algorithms and approaches need to be developed and verified in real practical applications and environments.

1.3 Scope

This research aims to develop an integrated two-stage exploration approach and algorithms for 3D map building and surface material-type identification of complex 3D environments. The approach consists of Stage One for geometric space and manipulator configuration space (C-space) exploration, and then Stage Two for surface material-type identification. Information theory-based algorithms are developed for Stage One to formulate both the geometric map information and the manipulator's C-space information. The outcome of each step of Stage One of exploration is a Pareto optimal viewpoint. Techniques are generated with the aim to fuse sensor data collected at each viewpoint and transform it into a more compact and usable representation. Stage Two of the exploration approach defines an optimisation problem to incorporate the surface material-type identification sensing constraints. An appropriate viewpoint is selected by solving the optimisation problem using Levenberg-Marquardt's algorithm. This two-stage approach enables an autonomous robotic system to develop a complete map of a complex 3D unknown environment.

There are a number of assumptions made to limit the scope of this thesis. It is assumed that the laser scanning sensor is mounted at the end-effector of an industrial robot manipulator, and thus the location of the sensor at any time is accurately known. The sensor is assumed to provide range measurements between itself and the surfaces in the environment, as well as intensity information that indicate the reflectivity and roughness of these surfaces. The exploration approach developed is based on a 6DOF robotic manipulator with high repeatability and accuracy. The environment is assumed to be static and to consist of predominantly planar surfaces. While it is assumed that the sensor can acquire partial information about the geometry of the environment from its initial location, any further robot motions need to avoid collisions between the robot manipulator body and the obstacles in the environment. Additionally, the manipulator needs to avoid moving into the regions that are not yet explored. The sensor is assumed to scan the environment while the robot manipulator arm is stationary. The time for scanning is a constant and the time required for motion between successive sensor locations (i.e. viewpoints) is related to the robot joint displacement and velocity (assumed constant).

The objective of geometric mapping is to autonomously acquire a complete 3D representation of surfaces from a fixed base position of the manipulator. According to maintenance safety practices [17] used in industry, the environment which could be damaged by grit-blasting is required to be completely known directly prior to the commencement of maintenance. An acceptable surface quality (i.e. accuracy of the 3D map) according to blasting manuals [18] and grit-blast planning research [19] will facilitate the motion planning of a robot holding a grit-blasting nozzle which directs a mid sized ($\approx 20mm$ diameter) maintenance grit-blast spot. Thus, surfaces to be blasted can be represented in an effective way as multiple small Scale-Like Disc (SLD) targets. Each SLD requires the material-type to be identified with high-confidence (several surface observations will attain greater than 99% certainty according to [20]).

The intended application-specific outcome is an implemented system referred to as AXBAM (Autonomous eXploration to Build A Map). The AXBAM system encompasses the hardware and the developed exploration approach presented in this thesis. The AXBAM system satisfactorily gathers surface geometry and material-type information incrementally and selects collision-free manipulator robot movements.

1.4 Contributions

The contributions of this thesis are:

- An information theory-based model measuring the remaining information in an environment, in terms of the geometry, the C-space of the manipulator and the material-type of surfaces [21][22][23].
- A technique for map segmentation which reduces the geometric data to a number of clustered, easily utilised and partially overlapping SLDs by employing probabilistic surface mesh fusion, Principal Component Analysis (PCA) and cluster analysis [24][25].
- A technique to determine the appropriate viewpoint (i.e. position and orientation) for a range sensor to collect surface property information such that the surface

material-type can be identified. The technique formulates the sensing constraints (distance and angle to surface, and targeting accuracy) and manipulator limitations into objective functions to be optimised with Levenberg-Marquardt's algorithm [23][5].

- A two-stage exploration approach to facilitate the pursuit of information on geometry, C-space and surface material-type in 3D complex environments by means of a 6DOF industrial robot manipulator [23].
- An integrated system (AXBAM) which implements the exploration approach and algorithms on the prototype robotic system to generate 3D maps for use in steel bridge maintenance operations [26][9][27].

1.5 Publications

1.5.1 Directly Related Publications

[1] **G. Paul**, D.K. Liu, N. Kirchner, S. Webb, "Safe and Efficient Autonomous Exploration Technique for 3D Mapping of a Complex Bridge Maintenance Environment", In *Proc. 24th International Symposium on Automation and Robotics in Construction*, pages 99-104, Kochi, India, 2007.

[2] **G. Paul**, D.K. Liu, G. Dissanayake, "Autonomous Robot Manipulator Exploration and Mapping System for Bridge Maintenance", *Robotics and Autonomous Systems*, Elsevier, *Under review - submitted April 2008.*

[3] **G. Paul**, D.K. Liu, N. Kirchner, "An Algorithm for Surface Growing from Laser Scan Generated Point Clouds", In T. Tarn, S. Chen, and C. Zhou, editors, *Robotic Welding, Intelligence and Automation*, pages 481-492, Springer-Verlag, Berlin, 2007.

[4] **G. Paul**, N. Kirchner, D. K. Liu, G. Dissanayake, "An Effective Approach to Simultaneous Mapping and Surface-type Identification of Complex 3D Environments", *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping*, 26(11-12 SI): pages 915-933, 2009.

[5] W.K. To, **G. Paul**, N.M. Kwok, D.K. Liu, “An Integrated Approach to planning for autonomous grit-blasting robot in complex bridge environments”, In *Proc. 4th Virtual International Conference, EU-funded FP6 I*PROMS Network of Excellence on Innovative Production Machines and Systems*, pages 313-318, Wales, UK, 2008.

[6] W.K. To, **G. Paul**, N.M. Kwok, D.K. Liu, “An efficient trajectory planning approach for autonomous robots in complex bridge environments”, *International Journal of Computer Aided Engineering and Technology*, Inderscience, 1(2):185-208, 2009.

[7] N. Kirchner, T. Taha, D.K. Liu and **G. Paul**, “Simultaneous Material Type Classification and Mapping Data Acquisition Using a Laser Range Finder”, In *Proc. International Conference on Intelligence Technologies: Intelligent Technology in Robotics and Automation*, pages 124-129, Sydney, 2007.

1.5.2 Related Publications

[8] N. Kirchner, **G. Paul** and D.K. Liu, “Bridge maintenance robotic arm: Mechanical technique to reduce the nozzle force of a sandblasting rig”, In *Proc. 1st International Symposium on Digital Manufacturing*, pages 12-18, China, 2006.

[9] **G. Paul**, D.K. Liu, “Replanning of Multiple Autonomous Vehicles in Material Handling”, In *Proc. IEEE International Conference on Robotics, Automation and Mechatronics, RAM*, pages 231-236, Bangkok, Thailand, 2006.

[10] N. Kirchner, D.K. Liu, T. Taha and **G. Paul**, “Capacitive object ranging and material type classifying sensor”, In *Proc. International Conference on Intelligence Technologies: Intelligent Technology in Robotics and Automation*, pages130-135, Sydney, 2007.

[11] D.K. Liu, X. Wu, **G. Paul**, G. Dissanayake, “Case Studies on an Approach to Multiple Autonomous Vehicle Motion Coordination”, *Journal of Wuhan University of Technology*, 28(164):26-31, 2006.

[12] M. Clifton, **G. Paul**, N.M. Kwok, D.K. Liu, Da-Long Wang, “Evaluating Performance of Multiple RRTs”, In *Proc. IEEE Conference on Mechatronic and Embedded Systems and Application*, pages 564-569, Beijing, 2008.

[13] D.K. Liu, G. Dissanayake, P. B. Manamperi, G. Fang, N. Kirchner, **G. Paul**, S. Webb, P. Chotiprayanakul, J. Xie, "A Robotic System for Steel Bridge Maintenance: Research Challenges and System Design", In *Proc. Australasian Conference on Robotics and Automation*, Canberra, 2008.

1.5.3 Open Source Project Authorship

1. Sourceforge Project: CAS robotics:
   http://sourceforge.net/projects/cas-robotics/
2. Mathworks Matlab FileExchange: Multiple submissions
   http://www.mathworks.com/matlabcentral/fileexchange/authors/31652

1.6 Thesis Outline

The thesis is structured so the first two chapters outline the research and provide background for the thesis. Chapter 3 gives the outline of the proposed two-stage approach which leads to a realisation about why the models described in Chapter 4 are required. Chapter 4 presents the details of modelling in each stage. Chapter 5 presents the map building framework subsequent to exploration data collection. Implementation results and conclusions are presented in Chapter 6 and Chapter 7 respectively. The detailed outline of each chapter is as follows:

Chapter 2 presents related work on robotic systems which perform autonomously in an unknown environment. Three challenging research aspects emerging from the related work are reviewed: the generation of complete and accurate maps for different purposes; the exploration of partial maps; and the planning of optimal and collision-free movements.

Chapter 3 presents a two-stage exploration approach to determining subsequent manipulator viewpoints. Stage One is composed of objective functions incorporating three categories: information measures, efficiency and movement. Stage One's objective functions are combined into a multi-objective optimisation problem which can be solved. Stage Two commences once the geometry of an environment is mapped. Based on the map, another set of six objective functions are devised to address sensing and manipulator constraints such that the material-type of surfaces can be identified. Stage Two objective functions are then minimised with a non-linear optimisation approach.

Chapter 4 presents an information theory-based measurement of the remaining information in an environment, for use in Stage One of the exploration approach. It first describes the importance of information about the environment's geometry in specific regions, and then describes how uncertainty can be used to measure the remaining information which must be minimised. It is demonstrated that there is a relationship between the geometric information of an environment and the remaining manipulator's C-space information. The material-type remaining information is measured so that it can be used in Stage Two. Also, it is shown to be possible to predict the information remaining (i.e. geometry, C-space and material-type) once sensing has occurred at a specific viewpoint.

Chapter 5 describes how laser range data is collected from the sensor, and then how this data is initially reduced into surface meshes. An algorithm employing statistical theory is then presented to transform the high density surface mesh geometric data into a representation (SLDs) specifically devised to assist in manipulator trajectory planning. This chapter then presents a novel technique to segment the map by considering the proximity of SLDs and a manipulator's trajectory over the SLDs.

Chapter 6 implements the exploration approach and developed algorithms into the AXBAM system, and then presents real-world experimental results. This chapter also details case studies where the AXBAM system is applied.

Chapter 7 summarises the research work presented in this thesis. It then discusses limitations of the algorithms, the exploration approach and the implemented system. Conclusions are then drawn with regards to this research and avenues for future work are proposed.

Appendices provide the relevant work on maintenance related blast planning using the maps generated; the laser scanning sensor-based surface material-type identification; and the set-up and hardware used for the autonomous maintenance robotic system.

Chapter 2

Review of Related Work

Developing a robotic system that must perform autonomously in an unknown and complex 3D environment is challenging. Three main research challenges are: generating maps based upon sensor data; making exploration viewpoint decisions based upon partial maps; and planning optimal robot motion from viewpoint to viewpoint. Although these research challenges are often closely tied, this chapter separates them into three sections: mapping, exploration, and motion planning, and then presents the related work.

2.1 Object and Environment Mapping

Robust robotic systems with a degree of autonomy must be able to map the surrounding environment. Where interaction with surfaces is required, detailed 3D maps must be generated that contain geometric information and material-type information about the surfaces. Advances in sensing technologies make it possible to measure a variety of object and environment properties which can be stored in a map. There are various techniques available to represent a map so as to encapsulate properties or information about the environment, depending upon how the map will be utilised in the application.

2.1.1 Sensor Selection for Exploration and Mapping

Various sensing technologies have been utilised in exploration and mapping research including: stereo or multi-positional mono vision, laser or infra-red range finders, ultrasonic, capacitive and magnetic sensors. For the application domain targeted in this thesis, a sensor is required to gather high resolution geometric information and sense surface material-types. Microwave non-destructive evaluation [28], capacitive [20] and magnetic active sensors [29][30] are attractive since they can differentiate material-types and can even detect the presence of rust in some situations. However, generally such sensors fall victim to limited sensing range. The sonar sensor provides a low-cost solution to two-dimensional (2D) mapping for mobile robots [31] although is not appropriate for detailed 3D mapping since it is prone to several measurement errors due to various phenomenon (e.g. multiple reflections, wide radiation cone, low angular resolution) [32]. Research into 3D geometric mapping has shown stereo vision provides a possible sensor solution [33][34][35][36]. It has also been shown possible to determine surface properties using cameras, as in [37], where image data is used to segment different types of wood with a neural network based classifier. However, in isolation stereo vision techniques are susceptible to lighting conditions and therefore camera-based mapping is often performed in combination with laser range sensors [35][38][39]. High resolution maps of the surface geometry are more easily gathered with laser range sensors than with stereo vision. Infra-red laser range finding sensors are popular in mapping and exploration literature since they provide high angular resolution and range accuracy to surrounding objects. These sensors are inherently equipped with an internal light source and hence are active type sensors. The Hokuyo URG-04LX laser range finder presented in [40] has been shown to enable surface material-type identification through the Laser Range Classifier (LRC) system [5]. In the LRC system, material-type identifications were based upon analysis of the range data and laser intensity data, which the sensor can be configured to return [41]. The small, lightweight URG-04LX laser range finding sensor along with the corresponding surface material-type identification technique will be used in this thesis to gather range and material-type data.

2.1.2 Environment Representation

A 2D laser range finder scanning sensor [40] returns sets of ranges to objects in an environment. These are in the form of sets of ray lengths on a scan plane with corresponding angles from the centre ray. By combining the ray lengths and the angles it is possible to determine the 2D location of an object in the laser's coordinate frame. In the case where a laser sensor is mounted to the end-effector of an industrial manipulator, since the pose of the manipulator (and the sensor's transformation from the end effector) is known, 3D locations can be determined with respect to the manipulator's coordinate frame. Adjacent points in a scan can be joined, and corresponding points from adjacent scans can also be joined to produce a 3D mesh composed of vertices (i.e. 3D points) and faces. Together vertices and faces in a mesh can represent surfaces in an environment [42][43]. However, a mesh map representation may not be ideal for use in exploration and maintenance task planning. Algorithms exist [44][45] for generating tool paths using point clouds. It has been shown in [25][19] that it is beneficial to represent a map as a group of small, partially overlapping targets to improve the efficiency of manipulator trajectory planning in maintenance operations.

In regards to capturing geometric properties of an environment, many representations can be used according to Thrun [46], including occupancy grids, meshes, surfaces and objects. For mapping when the sensor pose is known, a popular and effective algorithm is the Occupancy Grid (OG) method, developed by Elfes [47] and Moravec [48], which is "a stochastic tessellated representation of spatial information that maintains probabilistic estimates of the occupancy state of each cell in a lattice defined". OG-based mapping addresses the problem of generating a consistent map from noisy or incomplete sensor data and has widespread use for data fusion [49]. Since OGs are probabilistic they can also be extended to provide an information measurement of the geometry. However, an OG representation alone is not sufficient for surface interaction planning. In order to realise sensing occlusions which can occur in exploration, a more appropriate surface representation is required.

Developing high levels of abstraction (e.g. object recognition) is particularly useful in Human-Robot Interaction (HRI) [50]. Zender et al. [51] presented "conceptual spatial" representations for indoor mobile robots. This research drew upon human spatial cognition models to break up the representation into layers of abstraction (i.e. points, surface and objects). Recent works by Rusu et al. [52] and Vasudevan et al. [53] focussed on mapping and recognising objects from a representative database. By storing the association between objects and the place where the object is commonly found (e.g. room in a house), the system could "reason" as to the location of the robot. The mapping approach adopted by Rusu et al. [52] was based on a point cloud representation similar to [54] which fitted polynomials to a point cloud based only on coordinate data and estimated surface normals. This type of map representation is common in HRI applications, and with further work and modifications it could be used in other applications such as maintenance operations.

Other popular representation approaches such as marching cubes [55][56], tetrahedrons [57] and the volumetric method [42], can produce complex detailed surfaces by using millions of triangles. Qhull [58] is a popular dimension software package for computing convex hulls and Delaunay triangulations when mapping an object or part. Related work exists for surface reconstruction or updates of 3D triangular mesh models from unorganised range scanner points in manufacturing parts [59] or representing machining changes made to a part [60]. Research by Laidlaw et al. [61] fused geometric Magnetic Resonance Imaging (MRI) data and brain material-type properties using a combination of OGs and meshes. Although the application domain is different, combining OGs and meshes provided a representation framework within which to combine geometric, freespace and material-type information. Mesh-based representations such as those described are an ideal graphical display for an operator. However, for the planning surface interactions using a manipulator, small 3D targets or groups of targets (i.e. segments) [62][50] are more beneficial [19][63], particularly if tightly packed or overlapping [24] like scales. Additionally, exploration algorithms which make predictions based upon a partial map, can be aided by an appropriate map representation [46]. Surface generation methods exist for facial recognition [64][65] and 2D position estimation [66] using Principal Component Analysis (PCA). PCA mathematical methods can be used to reduce data and generate surface normals. Techniques for surface segmentation [67] [68] and refinable surface enclosures [69] are also popular in the literature. Extensions to the surface creation and representation techniques are required so they can be used during exploration and trajectory planning for a manipulator that is required to interact with an object's surface.

In this thesis, fundamental volumetric mesh surface fusion techniques based on the work by Curless [42] which was further extended by Webb [43], will be utilised to store surface data. OGs will be used to both divide an environment into discrete portions, and to maintain a probabilistic belief about the state of each portion. This probabilistic representation makes it possible to measure surface geometry and material-type identification information. OGs will be used to investigate whether geometric uncertainty in a manipulator's workspace relates to uncertainty for a manipulator's movement. Surface representation and segmentation techniques must be devised to enable the prediction of information remaining in an environment subsequent to sensing. The map generated as a result of exploration is required to reduce the data, and to appropriately represent the data in order to aid manipulator trajectory planning such as in maintenance operations.

2.1.3 2D Mapping

2D mapping is commonly used in mobile robotics. Performing mapping where the map is also used for localisation (because the position of a mobile robot is not known accurately) is known as Simultaneous Localisation And Mapping (SLAM) [70]. SLAM approaches can be used to generate 2D occupancy grid maps such as the work in [71] where Chow et al. developed a novel approach to direct the search for information at each viewpoint. The approach used a neural network to update parameters and Bayes update to generate 2D occupancy grid maps for a robot with holonomic constraints. Alternate techniques of 2D mapping exist using OG representations with the purpose of avoiding collisions [72]. 2.5D mapping for humanoid robots walking over flat surfaces was presented in [33][73]. Yguel [74] and Thrun [31] utilised the 2D OG concept with multiple calibrated laser range finders. Many exploration algorithms [75] gather 3D map data, but rely on 2D maps for making mapping decisions.

2.1.4 3D Mapping

According to Thrun [46], "a detailed 3D visual map [of an environment] may easily require millions of numbers". Statistically speaking, each number is a dimension of the underlying estimation problem, hence the 3D mapping problem can be extremely high-dimensional. Research on 3D mapping has shown that a 3D map can be generated of a complex terrain environment using a mobile robot equipped with sensors. However, when a geometric map is required to be complete, accurate, or to be interacted with, then 3D mapping strategies differ. Moreover, the additional degrees of freedom provided by a manipulator have been shown to be beneficial when positioning and orientating sensors.

A typical mobile robot system which performs 3D geometric mapping and SLAM is presented in [1]. This demonstrates that it is possible to acquire volumetric (3D) maps of underground mining environments Fig. 2.1. This system includes a novel sensor head, assembled from multiple long-range laser range finders, and an efficient scan-matching algorithm to construct 3D volumetric maps of environments hundreds of metres in diameter using a mobile robot.

(a) (b)

Figure 2.1: 3D mapping with a mobile robot [1]. *a*) Mine mapping cart with four laser range finders; *b*) Volumetric surface map visualisation using scan matching techniques.

Related work has also been done on solving the SLAM problem in 3D [76][77]. Moravec [78] developed algorithms for 3D mapping of indoor environments with the requirement of high precision localisation within the maps. Weingarten [62] presented a probabilistic orthogonal least-square sense plane fitting method which was implemented on a Pygmalion robot with two horizontal SICK scanners and a single vertical scanner for 3D data collection. 3D mapping and exploration was performed using probabilistic information and 2D map representation with results showing high resolution 3D maps were generated for an indoor environment with consistent surface material-type. Thrun [39] presented an algorithm for acquiring 3D maps (compact sets of textured polygons) of indoor environments, using a mobile robot equipped with range and imaging sensors in order to extend the popular expectation maximisation algorithm. Using laser data, 3D SLAM has been conducted outdoors [79] [80] to gather operator recognisable representations of unstructured environments. A great deal of the literature focuses on a robot moving on a 2D plane to build a 3D map of an environment [81] [82] [83], or focuses on localisation rather than generating fine resolution 3D maps [84] [34]. However, if interactive tasks must be performed on the surfaces, then more dexterity is required so as to position and orientate sensors to overcome occlusions and to increase the geometric and material-type accuracy of surfaces.

A great deal of research has been performed using an industrial robot manipulator equipped with sensors [85] to create 3D virtual enclosed objects for computer aided manufacturing and painting. Other methods exist to sense an environment with stereo vision cameras on a manipulator [35], however calibration and limited geometric data resolution still present research challenges. Autonomous mapping techniques using robot manipulators to produce high quality maps suitable for surface interactions typically have pre-programmed robot movements. One such system [2] known as "RoboScan" (Fig. 2.2*a*) was developed for fast unattended acquisition of accurate and complete 3D models (Fig. 2.2*b*).

Working in unstructured environments with a manipulator is a challenging research problem [86][87][88]. The currently available manipulator-based 3D mapping algorithms are not sufficient when the full range of manipulator motions are required and when working in a partially known, unstructured and complex 3D environment that contains obstacles in the manipulator's workspace. An exploration approach is needed to determine how best to map the environment from safe viewpoints in order to gather a detailed 3D geometric map.

(a) (b)

FIGURE 2.2: *a*) The RoboScan system [2] acquiring the "Deposed Christ" high-relief (real size: 2.5 by 1.5 metres); *b*) Corresponding high resolution digital model produced.

2.2 Exploration for Different Purposes

When *a priori* knowledge is limited, and partial maps must be used to make decisions about where to map next, exploration algorithms are required in addition to mapping algorithms. Exploration has been a constant research challenge in robotics, as robotic systems for exploration must cope with partial and incomplete maps. A successful exploration strategy must accommodate contingencies that might arise during map acquisition. According to Thrun [46], exploration is a challenging planning problem which is generally solved sub-optimally via heuristics, with quantities that must be traded-off, including the expected gain in map information, and the time and energy it takes to gain this information. Three categories of exploration which aim to determine the next viewpoint and are of particular relevance to this thesis have been selected: 'environment' exploration where the aim is to determine the basic geometry of the environment using a mobile robot; 'inspection' type exploration to examine surfaces to gain high resolution details using a manipulator/ sensor combination; and 'Configuration space' (C-space) exploration which is a special type of exploration which can expand a manipulator's freedom of movement.

2.2.1 Environment Exploration

In the case of an indoor environment which must be explored, if it were possible to produce a graph whose vertices correspond to rooms and corridors, and whose edges correspond to doors, then the exploration problem would reduce to the well-known travelling salesperson problem [81]. However, by definition exploration is performed without complete knowledge of the map, therefore viewpoints must be located in the freespace map. Thus the more freespace available, the more viewpoint possibilities exist. It is noted by Isler [81] that generally, "today's approaches are usually greedy, that is, they choose control by greedily maximising information gain". To do this, the representation of an environment used in exploration requires evidence grids [89] or a probabilistic framework [90].

Many exploration algorithms exist for sensor-equipped mobile robots to avoid obstacles while exploring a 2D environment (e.g. unstructured indoor environment) to gather geometric information of the environment [91][92]. Another recent 2D exploration approach for mobile robots in dynamic environments was presented in [93]. It uses the extended Voronoi transform and fast marching method to provide "potential field maps" for robot navigation. Using this method, exploration was directed towards the most unexplored and free zones of the environment which have safe associated trajectories. Stamos [38] utilised vision-based exploration techniques to produce depth maps.

Machine and reinforcement learning and other artificial intelligence methods have been used in exploration, so as to develop state-to-state models through intermediate actions with defined reward functions. Ip et al. [94] developed such a strategy, with freespace update via Bayesian theory and an external reactive navigation scheme, for machine learning to perform autonomous exploration with a mobile robot. The system developed as a result of Ip's research successfully explored online an initially unknown a flat indoor area and produced 2D maps. Martinez-Cantin et al. [95] presented a policy learning focused algorithm for reinforcement learning while performing exploration, while Kollar and Roy [96] presented an exploration technique using reinforcement learning to automatically determine trajectories that lead to complete 2D maps being built. Given the state space for reinforcement learning techniques, it is a non-trivial task to explore and map in true 3D with a manipulator.

Another way to address the OG mapping problem is to use the Expectation Maximisation (EM) algorithm [97], hence formulating the mapping problem as a maximum likelihood problem in high-dimensional space. EM is an iterative algorithm that gradually maximises the expected log-likelihood [98] [99]. According to Thrun [46], EM was developed in the context of maximum likelihood estimation with latent variables. EM algorithms are said to constitute today's best solutions to the "correspondence problem" [100] in mapping where there is viewpoint location uncertainty. The incremental maximum likelihood method was used in [101] for small-scale map construction, where each map is used as a building block for a larger map. OGs and information gain have also been shown to be possible for 2D exploration [102]. Another exploration methodology [84] utilised hidden Markov models to map and classify a 3D outdoor terrain.

Research on mobile robotics has led to the development of algorithms to explore and gather data about an environment's geometry when moving on a planar surface or even in rough terrain. However, localisation research challenges, along with an inability to move a sensor to a required 3D position and orientate it with high dexterity, means accurate surface inspection requires more than a typical mobile robot. By using a repeatable manipulator and the exploration enabling algorithms studied in this thesis to define and predict the information remaining in an environment, detailed environment surface exploration is expected to become viable.

2.2.2 Inspection of Objects Surfaces

Inspection of objects is necessary when the aim of the exploration is to generate high quality complete surface representations. This can be conducted with the aid of an accurate industrial manipulator. However, in cases where a manipulator is used, strict constraints are often placed on manipulator movements and/or assumptions are made about a selected region of the manipulator's workspace being free of obstacles.

In research by Sujan [103], a small 3DOF robot manipulator was mounted on top of a mobile robot system to enable object surface inspection. Enabling algorithms based on iterative sensor planning and sensor redundancy were included to determine the next viewpoint for a camera so as to build 3D point cloud geometric models of an object in front of the mobile robot. Renton et al. [104] presented a sensor-based 6DOF manipulator system, called Plan-N-Scan, for collision-free, autonomous exploration and workspace mapping. This work developed a voxel (volumetric pixel) prediction algorithm to calculate the freespace within the manipulator's workspace which could be found from a viewpoint, and a manipulator movement planner between viewpoints. The algorithm developed allowed for the existence and basic shape of obstacles within the sensing range to be determined in a coarse OG. In order to improve the exploration predictions and the surface inspection capabilities of these algorithms, a surface map representation must be considered. Further research is necessary to enable close surface inspection so that sensing viewpoints can be compared based upon important exploration factors, such as the different types of information to be gathered and the difficulty of positioning and orientating a sensor.

Next Best View (NBV) is a type of incremental exploration approach which is of particular relevance for automated surface acquisition [105][106]. Given a set of range images, NBV research focuses on where to position and orientate a range sensor so as to scan the visible surfaces of an unknown object or scene. Ideally the enabling algorithms allow the object model or scene map to be built from a minimum number of range images. NBV research [107] for autonomous 3D object reconstruction has attempted to minimise the number of range images needed to inspect a convex hull object. The differentials of consecutive range sensor readings were used to predict occlusions and alternative viewpoints. However, generally the NBV algorithms investigated were used in environments which had been carefully structured, or where the manipulator's movement was manually or mechanically restricted so that collisions were not possible.

Algorithms [108] also exist for improving the view of complex surfaces. Reed et al. [109] investigated if a sensor could be positioned based upon a "visibility volume" of that target surface so as to generate a triangular mesh from range images. Surfaces on the solid hull were marked as "imaged" or "occluded" and the environment map was built incrementally by positioning a sensor to gain access of an unseen surface. The results showed a strong dependence upon the density of objects in the environment. Stuerzlinger's [110] work on "Imaging All Visible Surfaces" presented a heuristic approach to determine a theoretically near-optimal set of viewpoints for a pre-determined model. A hierarchy of viewing regions and visible elements was generated and viewpoints were taken from the viewing regions.

Son [111] showed how data about a free-form complete surface could be gathered for reverse engineering purposes by calculating the density of laser scan rays which were reflected from a surface. In Pito's work [112], each scan partitions the volume into "seen" and "void". This allows for the spatial representation of the surface that encloses the entire viewing volume and directions of the scanning rays. Therefore, seen and void surfaces are projected (using the well established technique of ray casting [113]) to a bounding surface and the next scanning position is determined by maximising the projected seen and void areas. More recently Larsson et al. [114] developed an industrial robot view/ path planning approach based upon the occluded regions from a scan. This approach also focused on sensing an object smaller than the manipulator, which was positioned in a known region of space. This constrained inspection problem means the manipulator is able to move safely within specific boundaries.

As shown, various related work exists on inspection and finding the NBV to construct a virtual model of an object. The NBV approaches generally use a small portion of the manipulator workspace, therefore the movements can be limited to be within known safe manipulator configurations. The focus is thus on "outside-looking-in" viewpoint planning [112], i.e. where the model to be acquired is bounded by a closed volume and scans are taken from a limited set of poses. Intuitively, there is a difference when attempting to inspect and map a surrounding environment where the manipulator and sensor are "inside-looking-out" [115]. In this case, algorithms must enable viewpoint planning which looks at the surrounding environment and utilises the whole manipulator's workspace to position a sensor. Although inspection systems from related work have novel algorithms to determine the information to be gained about a surface or environment from a viewpoint, no work has been found to address the problem of simultaneous surface inspection and freespace exploration, whilst considering the safe and efficient movement of a manipulator in a partially known environment.

2.2.3 Exploration of a Manipulator's Configuration Space

One challenging issue when exploring a complex 3D environment using a manipulator is to consider the manipulator's workspace and the necessity to explore the surrounding Euclidean space to enable greater freedom of movement. This challenge is known as exploring the manipulator's Configuration space (C-space) [116]. In the case of a 6DOF manipulator, a pose corresponds to a six-dimensional joint vector, henceforth referred to as a "C-space node". A single obstacle in Euclidean (3D) space (which by definition lies inside the manipulator's workspace) can potentially create multiple obstacle C-space nodes. Similarly, unknown, unexplored Euclidean space potentially creates unknown C-space nodes. An obstacle or unknown C-space node is one which cannot be used in a movement path between two manipulator poses. Therefore, if the immediately surrounding workspace of a manipulator is unknown, then movement without first exploring would result in entering unexplored space, which presents the danger of possible robot-environment collisions.

Early research on the determination of viewpoints for C-space exploration [117] focused on planning. Rating functions were defined in C-space with respect to the robot's motions, then sensing actions which enabled configurations with high ratings were selected. Mehrandezh [118] presented a generalised C-space exploration theoretical framework for a manipulator equipped with many proximity sensors. Theoretically, the freespace can be sensed, memorised, and then reutilised for further planning. However, in implementation Mehrandezh noted significant challenges when interpreting sensor readings.

Research into the exploration of a manipulator's C-space is also referred to as the "sensor-based motion planning" problem [119]. The purpose of this type of exploration is to facilitate manipulator path planning, that is, areas that affect or obstruct the movement of the manipulator are explored. Yu and Gupta [3][116] presented a sensor-based motion planner where the problem is defined as "determining the viewpoint which will lead to an expansion in C-space representation". An information theoretic approach was developed to estimate the C-space entropy, then direct sensing towards spatial regions which are both unknown and in the manipulator's workspace. As shown in Fig. 2.3, where the manipulator's workspace is initially unknown, the algorithm allowed for the expansion of movement possibilities (i.e. the freedom of movement) which in turn enables further

C-space exploration. Approaches to determine exploration viewpoints for C-space uncertainty reduction, for the sake of efficiency, generally do not consider sensing occlusions. This limits the accuracy when predicting the C-space uncertainty reduction in cases where obstacles are present in the manipulator's workspace.

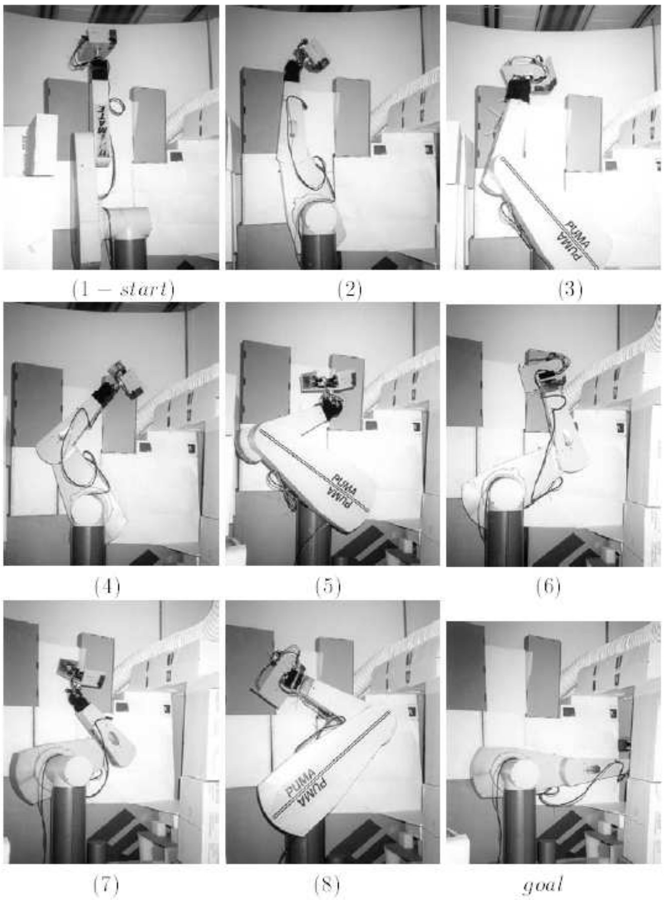

(1 − *start*) (2) (3) (4) (5) (6) (7) (8) *goal*

FIGURE 2.3: C-space exploration [3] reduces the uncertainty in C-space and enables the manipulator a greater freedom of movement, so as to traverse C-space and achieve the final *goal* configuration. Initially, all but the C-space of the start pose was unknown.

C-space exploration algorithms focus on the exploration of freespace in a manipulator's workspace which would affect the manipulator's movement if it were unknown. Exploration of C-space assists manipulator movement planning but is not concerned with generating a complete geometric map [120]. Exploring a complex environment and closely inspecting surfaces in that environment by means of a manipulator necessitates that the manipulator's C-space exploration be simultaneously considered.

2.3 Efficient Collision-free Motion Planning

Once an exploration algorithm has determined the information gain of potential viewpoints, the safe motion of the manipulator from viewpoint to viewpoint must be considered. Motion planning is required to determine safe movement paths through known space. Path and motion planning problems have been researched for several decades and during this period many approaches have been developed [121]. Unfortunately, there is no single approach which can be applied effectively across most environment circumstances. Two branches of motion planning are briefly reviewed here: maintaining robot safety in planning; and movement efficiency. Collision avoidance methods are necessary to maintain the safety of a robot. Collision avoidance focused planners which do not search for efficient paths are often referred to as reactive planners. Reactive planners can make step-ahead decisions based upon the immediate surrounds even without *a priori* knowledge. Efficient movement planning involves sampling and search-based strategies which are capable of assuming a more holistic planning view to determine existing efficient paths, but require more computational expense.

2.3.1 Maintaining Robot Safety in Planning

Reactive planners are generally used to maintain the immediate safety of a robot. They can react to actuator feedback [122][123] and sensor data returned from objects in an environment [124]. Alternatively, artificial potential field algorithms which generate force-fields to encase a robot or manipulator can be used to give control actions so as to act appropriately. Virtual bounding field algorithms for collision detection and avoidance reduce computational time compared to calculating the distance from the manipulator to obstacles using accurate topology models. Many techniques exist to cover the links of a robot manipulator, including polyhedrons [125], spheres [126][127] and ellipsoids [128]. Based on the measured distance between the manipulator and obstacles in the environment, collision avoidance algorithms are then used to steer the manipulator away from the obstacles.

Reactive planners combined with force-field collision avoidance methods are popular when planning a manipulator's motions. Juang [125] applied polyhedrons to represent a manipulator and potential field method to dynamically control it. Greenspan and Burtnyk [129] modelled a robot manipulator and obstacles with sets of spheres, measured the distances with a weighted voxel map and demonstrated successful collision avoidance and online planning. Lin and Chuang [130] used potential fields in a 3D workspace to generate a collision free path by locally adjusting the robot configuration for minimum potential. Yao and Gupta's approach [131] is able to avoid manipulator self collisions along pre-existing paths. Stilman et al. [132] presented a versatile algorithm for manipulation planning in a domain with movable obstacles named "spatial constraints". Garber et al. [133] also presented a constraint based motion planning approach for multiple collaborative robots in complex and dynamic 3D environments. Fernandez et al. [120] presented a collision avoidance framework for sensor-based (skin sensors) manipulator reactive planning. Chotiprayanakul et al. [128] presented a 3D force-field method where ellipsoidal shapes covered selected links of a manipulator enabling motion planning and collision avoidance of a 6DOF manipulator in complex and dynamic environments, whilst keeping the planned end-effector's path and speed unchanged.

Ellipsoidal bounding fields for a manipulator must be generated so that the computational intensive distance queries and collision checking can be reduced. Algorithms developed for reactive motion planning and collision avoidance can be very efficient and can significantly simplify collision checking. Therefore, it is advantageous to use bounding virtual fields for collision checking and to define the safety of the manipulator at, and in a trajectory to, a candidate viewpoint. However, reactive planners can fall victim to local minima problems [134], where the manipulator is drawn towards a goal by a virtual force but then resisted by a virtual force preventing a collision with an obstacle.

2.3.2 Efficient Movement Planning

Planning efficient manipulator movements can be done in C-space, where an n-degree freedom manipulator can be represented as a node in n-dimensional C-space. This problem is then solved with a C-space node-to-node path planner, while considering joint limitations and control steps. This type of planner searches the C-space from a start point to a goal. In order to select a viewpoint (i.e. a manipulator pose) for exploration and mapping, the efficiency of manipulator movement between viewpoints must be considered. This section reviews the most widely used movement planning approaches.

Heuristic approaches exist to search through a 6DOF manipulator C-space, often in a breadth first manner as in [135], avoiding what is often known as “C-space obstacles” [136]. Other approaches simplify the problem by searching consecutive degrees of freedom with a V-graph methodology [137], or by making some degrees of freedom inactive [135].

Recently, algorithms such as LaValle’s Rapidly-exploring Random Trees (RRTs) [138] and Probabilistic Road Maps (PRM) [139] have gained popularity. The RRT search algorithm is a type of randomised path planner, especially suited to search for routes through complex environments. As such, the RRT is a means of rapidly searching high-dimensional spaces that have both algebraic and differential constraints [140]. It is primarily designed to act as a fast, single-query planner and has the advantage that pre-processing of the environment is not required. The effectiveness of the RRT is demonstrated in [141] and has since been applied to many complex path and motion planning problems, such as in [142][143][144]. Extensions to the RRT algorithm include dynamic-domain RRTs [145], nearest-neighbour searching [146] and the anytime RRT algorithm [147].

The Probabilistic Road Map (PRM) method, advanced particularly by Hsu et al. [148][149], is a popular probabilistically complete motion planning algorithm in robotics. It first takes random node samples from a robot’s C-space, often with more sample density in narrow passages [150][151][152], then tests whether the C-space node samples are in freespace. A local planner is then used to attempt to connect these C-space nodes to nearby nodes. PRMs are popular in the literature when planning in already known environments, however some research on PRMs has been done even when *a priori* information is limited. In [3][153] and then later in [154], it was shown that as the manipulator C-space is explored, more safe C-space nodes can be added to the PRM. Once the C-space has been sampled based upon PRM or other methods, traditional graph search methods can be applied such as Dijkstra [155] or its extension, A* search algorithm [156]. The A* search has become one of the most popular graph search algorithms that finds the least-cost path between two graph nodes.

In the case of exploration by means of a manipulator, where the exploration process is required to be online, the planning of safe and efficient movements must also occur online. Additionally, once an environment is explored and maintained (i.e. grit-blasting is performed), the map or planning roadmap is unnecessary. Therefore, virtual ellipsoidal bounding field collision detection and avoidance techniques can be incorporated to ensure safety, while efficient movement planning from viewpoint to viewpoint can use the C-space sampling and search methods. Thus, it is possible to measure the difficulty and safety of moving the manipulator during exploration and mapping tasks.

2.4 Conclusion

An industrial robotic system required to operate in complex, unstructured and diverse environments is faced with many challenges. In particular, acquiring the geometry and properties of surfaces in the environment in a timely manner is required by many field robotics applications. Literature on mobile robotics presented 2D and 3D mapping algorithms, with both types generally relying upon 2D maps for exploration. When environment-surface interactions are required, there is a strong need to closely inspect the surrounding surfaces and explore partially known complex environments, so as to produce a quality map. Related work showed that object surface inspection is possible when environments are carefully structured and a manipulator is utilised to position and orientate sensors. Exploration of a manipulator's C-space was found to allow the manipulator a greater freedom of movement. To explore and build a map for use in surface interaction tasks such as maintenance operations, the information remaining in an environment (e.g. C-space, freespace and the surface geometry/ material-type) must be measured and predicted. Safe and efficient manipulator movement to position and orientate a sensor during exploration must also be considered. Therefore, various research issues must be addressed when devising an effective approach to exploration of complex 3D environments using an industrial robotic manipulator.

Chapter 3

Exploration of Unknown 3D Complex Environments: A Two-stage Approach

3.1 Architecture of the Two-stage Exploration Approach

When an environment is not structured for an industrial robot manipulator and no information is initially available about the geometry of surfaces in the environment, then exploration is required. Once the environment's geometry is mapped, including details of obstacles which will obstruct the manipulator's movement, then it is possible to perform detailed inspections of the surfaces to gain information such as the surface material-type. This chapter introduces an exploration approach comprised of two stages with different objectives. Each of the two stages focuses on determining viewpoints, for the specific purpose of discovering the geometry and the material-type of surfaces, respectively. Stage One introduces four objective functions. The first two objective functions model the geometric space and the manipulator's Configuration space (C-space) information remaining, and the other two incorporate the considerations of safe, efficient manipulator movements. Once the information remaining can no longer be reduced in Stage One and hence the surface geometry has been mapped, Stage Two commences. For each viewpoint in Stage Two, a small Scale-Like Disc (SLD) (selected because of the material-type information remaining) is considered in order to formulate another set of objective functions. By optimising the objective functions, a manipulator pose can be determined whereby the inspection of a surface is enabled so that the material-type can be identified. Each stage is optimised using an appropriate optimisation technique to search for the next viewpoint.

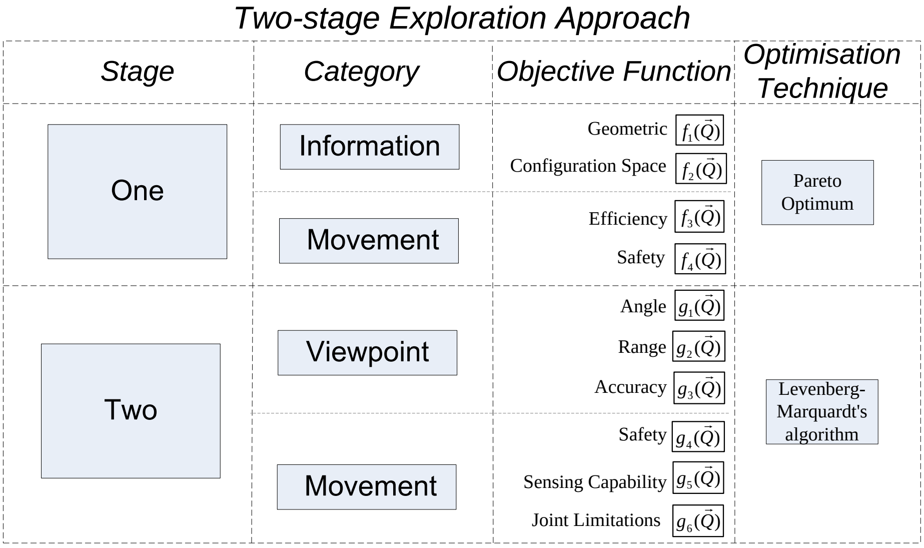


FIGURE 3.1: Outline of the two-stage exploration approach showing the categories which each of the objectives falls into, and the named objective functions.

A breakdown of the objective functions in the two-stage exploration approach is shown in Fig. 3.1. The output of each stage of exploration is a viewpoint for a laser range finder sensor, which is mounted at the manipulator's end-effector, to further explore the environment. Each stage consists of a number of viewpoints selected with the aim to reduce the measure of information remaining in the environment. A widely accepted approach to information measurement [157] is entropy. In accordance with Shannon's entropy [158], the remaining information in an environment can be measured. The information of interest in Stage One concerns the geometry and C-space. This information gathered in Stage One will be used by Stage Two. Objective cost functions will be formulated then minimised so Stage Two determines viewpoints which enable the surface material-type to be identified. Each viewpoint is realised by a valid pose of the manipulator, $\vec{Q} = [q_1, q_2, \dots q_6]^T$. A valid pose is one which lies in the C-space of the manipulator, within the joint limitations, and keeps the manipulator in safe, known freespace.

3.2 Stage One: Geometric Exploration

Viewpoints in Stage One of the exploration approach need to be selected to allow a sensor to gather both accurate geometric and manipulator C-space information. The geometric and C-space information are related, and both are necessary. Exploring an unknown geometric region that is within the manipulator's workspace reduces the remaining geometric and C-space information at different rates. A region of Euclidean space can contain significant C-space information if the region's occupancy state is unknown, and because its state is unknown, many manipulator poses cannot be used in trajectory planning due to possible environment-manipulator collisions. However, if the unknown region only affects a few manipulator poses or is outside the manipulator's workspace, then the C-space information remaining is small or non-existent.

The viewpoints determined by Stage One must be within known geometric freespace (i.e. from a safe C-space node). Therefore, some geometric information gathered at each viewpoint may be redundant and can be discarded by the data reduction process. As well as information reduction, manipulator movement efficiency and manipulator safety must also be considered when selecting viewpoints.

In order to have a successful exploration approach, Stage One needs to take the following four objectives into account: (1) the geometric information remaining ($f_1(\vec{Q})$), (2) C-space information remaining ($f_2(\vec{Q})$), (3) efficient manipulator movements ($f_3(\vec{Q})$), and (4) manipulator safety ($f_4(\vec{Q})$). These four objectives form the objective functions of the geometric and C-space exploration problem. The multiple objectives must be combined and solved as a multi-objective optimisation problem, such that Pareto-optimality [159] is assured. The Pareto front optimisation technique will be explained later in this section along with the constraints on the manipulator which must be satisfied. The formulation of the information-related objective functions, including detailed modelling of information remaining in an environment, is presented in Chapter 4, while the map representation which enables this will be the focus of Chapter 5.

3.2.1 Objective Functions

3.2.1.1 Geometric Information Reduction

The first objective function, $f_1(\vec{Q})$, is a measure of the amount of geometric information remaining after sensing the environment from a viewpoint. Thus, a set of possible contending viewpoints, $\mathbf{Q}$, can be compared so as to select the viewpoint which minimises the amount of geometric information remaining in an environment.

$$\min_{\vec{Q}\in\mathbf{Q}} f_1(\vec{Q}) = \min_{\vec{Q}\in\mathbf{Q}} \mathbf{H}_1(\mathbf{X}_{\vec{Q}}) \tag{3.1}$$

Minimising Equation 3.1 over a solution set of viewpoints facilitates the pursuit of geometric information. The geometric information objective function, $f_1(\vec{Q})$, measures the information remaining (i.e. uncertainty), $\mathbf{H}_1(\mathbf{X}_{\vec{Q}})$, in terms of the uncertainty with the Euclidean space state variable for an environment which has n_v voxels, $\mathbf{X}_{\vec{Q}}$, after sensing from the viewpoint with pose, $\vec{Q}$. This information theory-formulation/ model will be detailed in Chapter 4. The model will make it possible to measure the remaining information, $\mathbf{H}_1(\mathbf{X}_{\vec{Q}})$, in an environment in terms of the geometry. In order to determine how much information can be gained from a viewpoint, it is necessary to predict areas which are obscured by object surfaces (i.e. occlusions) and areas which contain freespace. The goal is to Maximise the Entropy Reduction (MER) as shown in [3], which equates to reducing or minimising the remaining information at each viewpoint.

3.2.1.2 Configuration Space Information Reduction

As was shown in the related work [3][116], the exploration of C-space can result in a greater freedom of safe movements for the manipulator. Hence, the necessity to reduce the C-space information remaining is included as the second objective function, $f_2(\vec{Q})$. Similar to the previous objective function, a set of viewpoint solutions, $\mathbf{Q}$, can be compared based upon how much C-space information is predicted to remain after sensing from a viewpoint.

$$\min_{\vec{Q}\in\mathbf{Q}} f_2(\vec{Q}) = \min_{\vec{Q}\in\mathbf{Q}} \mathbf{H}_2(\mathbf{C}_{\vec{Q}}) \tag{3.2}$$

As introduced in Chapter 2, a node in C-space corresponds to a manipulator pose. Obstacles in Euclidean space (the 3D space where sensing actually occurs), which by definition also lies inside the manipulator's workspace, could result in one or many C-space obstacles. Chapter 4 will investigate the transformation (i.e. mapping) of obstacles and unknown, unexplored Euclidean space into C-space obstacles. The relationship between the Euclidean space and C-space will be based upon the probability that a specific voxel (i.e. volumetric pixel) is inside the volume of space occupied by a manipulator when in a pose (i.e. at a node in C-space). Unknown voxels which are in close proximity to the manipulator's base are expected to be related to many C-space nodes, which suggest they contain a relatively large amount of C-space information. Sensing the occupancy state of other unknown voxels, such as those outside the workspace of the manipulator, reduces the geometric information remaining only, without affecting the C-space information remaining. Thus, in cases where a proposed viewpoint can significantly reduce the geometric information ($f_1(\vec{Q})$) outside the manipulator's workspace, there may be conflict between the first and second objective functions. This conflict between objectives is best approached as a multi-objective optimisation problem. Using the devised models it will then be possible for the remaining information, $\mathbf{H}_2(\mathbf{C}_{\vec{Q}})$, about a configuration space, $\mathbf{C}_{\vec{Q}}$, after sensing from the viewpoint, $\vec{Q}$, to be estimated. The MER criterion is once again pursued to reduce the C-space entropy at each viewpoint. The second objective function, $f_2(\vec{Q})$, thus allows possible solution viewpoints to be ranked in order of the predicted reduction of the C-space information remaining.

3.2.1.3 Manipulator Movement Minimisation

At every iteration of the exploration approach a viewpoint must be determined. Ideally, the manipulator pose corresponding to the current viewpoint will be similar in C-space to the pose corresponding to the next viewpoint. Therefore the third objective, $f_3(\vec{Q})$, aims to minimise the movement of the 6DOF manipulator between the manipulator pose, $\vec{Q}_k$, at viewpoint k, to the pose, $\vec{Q}_{k+1}$ at viewpoint $k+1$. The ith joint (of the 6DOF manipulator) at viewpoint k, is $q_{i,k}$. The movement is the sum over all joints between the current pose and the next pose. Therefore, the third objective function is formulated as

$$\min_{\vec{Q}\in\mathbf{Q}} f_3(\vec{Q}) = \min_{\vec{Q}\in\mathbf{Q}} \sum_{i=1}^{6} |q_{i,k} - q_{i,k+1}| \tag{3.3}$$

Planning a manipulator's trajectory between two poses is a well established problem even with many obstacles. Given the links among nodes in C-space, searching for trajectories can be performed based on popular algorithms such as A*.

3.2.1.4 Safe Manipulator Pose Selection

The safety of manipulator movement is another vital consideration during exploration. Safe movements are those that can avoid any intersection (collision) of the manipulator with obstacles or with unexplored space. In order to improve the efficiency of collision-free pose planning, ellipsoidal virtual bounding fields [126][128] can be used to perform safety checks. The ellipsoidal field of the ith manipulator link has the centre vector $\mathbf{p}_{c,i}$, a vector of equatorial radii $[a_{e,i}, b_{e,i}]$, and a polar radius, $c_{e,i}$.

The position and orientation of an ellipsoid in the global coordinate frame, which represents a link, i, of the manipulator is specified by the 4×4 rotation and translation homogeneous transformation matrix, ${}^{0}T_i(\vec{Q})$, calculated as

$${}^{0}T_i(\vec{Q}) = \prod_{j=1}^{i} {}^{j-1}T_j(q_j) \tag{3.4}$$

Although this transformation for the ith ellipsoid is presented as a function of the pose, $\vec{Q}$, only the first i joints are used when transforming the ith coordinate frame. Each voxel containing unexplored space or an obstacle is represented by a position vector, $\mathbf{p} \in \mathsf{P}$, where P is a set of points (positions) in the environment. A $\mathbf{p}$ can be transformed into the ith ellipsoid's coordinate frame (where the ellipsoid is at the origin and an obstacle is positioned and orientated relatively) by using the inverse of the transformation matrix. A transformed point is thus denoted as $\mathbf{p}^{{}^{0}T_i(\vec{Q})^{-1}}$. The algebraic distance can then be calculated from the ellipsoid to each point.

For a manipulator pose $\vec{Q}$, and for each of the $i \in \{1, \ldots 6\}$ joints in a pose, it is necessary to determine how safe the pose is. The minimum algebraic distance, $dist(\vec{Q})$, is then a function of $\vec{Q}$ and for points $\mathbf{p} \in \mathsf{P}$, and all a manipulator's encompassing ellipsoids (after the joints $i \in \{1, \ldots 6\}$) as follows

$$dist(\vec{Q}) = \min_{\mathbf{p} \in \mathsf{P}} \left(\min_{i \in \{1, \ldots 6\}} \left(\left(\mathbf{p}^{{}^0T_i(\vec{Q})^{-1}} - \mathbf{p}_{c,i} \right)^T \cdot \begin{pmatrix} a_{e,i}^{-2} & 0 & 0 \\ 0 & b_{e,i}^{-2} & 0 \\ 0 & 0 & c_{e,i}^{-2} \end{pmatrix} \cdot \left(\mathbf{p}^{{}^0T_i(\vec{Q})^{-1}} - \mathbf{p}_{c,i} \right) \right) \right) \tag{3.5}$$

The fourth objective function, $f_4(\vec{Q})$, is to choose a viewpoint, $\vec{Q}$, which maximises the smallest algebraic distance ($dist(\vec{Q})$) from all a manipulator's ellipsoids to all obstructing points. If a bounding ellipsoidal field around the manipulator at a viewpoint had its smallest algebraic distance less than unity ($f_4(\vec{Q}) \leq 1$), it is unacceptable since this means a collision will occur or the manipulator will enter unexplored space. The fourth objective function ensures that safe poses and motions are considered desirable, and constrains the viewpoint solution to being safe poses (i.e. $f_4(\vec{Q}) > 1$). The most appealing viewpoint has the maximum value of $f_4(\vec{Q})$, as follows

$$\max_{\vec{Q} \in \mathbf{Q}} f_4(\vec{Q}) = \max_{\vec{Q} \in \mathbf{Q}} dist(\vec{Q}). \tag{3.6}$$

The four objective functions presented in Stage One have been formulated for different purposes. Therefore, the viewpoint order based upon each objective function does not necessarily relate to each other. A proposed viewpoint which minimises the remaining geometric information ($f_1(\vec{Q})$), may require an inefficient movement ($f_3(\vec{Q})$), which is unsafe ($f_4(\vec{Q})$). Alternatively, a very safe trajectory to a contending viewpoint may not reduce either the remaining geometric or the C-space information ($f_1(\vec{Q})$, $f_2(\vec{Q})$) and may require a particularly inefficient movement ($f_3(\vec{Q})$). It is therefore reasonable to assume that objectives may be in conflict. An optimisation approach is required to combine them so that a viewpoint can be selected.

3.2.2 Multi-objective Optimisation

In order to generate a quality viewpoint which satisfies the multiple objectives of Stage One of exploration, the problem is best solved as a Multi-objective Optimisation Problem (MOP)[159]. Clearly, in this case the objectives do not share common units and may even conflict as discussed before. The Multi-objective Optimisation Problem (MOP) is formally defined as:

Find a set of viewpoints each with the joint vector, $\vec{Q}^* = [q_1^*, q_2^*, \dots q_6^*]^T$*, to satisfy the joint limitation and safety constraints, and optimise the vector function of the objective functions*

$$\mathbf{f}(\vec{Q}) = [f_1(\vec{Q}), f_2(\vec{Q}), f_3(\vec{Q}), f_4(\vec{Q})]^T \tag{3.7}$$

The Pareto front design approach, which is a popular methodology in engineering optimisation problems [160], has been applied in this research to solve the multi-objective exploration problem. Pareto optimality is a concept that formalises the trade-off between a given set of mutually contradicting objectives. A solution is Pareto optimal when it is not possible to improve one objective without deteriorating at least one of the other. A set of Pareto optimal solutions constitute the Pareto front. Therefore, given a set of viewpoint choices, $\mathbf{Q}$, which denotes the feasible region of the problem (solution space) the Pareto front can be built. A vector of decision variables corresponding to a contending viewpoint, $\vec{Q}^* \in \mathbf{Q}$, is a Pareto optimal viewpoint if there is no, $\vec{Q}^{**} \in \mathbf{Q}$, such that

$$f_i(\vec{Q}^*) \geq f_i(\vec{Q}^{**}) \forall i \in \{1, 2, 3, 4\} \tag{3.8}$$

and there is at least one objective value where $f_i(\vec{Q}^*) > f_i(\vec{Q}^{**})$. Therefore by definition, the viewpoints (i.e. decision variable), $\vec{Q}^*$, are Pareto optimal if no feasible viewpoint exists, $\vec{Q}^{**} \in \mathbf{Q}$, which would decrease some criteria without causing a simultaneous increase in at least one other criterion. Intuitively this multi-objective optimisation approach will generally give a set of solutions (i.e. the Pareto optimal set [159]). Each vector, $\vec{Q}^*$, corresponding to a viewpoint included in the Pareto optimal set is both viable and desirable.

The ranks in terms of the objective functions are summed for each contending viewpoint in the Pareto optimal set. In nearly all cases the sum of the ranks of the contending solutions will allow a clear 'winner' pose to be selected. This pose will have the lowest sum of ranks over the specified optimisation functions. However, in the rare case where multiple solutions result with the same aggregated rank and since only one pose can be selected, the candidate poses are differentiated based upon their reduction of information remaining, i.e., the viewpoint which minimises the geometric information remaining, $f_1(\vec{Q})$) is selected.

In order to have a solution space, $\mathbf{Q}$, then n_Q contending viewpoints must be sampled from the manipulator's C-space. The efficiency of exploration can be improved by reducing the number of contending viewpoints in the solution space. Fewer contending viewpoints mean less objective function computations. However, this must be balanced so that there is adequate convergence of the C-space. A brief investigation into the effect of varying n_Q is presented henceforth, by varying the number of viewpoints and comparing the geometric information coverage of the combined contending viewpoints.

3.2.3 Viewpoint Solution Space Sampling

For Stage One of the presented exploration approach, it is intractable to move to all possible viewpoints, $\vec{Q}$. For a typical industrial manipulator (Appendix 3) which can have tens of thousands of discrete angular steps per joint, there are over 10^{24} possible manipulator pose solutions in C-space. In order to generate a manageable number of contending viewpoint solutions, the C-space must be sampled to develop a solution space, $\mathbf{Q}$. An increase in the sample size n_Q (i.e. number of nodes in C-space), can theoretically lead to a greater reduction in the geometric information. Each viewpoint can potentially enable the geometric information to be somewhat reduced, in effect 'covering' this information. Collectively, the n_Q contending viewpoints should cover the information remaining in a representative manner, as if all possible manipulator poses in C-space were used. The trade-off is between the efficiency gained by limiting the number of samples, and the information coverage which is possible by the contending viewpoints. While a solution is not guaranteed to be global optimum, it is satisfactory for the requirements of efficiency and real-time performance which are vital for this particular application.

An investigation was undertaken regarding how the number of C-space nodes, n_Q, affect the geometric information coverage of the collective contending viewpoints. The experiment devised for this tested 650 different C-space sample sizes n_Q, where the samples were uniformly distributed in C-space. The sampling sizes were determined based upon the sensor resolution/ range and the obstacle dimensions in the application environment considered in this research. The measurement of information remaining if all samples were used is compared as n_Q changes. This gives an indication for a typical environment of the potential sensing coverage possible from a set of viewpoints. The information remaining is based upon ray casting predictions [113] and geometric information measurement techniques (Chapter 4). This allowed a prediction of the information remaining to be calculated, assuming all viewpoints are used.

The different sample sizes of $\mathbf{Q}$ checked are between $n_Q = 50$ and $n_Q = 3500$. A coarse sampling resolution gives a smaller sample size, while a fine sampling resolution gives a larger sample size. Fig. 3.2 displays the results.

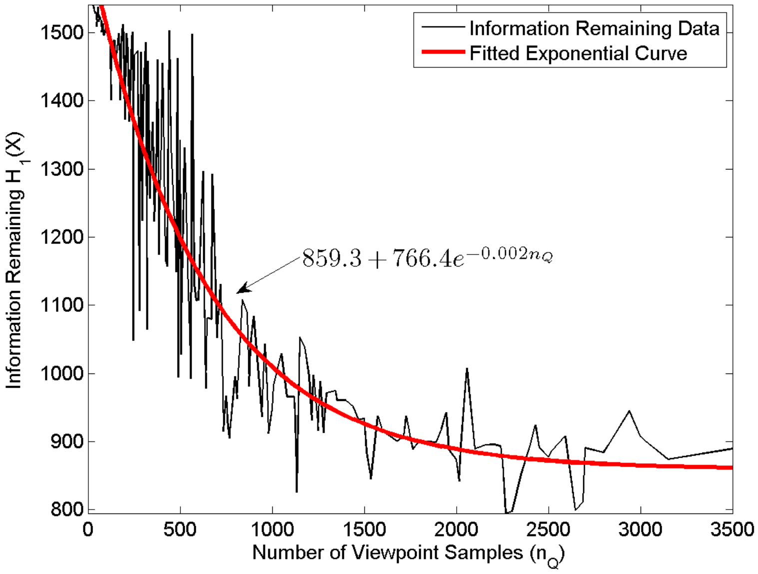


FIGURE 3.2: The predicted coverage of information remaining, assuming all n_Q contending viewpoints are used. An exponential curve is fitted to the data so as to observe convergence characteristics.

Fig. 3.2 shows that the information remaining is generally reduced with an increase in the number of viewpoints n_Q. This figure also shows that a higher sampling size results in a more consistent reduction in the geometric information remaining. An exponential curve was fitted to the data, which shows that increasing n_Q any further than this range is not expected to significantly improve the predicted reduction of the information remaining. It was observed that due to sensing occlusions in an environment containing objects, it may be impossible to gather all the remaining information. Within the viewpoints, there are often some key viewpoints which gain significantly more information than other viewpoints. It was also observed during the experiment that within any set of C-space samples (i.e. viewpoints), a significant number are often not feasible since the pose could cause the manipulator to collide with the environment. The combination of a limited number of feasible viewpoints, and the characteristic that a small number of viewpoints can exist which reduce the information significantly, necessitates that n_Q is sufficiently large enough to allow a reasonable cross-section of contending viewpoints to be compared. Thus, the number of contending viewpoints, n_Q, can be selected to be the solution space for Stage One of exploration. The contending viewpoints can then be efficiently compared via the multi-objective Pareto optimisation approach.

3.2.4 Stage One Termination

The focus of Stage One is on the minimisation of information remaining, $f_1(\vec{Q})$ and $f_2(\vec{Q})$, and efficient safe manipulator movements. Ideally, the information remaining ($\mathbf{H}_1(\mathbf{X})$ and $\mathbf{H}_2(\mathbf{C})$), of an environment's geometric and C-space state ($\mathbf{X}$ and $\mathbf{C}$) would reach 0. Realistically, this is intractable due to occlusions in the environment. Therefore, Stage One of exploration must continue to determine viewpoints until the information remaining can be assumed to converge to a minimal value.

Stage One presents a smooth unconstrained function problem where the information remaining decreases at each sensing viewpoint, therefore first-order optimality can be used to terminate the optimisation. In accordance with Nocedal and Wright [161], first-order optimality is a measure of how close a value (i.e. an information remaining measurement) is to reaching its optimal. The optimality measure is the infinity-norm (i.e. the maximum absolute value) of the change in information remaining, $\nabla\mathbf{H}_1(\mathbf{X})_k$ and $\nabla\mathbf{H}_2(\mathbf{C})_k$, after the kth viewpoint. This measure of optimality is based on the condition that for a smooth function to achieve a minimum, "its gradient must be zero". When the first-order optimality measures of Stage One are nearly zero, $\nabla\mathbf{H}_1(\mathbf{X})_k \approx 0$ and $\nabla\mathbf{H}_2(\mathbf{C})_k \approx 0$, the gradients (over a change in k) are nearly zero, so the objective functions could be nearly minimised [161]. If this first-order optimality measure is not small, the information remaining is not minimised. Hence, the gradient of the geometric and C-space information remaining is monitored. The information is measured prior to exploration as $\mathbf{H}_1(\mathbf{X})_0$ and $\mathbf{H}_2(\mathbf{C})_0$, respectively. When the gradient of information change at viewpoint k is less than 1% of the environment's information (as measured prior to the commencement of exploration for both geometric information), $\nabla\mathbf{H}_1(\mathbf{X})_k < \frac{\mathbf{H}_1(\mathbf{X})_0}{100}$, and for C-space information, $\nabla\mathbf{H}_2(\mathbf{C})_k < \frac{\mathbf{H}_2(\mathbf{C})_0}{100}$, then Stage One is terminated.

3.3 Stage Two: Material-type Exploration

After Stage One has been completed, it is assumed that the information remaining about the environment's geometry and the manipulator's C-space will not be reduced significantly with additional viewpoints. The sensor data gathered after each viewpoint of Stage One is translated into global coordinates and fused with previous data, then n_{ss} small SLD material-type identification targets are generated. Each SLD consists of a 3D centre vector point and is normal to the surface (Chapter 5). In the first stage of exploration (Stage One), some surfaces also had their material-types identified by means of a Laser Range Classifier (LRC) system (Appendix B) which analyses the laser scanner data. However, since material-type identification was not an objective of Stage One, and the constraints of the identification technique were not considered, a significant amount of material-type information remains to be gathered. In Stage Two of the exploration approach, the aim is to determine appropriate sensing viewpoints such that the material-type of each surface can be identified.

In general, laser range scanners, which are used to determine the geometry of an environment, operate by emitting light and receiving reflections off the surface, scaling them based on their return intensity and then measuring phase shift. This enables a range to an object's surface to be determined. It was shown in our research, [5], that surface material-type identification is achievable through the study of the physical properties that affect the light reflected from a surface. Further examination of the physical properties which enable identification is outside the scope of this thesis but is briefly presented in Appendix B. The system which processes the range and intensity data to identify the surface material-type is named the LRC. The LRC is able to produce surface material-type identifications using the intensity (which the laser range scanner can be specially configured to return [41]) and range curves. However, the LRC has specified constraints which must be considered when determining a sensing viewpoint. Therefore, appropriate manipulator joint angle vectors, $\vec{Q}$, must be calculated so that the sensor's position and orientation is appropriate.

At each sensing instance in Stage Two, the data about the material-type that is returned from the LRC is fused with data already collected to update the probability of the surface material-type states. This is known as updating the material-type probabilistic belief. This update theoretically results in a reduction of the uncertainty about a surface's material-type, and also the material-type information remaining in an environment, $\mathbf{H}_3(\mathbf{M})$. The probabilistic belief updates, along with the information theory-based measurements of material-type information remaining are detailed in Chapter 4. The information remaining on each of $i \in \{1, \dots n_{ss}\}$ small SLDs which are required to have their material-type identified can be given a measure, $\mathbf{H}_3(\mathbf{M}_i)$. In Stage Two, with the aim to reduce the material-type information remaining, the SLD target with the most information remaining is selected to be identified. Then a manipulator pose, $\vec{Q}$, that corresponds to an adequate sensing viewpoint for the targeted SLD must be determined.

Inverse kinematics to determine $\vec{Q}$ would be appropriate if there were no obstacles and there was only one possible viewpoint. However, there is a strong possibility of obstacles and theoretically there are an infinite number of possible viewpoints that are within the LRC constraints. Therefore, pose selection must be used to find a pose which corresponds to a desirable viewpoint. In order to perform pose selection, a relationship between the pose $\vec{Q}$ and the quality of the resulting viewpoint is established. Through forward kinematics it is possible to compute the position and orientation of the manipulator's end-effector where the sensing tool is mounted. This is expressed in the homogeneous transformation matrix, ${}^0T_f(\vec{Q})$, by performing transformations through the $i \in \{1, \ldots 6\}$ links for each of the joint angles, q_i.

$$
{}^0T_f(\vec{Q}) = \prod_{i=1}^{6} {}^{i-1}T_i(q_i) \tag{3.9}
$$

The transformation between the end-effector and the mounted sensor is given by ${}^fT_s$. Together these two matrices describe the position and orientation of the sensor (i.e. a viewpoint) in the robot base coordinate frame ${}^0T_s(\vec{Q}) = {}^0T_f(\vec{Q}){}^fT_s$. Then where the sensor position is $[x, y, z]^T$, the centre laser ray is in the direction of the vector $[a_x, a_y, a_z]^T$, the ray's axis of rotation is $[n_x, n_y, n_z]^T$, and the ±90° laser ray direction is $[o_x, o_y, o_z]^T$, the transformation matrix is,

$$
{}^0T_s(\vec{Q}) = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}
$$

Relative to the sensor, the parameters for surface material-type identification (i.e. the LRC constraints) that must be considered during pose selection are shown in Fig. 3.3.

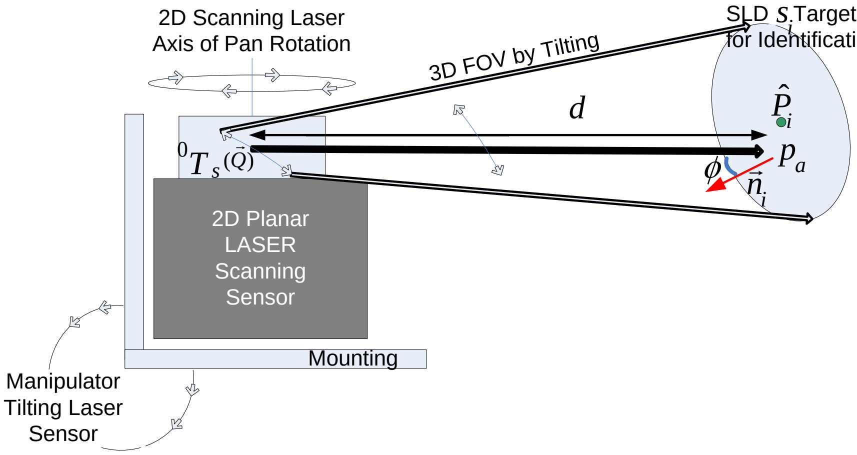


FIGURE 3.3: Pose selection parameters in surface material-type identification (distance to SLD targeted, d, surface centre point, $\hat{P}_i$, actual sensor's centre ray surface intersection point, $\mathbf{p}_a$, and angle, ϕ, to surface normal $\vec{n}_i$). A 3D Field-Of-View (FOV) is attained by tilting the 2D laser range finder with the 5th joint of the manipulator.

The targeted ith SLD, s_i, has a surface centre point $\hat{P}_i$ and a surface normal $\vec{n}_i$. There is a direct relationship between the pose $\vec{Q}$ and the values of the parameters in the figure. The position of the sensor is the 3D vector from column four of the sensor's homogeneous transformation matrix, $\mathbf{p}_1 = [x, y, z]^T$, and the laser scanning sensor's centre ray orientation is the 3D vector from column three of the transformation matrix $\mathbf{p}_2 = [n_x, n_y, n_z]^T$. The sum of the position of the sensor, $\mathbf{p}_1$, and the centre ray, $\mathbf{p}_2$, gives a point on the centre scan ray, $\mathbf{p}_3 = \mathbf{p}_1 + \mathbf{p}_2$. Thus, the three following parameters can be calculated. The laser scanner's centre ray intersection point with the SLD is

$$\mathbf{p}_a = \mathbf{p}_1 - (\mathbf{p}_3 - \mathbf{p}_1)\frac{\vec{n}_i \cdot (\hat{P}_i - \mathbf{p}_1)}{\vec{n}_i \cdot (\mathbf{p}_3 - \mathbf{p}_1)} \tag{3.11}$$

then distance from the sensor to point of intersection is

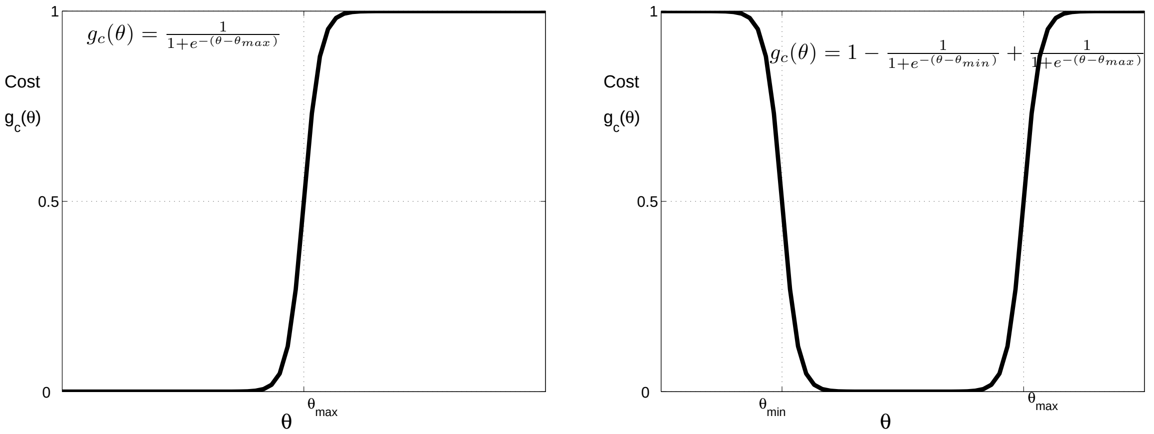
$$d = \|\mathbf{p}_a - \mathbf{p}_1\| \tag{3.12}$$

where $\|\cdot\|$ is the Euclidean distance. The angle between the laser scanner's centre ray and the SLD's surface normal is given by

$$\phi = \arccos(\vec{n}_i \cdot \mathbf{p}_2). \tag{3.13}$$

3.3.1 Cost Functions

For surface material-type identification, the expected sensing area is small and there are several constraints which must be incorporated. An optimisation approach similar to Stage One, where a sample of the solution space is selected and the contending viewpoints are ranked, cannot be used. This is due to the high dimensionality of the problem, making it intractable to sample the solution space with a fine enough resolution to ensure a valid viewpoint. Instead, the viewpoint determination process requires a fine tuning optimisation approach. Therefore, each objective function will be formulated as a "cost function", g_c, by incorporating material-type identification objectives in sigmoid functions based on target tracking pose selection research [43]. A sigmoid function is appropriate since it enables both minimum and/or maximum "cost functions" to be formulated. If for a general constraint case, an angle θ, is required to be less than a maximum angle θ_{max}, then this can be incorporated into a sigmoid to give a cost function $g_c(\theta) = \frac{1}{1+e^{-(\theta-\theta_{max})}}$, as shown in Fig. 3.4$a$. If the solution is required to be greater than an angle θ_{min}, then $g_c(\theta) = 1 - \frac{1}{1+e^{-(\theta-\theta_{min})}}$. If the acceptable angular value is between two angles, θ_{min} and θ_{max}, then $g_c(\theta) = 1 - \frac{1}{1+e^{-(\theta-\theta_{min})}} + \frac{1}{1+e^{-(\theta-\theta_{max})}}$, as shown in Fig. 3.4$b$.



(a)

(b)

FIGURE 3.4: Sigmoid functions for a cost function, $g_c(\theta)$.
a) Cost function for a maximum constraint e.g. $\theta < \theta_{max}$.
b) Cost function for minimum and maximum constraints e.g. $\theta_{min} < \theta < \theta_{max}$.

An objective function formulated as a sigmoid function is continuous with a range between 0 and 1, and can be optimised (minimised) by Levenberg-Marquardt's (LM) algorithm [162]. The LM algorithm interpolates between the Gauss-Newton algorithm and the method of gradient descent, and is primarily used in the least squares curve fitting problem, meaning that multiple objective functions can be combined by summing their squares. Therefore, the multiple Stage Two objectives for a viewpoint of a target surface and a manipulator pose will be formulated, and then combined. The combined objective functions can then be optimised so as to determine a pose $\vec{Q}$, corresponding to an appropriate viewpoint which enables the surface material-type to be identified.

3.3.1.1 Viewpoint-to-Surface Orientation

The laser sensor's centre ray meets an object's surface with an angle of incidence. This angle affects the reflection received by the sensor and consequently how well the material-type can be identified by the LRC system. In order for the identification to be successful, the angle formed with the target surface normal, ϕ, must be less than a maximum angle, ϕ_{max}. Adherence to this constraint improves the central laser ray's reflection and thus the performance of the LRC. ϕ is calculated as the arccosine of the dot product of the two unit vectors: the normal to the target SLD and the laser scanner's centre ray (Equation 3.13). The first cost function, $g_1(\vec{Q})$, is formulated with the maximum constraint as a sigmoid function

$$\min_{\vec{Q}} g_1(\vec{Q}) = \frac{1}{1 + e^{-(\phi - \phi_{max})}} \tag{3.14}$$

As with all the objective functions in Stage Two, $g_1(\vec{Q})$ must be minimised. The minimisation results in an appropriate manipulator pose, $\vec{Q}$, being determined such that the sensor's rays have an acceptable angle of incidence to the surface.

3.3.1.2 Viewpoint-to-Surface Range

As shown in Fig. 3.3 the viewpoint range, d, is the distance from the sensor to the SLD which is being targeted for material-type identification. This ideally falls between the sensing range constraints, d_{min} and d_{max}, as required by the LRC to perform surface material-type identification. d is the distance between the position of the laser scanning sensor and the SLD centre point $\hat{P}_i$, as shown in Equation 3.12. This is formulated as the second cost function, $g_2(\vec{Q})$, where a sigmoid incorporates two constraints as follows

$$\min_{\vec{Q}} g_2(\vec{Q}) = 1 - \frac{1}{1 + e^{-(d - d_{min})}} + \frac{1}{1 + e^{-(d - d_{max})}} \tag{3.15}$$

In the case where the distance to the surface is too close (i.e. $d < d_{min}$), the laser range sensor will not provide valid readings. If the distance is too great (i.e. $d > d_{max}$), then the sensor data about the surface will be too sparse for the LRC system to successfully perform material-type identifications [5]. Minimising the second cost function, $g_2(\vec{Q})$, ensures that the viewpoint range, d, satisfies the LRC's constraints: $d_{min} < d < d_{max}$. Viewpoints outside of these constraints are not accepted.

3.3.1.3 Viewpoint-to-Targeted Surface Centre Accuracy

The sensor must be aimed at the specific target SLD accurately (Fig. 3.3) so that the surface material-type can be correctly identified. The requirement to accurately aim at the centre of the targeted SLD is separate from the viewpoint-to-surface range objective ($g_2(\vec{Q})$). The laser scanner's centre ray vector intersects with the SLD plane at $\mathbf{p}_a$, according to Equation 3.11. The centre of the ith SLD target is at $\hat{P}_i$. The minimum allowable tolerance in distance between $\mathbf{p}_a$ and $\hat{P}_i$ is ρ_{min}. This third objective, $g_3(\vec{Q})$, is formulated as

$$\min_{\vec{Q}} g_3(\vec{Q}) = 1 - \frac{1}{1 + e^{-(\|\mathbf{p}_a - \hat{P}_i\| - \rho_{min})}} \tag{3.16}$$

where $\| \cdot \|$ is the Euclidean distance.

3.3.1.4 Safe Manipulator Pose Selection

The safety of the manipulator must also be ensured by pose selection optimisation in Stage Two. Collision avoidance is implemented using the ellipsoidal bounding fields around each manipulator link, based upon the combination of research [126] and [128] similar to Stage One, where only the safe, feasible solutions are considered in the implementation. The ith manipulator joint is once again enclosed by ellipsoidal virtual bounding fields, centred at $\mathbf{p}_{c,i}$, and with parameters $[a_{e,i}, b_{e,i}, c_{e,i}]$. Since an obstacle (or unexplored voxel), $\mathbf{p} \in \mathsf{P}$, within an ellipsoid has an algebraic distance less than 1, the minimum constraint for the sigmoid is 1. For each of the six joints, q_i in $\vec{Q}$, the corresponding ellipsoid's algebraic distance, $dist(\vec{Q})$, to all obstacles and unknown voxels is returned using Equation 3.5. Then this can be used in the fourth sigmoid cost function.

$$\min_{\vec{Q}} g_4(\vec{Q}) = 1 - \frac{1}{1 + e^{-\left(dist(\vec{Q}) - 1\right)}} \tag{3.17}$$

3.3.1.5 Field-of-View Sensing Constraint

The 2D laser range sensor must be tilted with the relevant manipulator joint, q_t (which is the fifth joint in this case). Tilting is done so that the 2D sensing plane is moved through a 3D Field Of View (FOV), as shown in Fig. 3.3. In order to allow for an appropriate amount of angular tilting, q_t is parameterised to facilitate tilting the sensor through a 3D scan so as to sense the target surface. This objective is incorporated as the fifth sigmoid cost function to be minimised.

$$\min_{\vec{Q}} g_5(\vec{Q}) = \frac{1}{1 + e^{-(q_t - q_{t,max})}} \tag{3.18}$$

The resulting $\vec{Q}$ is only valid (i.e. can gather the required 3D FOV) if the tilting joint, q_t, is less than the specified $q_{t,max}$. Adhering to the 3D FOV constraint provides enough measurements of the targeted SLD to enable material-type identification with the LRC system.

3.3.1.6 Manipulator Joint Limitations

For a viewpoint, the manipulator joints (i.e. pose) are described by the vector of joints, $\vec{Q}$, which must fall within the physical angular limitations. For the $i \in \{1, \ldots 6\}$ joints, these are defined as the positive maximums, $q_{i,max}$, and negative minimums, $q_{i,min}$. In Stage One of exploration the solution space, $\mathbf{Q}$, was sampled so all solutions fell within the joint limits. In the case of Stage Two's pose selection, where the optimisation approach is different, physical joint limitations must be parameterised so the viewpoint solution is valid. The joint hard limits are enforced in implementation to ensure compliance. Since the LM algorithm used for Stage Two solves the cost functions by minimising the sum of the squares, this objective function, $g_6(\vec{Q})$, has combined the six pairs of joint constraints into one function by summing the squares as follows

$$\min_{\vec{Q}} g_6(\vec{Q}) = \sum_{i=1}^{6} \left(1 - \frac{1}{1 + e^{-(q_i - q_{i,min})}} + \frac{1}{1 + e^{-(q_i - q_{i,max})}} \right)^2 \tag{3.19}$$

3.3.2 Non-linear Optimisation

A manipulator joint vector, $\vec{Q}$, must be determined to achieve the viewpoint which enables the targeted SLD to be appropriately sensed. Appropriate sensor data will allow the LRC system to identify the material-type. In order to ensure successful identification, the constraints of the LRC and the manipulator have been formulated into objective functions. The LRC constraints include the sensor range and accuracy limitations, and the necessity to have the sensor's central ray facing the target SLD with a minimum angle of incidence. Additionally, limitations of the manipulator and safety considerations have been included in the pose selection. The objectives have been formulated by using sigmoid functions so the acceptable cost range is between zero and a half (Fig. 3.4). In Stage Two, the problem can be solved as a non-linear optimisation problem rather than a MOP, as was the case in Stage One.

The non-linear optimisation problem is formulated so that it can be solved by the Levenberg-Marquardt (LM) algorithm [162], which interpolates between the Gauss-Newton algorithm and the method of gradient descent. This gradient-based pose selection using least squared optimisation is modified from a manipulator based probabilistic tracking solution in [43], and originally from camera-based viewpoint planning research [103]. LM is an established iterative technique which can be used to locate the minimum of the sum of squares of a non-linear function, $\mathbf{g}(\vec{Q})$, as shown in [163]. The LM algorithm is used to determine the step direction towards $\vec{Q}$, using the Jacobian of the vector of cost function, $\mathbf{g}(\vec{Q})$, and to return the new joint angles. The combined cost function is thus defined as

$$\min_{\vec{Q}} \mathbf{g}(\vec{Q}) = \sum_{i=1}^{6} \left(g_i(\vec{Q}) - \hat{g}_i \right)^2 \tag{3.20}$$

where the desired value, $\hat{g}_i$, for all $i \in \{1, \dots 6\}$ cost functions is zero (i.e. $\hat{g}_i = 0$). The pose selection algorithm is designed for the rapid discovery of a valid joint configuration for a viewpoint of a targeted SLD that contains significant material-type information. Chapter 4 presents details of the material-type belief update process and the resulting information remaining measurement. The SLD target generation is presented in Chapter 5. Relevant experimental results on an implemented system are presented in Chapter 6 of this thesis.

3.4 Conclusions

This chapter has introduced a two-staged approach to the problem of exploring unknown environments with a manipulator. The exploration and mapping approach presented includes four objectives in Stage One, focused on geometry and C-space information, and manipulator movements and safety. Stage Two includes six objectives so that surface material-types can be identified. The objective functions formulated are solved using established optimisation techniques appropriate to the problem. The following chapters complement the exploration approach presented in this chapter by detailing the measurement of information and describing the map representation.

Chapter 4

Information Measurement-based Exploration

When exploring an unknown environment there is often minimal or no *a priori* information (i.e. no map) about the environment's surface geometry and material-type. In manipulator-based exploration, an additional consequence of the unknown geometry is that no information is available about the free areas of the manipulator's configuration space (C-space). Each valid manipulator pose (i.e. a safe C-space node) corresponds to a volume of freespace in the environment which is occupied by the manipulator. The only assumption is that the initial manipulator pose is safe. The two-stage exploration approach in Chapter 3 determines a sequence of viewpoints from which to sense the environment. During the exploration and mapping process, the information remaining about an environment decreases each time the environment is sensed. The information about an unknown environment can be quantified into 'information measurements' based on information theory, and then used to make exploration decisions about quality viewpoints. Sensing from a quality viewpoint results in the efficient gathering of information about the environment (i.e. efficient map building), which is equivalent to decreasing the remaining information (i.e. uncertainty) about the environment.

A widely used approach to represent the environment is the one which divides space up into a grid. Each grid cell can be assigned a probability of being in a certain state. This representation will also be used in this research for exploration and will take the form of voxels (volumetric pixels) in Euclidean space. Each voxel has a 3D position in space, a probability of occupancy, a probability of the surface material-types (if occupied), and a probability of being inside the volume occupied by a manipulator in a random pose (i.e. C-space node). C-space is similarly divided into nodes, such that each C-space node represents a manipulator pose. Each node is assigned a probability of usage during manipulator motions in a path between any two viewpoints, and the voxels which occupy the manipulator's volume when at the C-space node (i.e. the mapping between geometric voxels and each manipulator pose).

In this chapter, probabilistic methods are studied for representing an environment and to update state-space beliefs. Subsequent to sensing from a viewpoint, the probabilistic belief about the state of the geometry, C-space and material-type can be updated. When there is uncertainty about the state of a voxel, then information theory can be used to measure the information remaining. Geometric regions of interest can ensure regions with the most desirable information remaining are given higher priority during viewpoint selection. The measures of information for geometry, C-space and material-type are then formulated so that viewpoint comparisons are possible. The exploration approach can then be used so a new viewpoint can be selected from contending viewpoints.

4.1 Measurement of Remaining Information

A grid-based representation was chosen since it facilitates a robust method to discretise an environment space and update the state-space belief. This representation is conceptually similar to the Occupancy Grid (OG) which was first defined by Elfes [47], and is used extensively in robotic applications. A standard way to measure information [157] is with the mathematical notion of entropy. Entropy describes the information which remains to be discovered, i.e. the amount of "uncertainty". In accordance with Shannon's entropy [158], the general measure of information, $H({}^{j}Y)$, can be taken of a single grid cell, j. In this case, ${}^{j}Y$ is a discrete state random variable that takes the state values in a set $\{{}^{j}y_1, {}^{j}y_2, \dots {}^{j}y_n\}$. Therefore, a state ${}^{j}Y = {}^{j}y_i$, over all the possible states $i \in \{1, \dots n\}$, have probabilities of occurrence $P({}^{j}Y = {}^{j}y_i)$, $i \in \{1, \dots n\}$, such that

$$\sum_{i=1}^{n} P({}^jY = {}^jy_i) = \sum_{i=1}^{n} P({}^jy_i) = 1 \tag{4.1}$$

Using information theory [158] the information remaining in the jth grid cell, $H({}^jY)$, is defined as

$$H({}^jY) = -\sum_{i=1}^{n} P({}^jY = {}^jy_i) \log\left(P({}^jY = {}^jy_i)\right) = -\sum_{i=1}^{n} P({}^jy_i) \log\left(P({}^jy_i)\right) \tag{4.2}$$

This enables the measurement of the entropy, $H({}^jY)$, as the information remaining to be discovered about ${}^jY$. Shannon specified in [158], that the information remaining is zero, $H({}^jY) = 0$, if all the probabilities $P({}^jy_i)$, of ${}^jY$ being in state ${}^jy_i$, for $i = \{1, \dots n\}$ are zero except one with index i^*, which has unity value as in

$$P({}^jy_i) = \begin{cases} 0 & \text{for} \quad i \in \{1, \dots, n\}, i \neq i^* \\ 1 & \text{for} \quad i = i^* \end{cases} \tag{4.3}$$

Thus, only when the states are known does the information remaining, $H({}^jY)$, vanish (i.e. $H({}^jY) = 0$) since it is assumed $P({}^jy_i)\log(P({}^jy_i)) = 0$ for $P({}^jy_i) = \{1, 0\}$. To have some information remaining, at least one state must be uncertain, since $(-P({}^jy_i)\log(P({}^jy_i))) > 0$ for $0 < P({}^jy_i) < 1$. Given a number of possible states, $\{{}^jy_1, {}^jy_2, \dots {}^jy_n\}$, then the information remaining in the jth grid cell $H({}^jY)$ is a maximum when there is the most uncertainty about the state. Hence, the information remaining is a maximum when the probability of all states is equal, such as

$$P({}^jY = {}^jy_1) = P({}^jY = {}^jy_2) = \dots P({}^jY = {}^jy_n) = \frac{1}{n} \tag{4.4}$$

Then the upper bound of the information is measured as

$$H({}^jY)^{up} = -\sum_{i=1}^{n} P({}^jy_i) \log\left(P({}^jy_i)\right) = n\left(-\frac{1}{n}\log\left(\frac{1}{n}\right)\right) = \log(n) \tag{4.5}$$

Therefore, the remaining information, $H(^jY)$, for the jth grid cell (i.e. a source of information) with state variable ^jY, is within the range $[0, \log(n)]$ when there are n possible states.

In the case where there are a number of grid cells, m, then the information can be calculated for each grid cell and summed. The state variable vector $\mathbf{Y}$ represents a ^jY, for $j \in \{1, \ldots m\}$. If the number of states and the number of grid cells are assumed to be known *a priori* then it is possible to sum the information remaining in the jth cell over the m grid cells to calculate the total information which is denoted as $\mathbf{H}(\mathbf{Y})$ as follows

$$\mathbf{H}(\mathbf{Y}) = \sum_{j=1}^{m} H(^jY) = -\sum_{j=1}^{m}\sum_{i=1}^{n} P(^jy_i) log\left(P(^jy_i)\right) \tag{4.6}$$

Similar to Equation 4.5, $\mathbf{H}(\mathbf{Y})$ is maximised when all grid cell states are unknown and hence the information is bounded by $[0, m \log n]$.

In exploration of a partially known environment the explored part of the environment has two states: free or occupied. The other part of the environment is unknown, i.e. neither in a free nor an occupied state. Thus, for the manipulator-based exploration in this research, in order to distinguish between information measurements (introduced for the general case as $\mathbf{H}(\mathbf{Y})$), the symbols in Table 4.1 will be used. The geometric occupancy state variable for the jth voxel is ^jX with possible states of occupied, free or unknown. A voxel's C-space state depends upon the number of C-space nodes where the voxel overlaps with the manipulator and the frequency of these C-space nodes' usage in manipulator movement planning. A voxel's material-type states are the material-types known to the Laser Range Classifier (LRC) system [5].

Table 4.1: Environment Information Measurement Variables

Information type	**State Variable**	**Symbol**
Geometric	$^jX = \{\text{free, unknown, occupied}\}$	$\mathbf{H}_1(\mathbf{X})$
C-space	$^jC = \{$voxel overlap: i.e. few/ many frequently/ infrequently used C-space nodes$\}$	$\mathbf{H}_2(\mathbf{C})$
Material-type	$^jM = \{\text{painted metal, timber, concrete, cloth, stainless steel}\}$	$\mathbf{H}_3(\mathbf{M})$

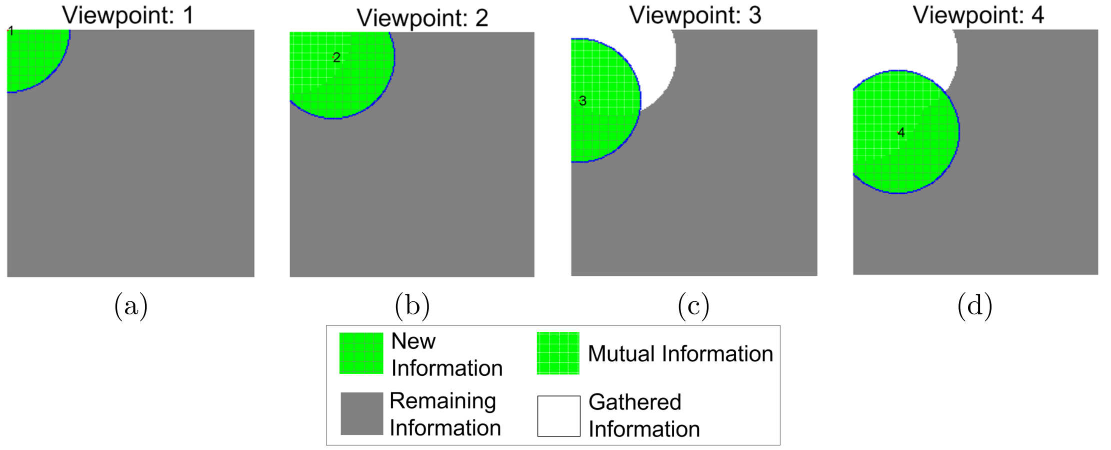
In the exploration approach, the information (of geometry, C-space and material-type) remaining measurements can be used to select viewpoints, allowing Maximum Entropy Reduction (MER) [3], which specifies that the expected change in information (i.e. information gain) is maximised. This is comparable to selecting viewpoints which reduce the information remaining.

4.1.1 Two-Dimensional Exploration Demonstration

This section presents a Two-Dimensional (2D) exploration demonstration to show the 2D space information remaining after each simulated range sensing viewpoint. During this exploration demonstration, information gathered from a sequence of viewpoints enables the update of a grid cell's probability of occupancy. Fig. 4.1 illustrates the exploration in 2D using a theoretical sensor that can perfectly detect all occupied regions within its field of view (in this case it is 360°). The 2D space is divided into 200-by-200 grid cells and the range of the sensor (radius from centre) is 50 grid cells. The legend underneath Fig. 4.1 illustrates the colour coding used. Shaded regions indicate unknown 2D space remaining (proportional to the information) where the probability of occupancy is 0.5. White regions represent freespace sensed from a previous viewpoint, while a grid pattern shows the new grid cells sensed from the current viewpoint. The sensing iteration number is shown at the position of the sensor.

The first sensing iteration (Fig. 4.1*a*) occurred from a given known safe starting position (top-left corner) and it detects the freespace (darker grid pattern). It is shown that subsequent viewpoints result in an increase in the information gathered about the 2D occupancy state of the grid.

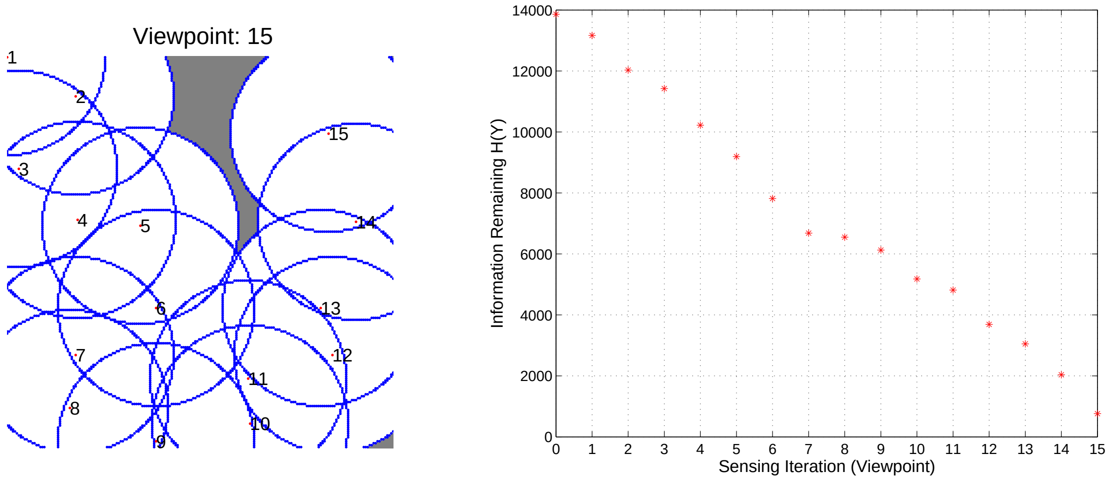
The selection of the viewpoints is arbitrary in this example since the objective is to show how the information remaining changes over the viewpoints. The only constraint is that each viewpoint's grid cell, indicated by the numbers in the sensing circles, needs to be within the space known to be empty. Fig. 4.2*a* shows that over the 15 viewpoints a significant amount of the 2D space is explored, while each viewpoint is kept within the explored area. Fig. 4.2*b* shows the information remaining after sensing from each viewpoint. It can be seen in Fig. 4.2*a* that the area sensed by the 8th viewpoint is almost completely



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

FIGURE 4.1: 2D exploration demonstration. The sensor has a 360° FOV and no obstacles exist besides the boundaries. The selection of next viewpoints is arbitrary. Sensing begins from the top left corner in *(a)*, then a viewpoint position is selected out of the data collected *(b)*, and finally sensing continues into the unexplored areas in *(c)* and *(d)*.

within known freespace, hence as shown in Fig. 4.2*b*, the information remaining is not visibly reduced. It is reasonable to assume for this case that given enough viewpoints, the information remaining will converge to 0.



(a) (b)

FIGURE 4.2: *a)* First 15 viewpoint sensing iterations from the 2D exploration example; *b)* Graphs of the information remaining after each sensing iteration. The more sensing that is performed, the less information remains in the environment.

In the case of a manipulator with a sensor mounted on its end-effector, viewpoints must still be selected such that the information remaining is minimised after each viewpoint. Application specific regions, including those regions in the work envelope and near incomplete surfaces, require a method to give that information more importance. Surfaces in the environment which occlude the sensor must be taken into account when predicting the information remaining after sensing. The challenge as detailed in Chapter 3, is to keep not just the sensor in explored freespace, but also the entire manipulator. Only in this way is it possible to guarantee safety and efficiency when the manipulator moves from one viewpoint to another during the information gathering exploration approach.

4.2 Geometric Information Model

Characterising and measuring the 3D geometric information of an environment is of particular importance for an exploration approach. During exploration, information gathered from a sequence of viewpoints enables the update of a voxel's probability of occupancy. The Hokuyo laser range finder sensor, mounted on the manipulator's end-effector, is capable of detecting objects in the line-of-sight up to a range of one metre with high accuracy ($\pm 10mm$), and up to four metres with lower accuracy ($> 1\%$) (see Appendix C for hardware details).

At each viewpoint the laser range scanner returns range values, r_i, for each ray such that a point $\mathbf{p}_i$, a 3D position vector, $\mathbf{p}_i = [\mathsf{x}_i, \mathsf{y}_i, \mathsf{z}_i]_r^T$, where the position can be calculated using the pose of the manipulator by:

$$\mathbf{p}_i = [\mathsf{x}_i, \mathsf{y}_i, \mathsf{z}_i, 1]_r^T = {}^0T_f(\vec{Q})^f T_s\,[0, -r_i \sin\theta_i, r_i \cos\theta_i, 1]^T \tag{4.7}$$

where ${}^0T_f(\vec{Q})$ describes the homogeneous transformation matrix of the end-effector when the manipulator is in a pose $\vec{Q}$, and ${}^fT_s$ is the transformation matrix between the end-effector and the sensor as detailed in Chapter 3. These two transformation matrices describe the viewpoint. The ith ray has an angle of θ_i with respect to the axis of the laser sensor, and r_i is the distance along the scan ray to the nearest object. The adjacency of the scan data points is maintained so a mesh could be formed by joining the adjacent points. The manipulator provides time-stamped configuration data at 10Hz. Range data is continuously acquired and time-stamped while the manipulator moves. To support fusion of multiple scans into a single surface location estimate (and thus a partial surface geometry map), an adaptive distance field map representation (volumetric technique) was implemented based upon research by Curless [42] and Webb [43].

In the special case where a sensor that can detect all occupied regions within its field of view is available, the voxel's probability of occupancy will take one of the three states: free, unknown and occupied (i.e. containing a surface). The geometric space is discretised to contain n_v equally sized volumetric pixels (voxels). Where ${}^{j}X$ is a discrete state variable, the probability that the jth voxel is occupied is contained in a tristate buffer given by $P({}^{j}X = {}^{j}x_o)$ for the voxels, $j \in \{1, \ldots, n_v\}$, and states $o \in \{\text{free}, \text{unknown}, \text{occupied}\}$. Therefore, the probability of occupancy is $P({}^{j}x_o) = \{0, 0.5, 1\}$, if the jth voxel's occupancy state, ${}^{j}x_o$, is {freespace, unknown, occupied}, respectively. Initially, all voxels except those in the volume occupied by the robot manipulator, are unknown, ($P({}^{j}x_o) = 0.5$). It can be recognised that the occupancy states of freespace and occupied are complementary, i.e. the probability of occupancy of the jth voxel is $P({}^{j}x_o)$ and the probability of non-occupancy (i.e. freespace) is $1 - P({}^{j}x_o)$. In the case where the voxel is occupied, then $P({}^{j}x_o) = 1$, if unknown then $P({}^{j}x_o) = 0.5$, or if the voxel is freespace then $P({}^{j}x_o) = 0$. As the environment is discretised into n_v voxels with a probability of occupancy $P({}^{j}x_o)$, the geometric entropy, $\mathbf{H}_1(\mathbf{X})$, of the environment's geometric state, $\mathbf{X}$, can be used as a measure of cumulative information remaining in all voxels.

Since there is no information unless there is uncertainty (according to Equation 4.6 and Equation 4.3), the information remaining measure, $\mathbf{H}_1(\mathbf{X})$, is directly related to the number of unknown voxels, n_u. This results in the count of voxels that have not yet been explored. In this case, since the 'unknown' state occurs where there is an equal probability of the state being free or occupied, the state space is effectively reduced to two states. Since there are only two occupancy states, binomial entropy can be used to calculate the information in the jth voxel

$$H_1({}^{j}X) = -P({}^{j}x_o)\log\left(P({}^{j}x_o)\right) - \left(1 - P({}^{j}x_o)\right)\log\left(1 - P({}^{j}x_o)\right) \tag{4.8}$$

As the environment is discretised into n_v voxels with a probability of occupancy $P({}^j x_o)$, the geometric information remaining, $\mathbf{H}_1(\mathbf{X})$, in all voxels with occupancy state vector, $\mathbf{X}$, can be used as a measure of cumulative information remaining in all n_v voxels

$$\mathbf{H}_1(\mathbf{X}) = \sum_{j=1}^{n_v} H_1({}^j X) \tag{4.9}$$

Based upon the voxel's probability of occupancy and the fact that there is no information in voxels with a known occupancy state (free or occupied), the information remaining is therefore proportional to the number of voxels with unknown occupancy state, n_u, as

$$\mathbf{H}_1(\mathbf{X}) = \sum_{j=1}^{n_u} -0.5\log(0.5) - (1-0.5)\log(1-0.5) = n_u \log(2) \tag{4.10}$$

In order to *minimise* the remaining information, the number of unknown voxels whose states are determined at each viewpoint must be *maximised*. Therefore, it is necessary to be able to predict the number of unknown voxels which can be sensed from a viewpoint. These predictions must account for occlusions faced by the sensor. SLDs can be used to represent the obstructing object's surfaces in an environment in a more efficient manner than a fine mesh. This is because the number of ray intersection calculations can be reduced if the surfaces are larger. The details of the technique to create SLDs will be presented in Chapter 5.

The unknown voxel's volume of space which can be sensed is denoted as, $V_{new}(\vec{Q})$. This volume is a function of sensor's position and orientation and is proportional to n_u. The prediction of $V_{new}(\vec{Q})$ can be determined by the established technique of ray casting [113]. Fig. 4.3 shows that for each of the laser scanner's rays, a ray is cast from the proposed sensor viewpoint, ${}^0T_s(\vec{Q})$, through the partially known map to the extent of the sensing range. If the ray doesn't intersect a known SLD, all unknown voxels which the ray passes through (up to the maximum sensing range) are included in the predicted volume $V_{new}(\vec{Q})$. If the ray intersects with one or more SLDs then, only the unknown voxels between the sensor and the closest SLD are included in $V_{new}(\vec{Q})$. Thus, the reduction in the number of unknown voxels can be predicted for each viewpoint, $\vec{Q}$.

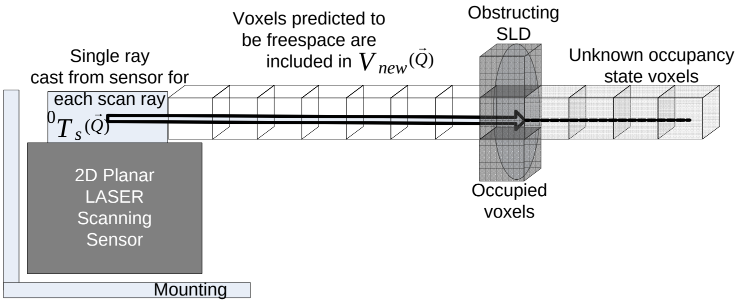


FIGURE 4.3: Ray casting occurs for each ray from the sensor. If a ray passes through a voxel it is determined to be free, voxels which contain a SLD are occupied and voxels which have not had a ray cast through them at any stage have unknown occupancy state.

4.2.1 Regions of Interest

Regions in an environment are not always of equal importance. Therefore, the application-specific requirements of exploration can be considered in order to devise weightings for specific regions. In this research, the concept of 'regions of interest' is applied for the geometric exploration and mapping [17][21]. The regions of interest are devised based upon the requirements for the information gathering (mapping) process. Fig. 4.4 shows the extent of three regions: A_1, A_2 and A_3. Surfaces in these regions could be affected by, or acted upon by the maintenance tool with varying levels of success.

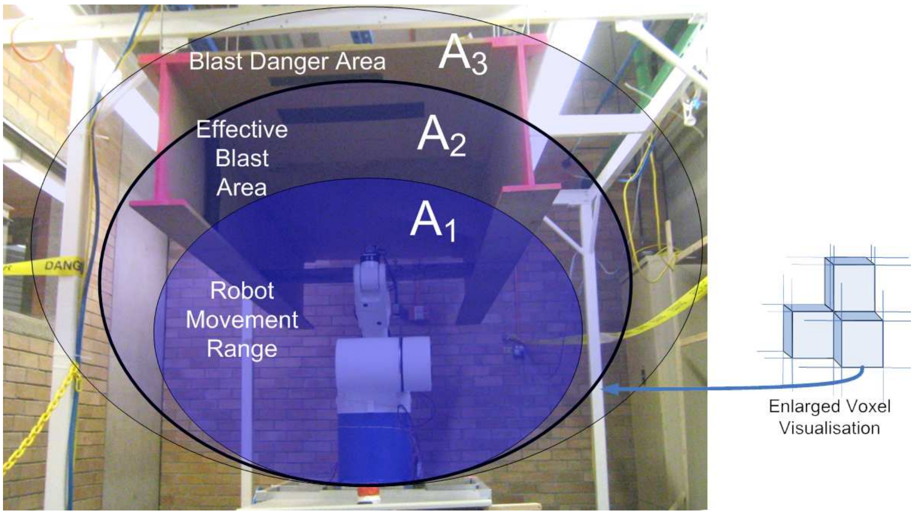


FIGURE 4.4: Application specific requirements are included. In this case the approximate workspace regions of interest $\{A_1, A_2, A_3\}$ can assist the exploration approach to select viewpoints. An enlarged voxel (volumetric pixel) visualisation is also shown to the right.

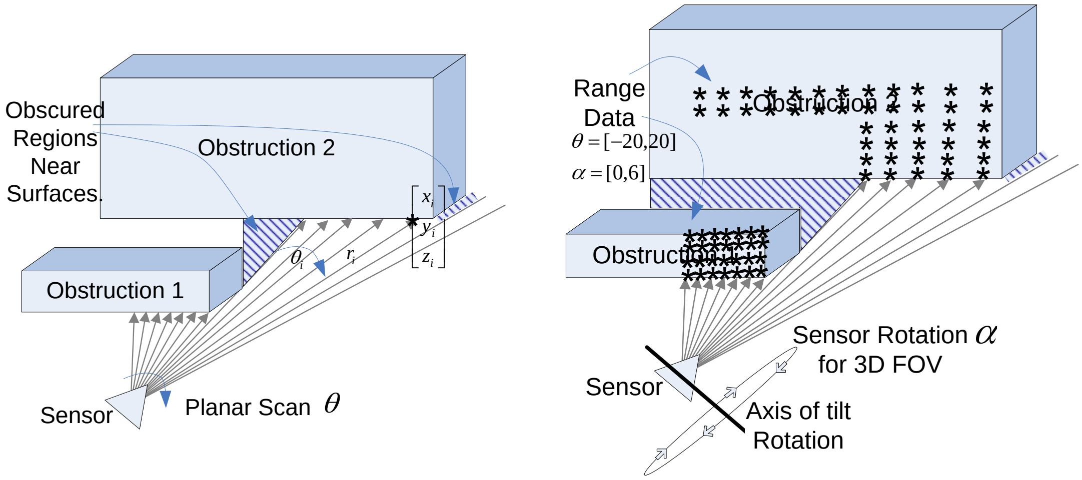
Mapping freespace and surfaces in A_1 is vital due to the manipulator's movement traversing this Euclidean space. The manipulator can collide with obstacles in A_1 including the walls, roof and floor of an environment. A_2 is outside the manipulator's workspace, however a tool on the end-effector can reach this region (i.e. A_2 is still in the robot's working-envelope). Surfaces in both A_1 and A_2 can be acted upon (i.e. maintained), therefore surfaces and freespace in these regions must have information gathered about them so that it is possible to perform maintenance operations. A_3 is the region containing surfaces which could also be damaged during maintenance operations [17], so this region too must be mapped, although with less concern. Note also that voxels in region A_1 are also in regions A_2 and A_3, so $A_1 \subset A_2 \subset A_3$.

In grit-blasting operation, the blasting stream (garnet and compressed air) length is allowed to change within a range of 200-500mm. Therefore, the distance between the end-effector of the manipulator and the surface to be treated is variable [18]. An approximation of the maximum work-envelope (i.e. the workspace of the manipulator with a tool attached) is adequate. The regions of interest are centred at the position of the first joint of the manipulator. They cover the maximum reach of the particular robot manipulator arm, plus the maximum maintenance tool range. Each region of interest, $\{A_1, A_2, A_3\}$, is discretised into a number of equally sized voxels, $\{n_{A1}, n_{A2}, n_{A3}\}$.

Voxels in each region are given weighting constants, $\{c_{w1}, c_{w2}, c_{w3}\}$, to indicate their relative importance during the exploration process. The sum of the weighting values is unity, $c_{w1} + c_{w2} + c_{w3} = 1$ and $0 < c_{w3} < c_{w2} < c_{w1} < 1$. Since the solution is closely related to the weighting values, their choice is carefully determined heuristically. The determination process was based upon an understanding of the map that is required and research on the (mobile robot) exploration of unknown spaces by maximising the understanding of different regions [164]. Where the number of voxels with unknown occupancy in each region is $\{n_{u,A1}, n_{u,A2}, n_{u,A3}\}$, respectively, the voxels' relative importance weights are then used to weight the geometric information measurement as

$$\mathbf{H}_1(\mathbf{X}) = (c_{w1} n_{u,A1} + c_{w2} n_{u,A2} + c_{w3} n_{u,A3})\, log(2) \tag{4.11}$$

A complete map of surfaces in the environment is required. It is possible to improve the map-completeness by increasing the ‘importance’ of voxels that are obscured during sensing, as shown in Fig. 4.5*a* and Fig. 4.5*b*. These are regions with a large differential in the range data values.



(a) (b)

FIGURE 4.5: Simulated occlusions. *a)* Occluded regions of high uncertainty occur where there are large ray length differentials; *b)* Tilting the laser sensor through α so as to get a 3D FOV. Fig. 4.6 graphs the data from this simulation.

The range data collected in Fig. 4.5*b* can be presented as a matrix of ray lengths, $R_{\theta,\alpha}$, corresponding to the tilting angle of the sensor, α, and the angle on the scan plane, θ,

$$\mathbf{R}_{\theta,\alpha} = \begin{pmatrix} r_{i,j} & r_{i+1,j} & \cdots \\ r_{i,j+1} & r_{i+1,j+1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{4.12}$$

where the index i refers to the pan angle on the scan plane, and j refers to the tilt angle which tilt the planar sensor through α to achieve a 3D FOV. This range data is shown in Fig. 4.6 with the actual angle pairs, $\{\theta, \alpha\}$, replacing the matrix index, $\{i, j\}$.

If the absolute value of the difference between adjacent range values (i.e. $|r_{i,j} - r_{i+1,j}|$ or $|r_{i,j} - r_{i,j+1}|$) is large, then there is likely to be a region that is obscured, which if viewed from a different viewpoint, could reveal the state of currently unknown voxels. The point

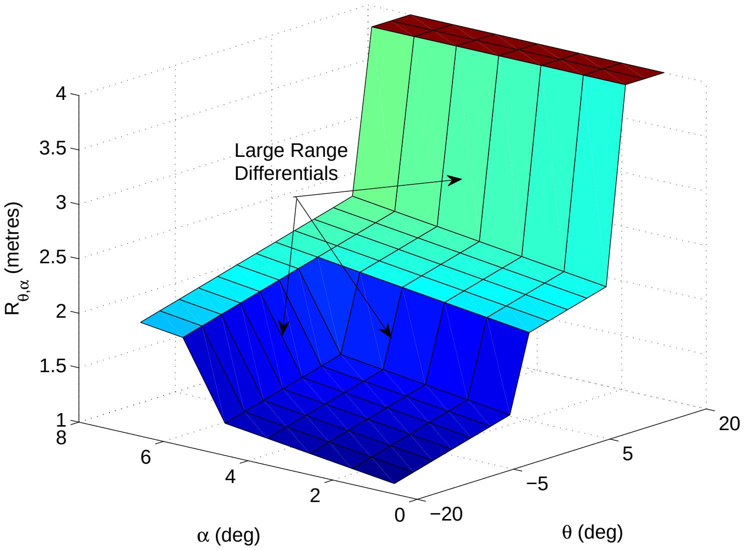


FIGURE 4.6: Range measurements, $\mathbf{R}_{\theta,\alpha}$, from Fig. 4.5*b* as θ_i changes on the scan plane and the sensor is tilted through α about the axis of tilt rotation. Large differentials in the range measurements exist which correspond to edges in the geometry of Fig. 4.5.

on the object surface where $r_{i,j}$ is returned from is denoted as $\mathbf{p}_{i,j}$, and the adjacent point (with a large differential) corresponding to $r_{i+1,j}$ and $r_{i,j+1}$ is denoted $\mathbf{p}_{i+1,j}$ and $\mathbf{p}_{i,j+1}$, respectively. The voxels in between which are occluded can be determined by ray casting between $\mathbf{p}_{i,j}$ and $\mathbf{p}_{i+1,j}$ (or $\mathbf{p}_{i,j+1}$). The voxels which are in between the range values with high differentials (which the ray passes through), are considered with more importance weighting than other voxels in the same region. Thus, the information in different regions of interest is weighted. This leads to viewpoints which can sense and gather information about these regions becoming more desirable.

4.3 Configuration Space Information Model

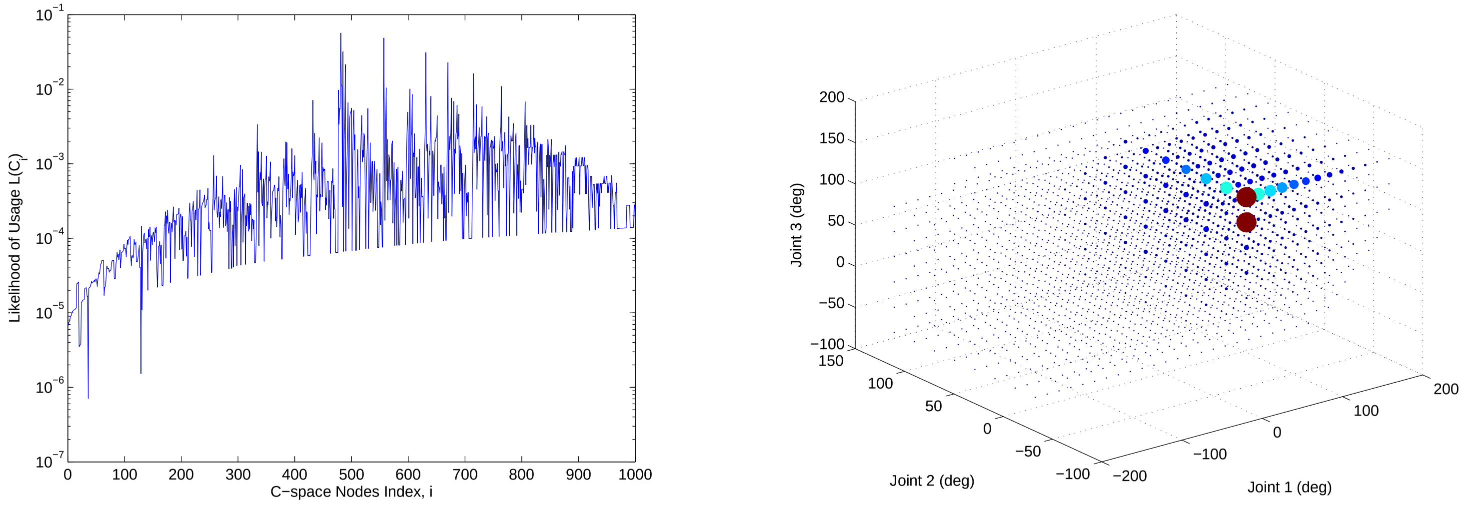
A manipulator with pose $\vec{Q}$, occupies the volume in 3D space $V_r(\vec{Q})$. This volume $V_r(\vec{Q})$, must not overlap with occupied voxels since this is equivalent to a manipulator-environment collision. Additionally, $V_r(\vec{Q})$ must also not overlap with unknown, unexplored voxels due to possible collisions with unknown obstacles. Manipulator poses (i.e. C-space nodes), where there is overlap between $V_r(\vec{Q})$ and voxels with an occupied or unknown state, are not available for use when moving the manipulator from one pose to another (e.g. between two viewpoints). Henceforth, a voxel inside $V_r(\vec{Q})$ will be described as 'interfering' with the C-space node $\vec{Q}$. Unknown, unexplored voxels in an environment which can be reached by a manipulator, correspond to 'unknown nodes' in C-space. In terms of manipulator motion planning, reducing the amount of information remaining (unknown nodes) about the C-space results in a greater freedom of movement. This was formulated in Stage One of the exploration approach (Chapter 3) by the second objective, $f_2(\vec{Q})$. The ability to explore based upon C-space information, $\mathbf{H}_2(\mathbf{C})$, enables an additional dimension which complements, but cannot replace, geometric information, $\mathbf{H}_1(\mathbf{X})$, exploration. Therefore, it is necessary to discover the state of voxels which are inside the manipulator's workspace, and interfere with frequently used C-space nodes, so as to minimise the information remaining in C-space, $\mathbf{H}_2(\mathbf{C})$.

The C-space information remaining in an unknown geometric voxel is based upon the relationship between the voxel and the C-space nodes, and the relative importance of each C-space node to manipulator motion planning. Geometric voxels which are known to be free or occupied contain neither geometric information nor C-space information. Therefore, it is necessary to determine for each voxel, the number of C-space nodes where the volume occupied by the manipulator, $V_r(\vec{Q})$, overlaps with that voxel (i.e. the potential of a voxel, if unknown, to result in unknown, unusable nodes in C-space). Additionally, the importance of a C-space node can be determined based upon the likelihood that it will be used during planned manipulator movements. For the jth geometric voxel, a probability of C-space node interference, $P({}^j x_c)$, can then be calculated based upon the likelihood that the voxel is in $V_r(\vec{Q})$ for any given C-space node $\vec{Q}$, and including each node's expected usage. Then the C-space information contained and remaining in a voxel can be measured used information theory.

To determine the relationship between voxels and C-space nodes, and the importance of each C-space node, an exhaustive search was conducted offline. A fixed-base manipulator in a pose will always overlap the same set of voxels in space (i.e. occupy the same space), thus this search can be conducted offline without considering the voxels' state. In a simulated environment containing no obstacles, manipulator motion paths can be determined via the planning module between each C-space node out of a set of C-space nodes ($n_Q = 1000$ was used in this case) and all other nodes in the set. Where n_Q is the number of C-space nodes, the likelihood that the ith C-space node, c_i for $i \in \{1, \ldots n_Q\}$, will be traversed when moving between any two poses is $L(c_i)$. Initially, a manipulator trajectory is planned from every C-space node (i.e. pose) to every other C-space node to determine the bottleneck nodes as we demonstrated in [165]. Statistics about the frequency of usage are collected, then the relative likelihood, $L(c_i)$, is graphed over the C-space nodes for the manipulator (Fig. 4.7). Fig. 4.7 shows that some C-space nodes are used more frequently than others and hence are of greater importance. In Fig. 4.7a the horizontal label is the C-space index number and is not recognisable as a pose or physical space since it is an unordered index of the 6DOF manipulator's C-space nodes. Fig. 4.7b shows this usage translated into the first 3 joints of C-space, with the colour and size of the small circles representing the usage of the set of C-space nodes. The usage of each C-space node as part of manipulator trajectories is then evident.

For each C-space node $\vec{Q}$, the voxels inside $V_r(\vec{Q})$ can be determined so a relationship is formed between each voxel and nodes in C-space. The volume occupied by the manipulator is determined by a 3D ellipsoidal virtual bounding field method based on [126][128], as used to ensure manipulator safety in the exploration approach (Chapter 3). In this way it is possible to determine the voxels which have an algebraic distance of less than 1 for at least one ellipsoid covering the links of the manipulator when in a pose $\vec{Q}$. These voxels are considered to potentially interfere with the C-space node, $\vec{Q}$.

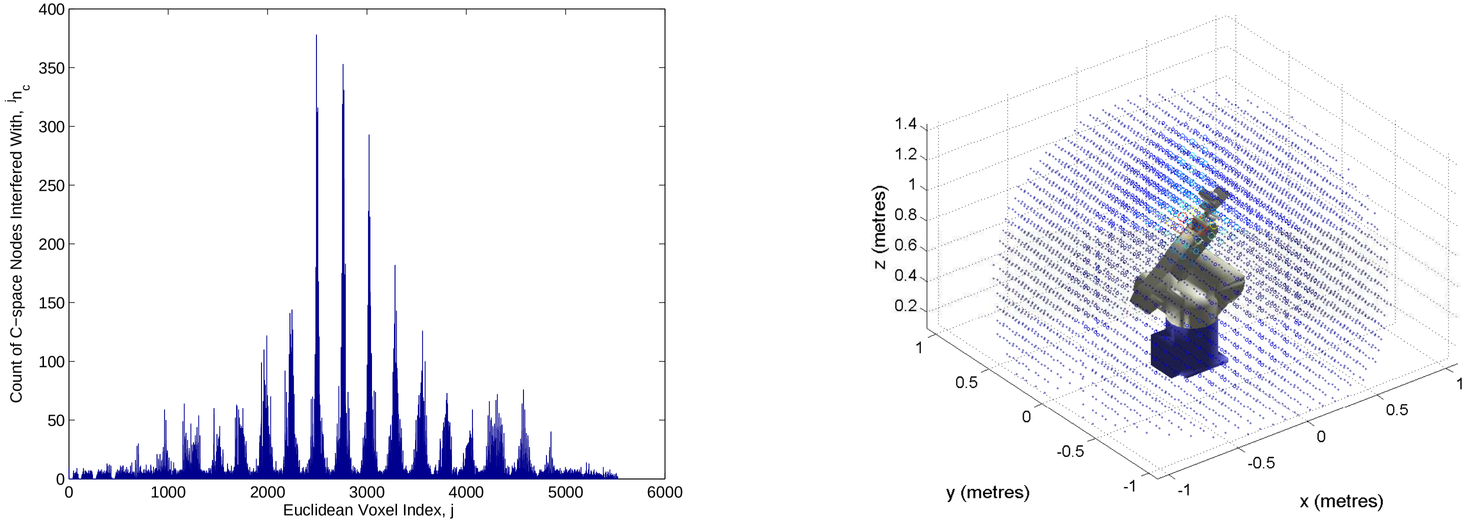
For the discretised C-space, each C-space node corresponds to a manipulator pose occupying the volume $V_r(\vec{Q})$, in Euclidean space. A histogram of the number of poses, ${}^{j}n_Q$, that the jth voxel interferes with, is shown in Fig. 4.8a. This corresponds to an importance which is shown in Euclidean space (Fig. 4.8b), with size and colour representing greater importance of a geometric region in terms of C-space. Generally, regions of space close to the base of the manipulator were found to interfere with a greater number of C-space nodes.



(a)

(b)

FIGURE 4.7: Usage of C-space nodes. *a)* The likelihood, $L(c_i)$, that the jth node is used when moving between any two poses, calculated by a count over all nodes used. Some nodes in C-space are traversed with greater likelihood than others (up to five orders of magnitude on the logarithmic vertical likelihood axis in some cases). The horizontal axis is the index of nodes;
b) Likelihood of usage of C-space nodes for the first 3DOF can be visualised in a 3D plot with the joints as axes. The likelihood of usage is indicated by the size and colour of the spheres.



(a)

(b)

FIGURE 4.8: *a*) Histogram of the poses interfered with by voxel index. *b*) Importance of the voxels in Euclidean space with colour and size based on the count of poses which they interfere with. The manipulator is also shown for scale.

The jth voxel has a C-space interference state, ${}^{j}x_c$. The probability that the jth voxel, if unknown, interferes with a C-space node is denoted as $P({}^{j}x_c)$. In order to calculate this probability, the set of C-space nodes from Fig. 4.8a which are interfered with by the jth voxel is denoted as ${}^{j}\mathbf{c}$. Therefore, voxels in the manipulator's workspace which can be reached have $0 < P({}^{j}x_c) \leq 1$, while all other voxels, including those outside of the workspace have $P({}^{j}x_c) = 0$. In Fig. 4.7a it was shown that the likelihood of use of each C-space node, $L(c_i)$, for $i \in \{1, \ldots n_Q\}$ is not equal. Therefore, the probability that the jth voxel interferes with a C-space node must be weighted based upon the likelihood of utilisation, $L(c_i)$. For each of the ${}^{j}n_Q$, C-space nodes interfered with by the jth voxel ${}^{j}\mathbf{c}$, instead of assuming equal importance (i.e. weighing = 1), the weighing is given as the likelihood of usage. The likelihood that the jth voxel interferes with C-space is the sum of the interfered-with C-space node's likelihoods of usage, $L({}^{j}\mathbf{c}) = L({}^{j}c_1) + \ldots L({}^{j}c_{n_Q})$. The probability is then the normalised likelihood

$$P({}^{j}x_c) = \frac{L({}^{j}\mathbf{c})}{\sum_{i=1}^{n_Q} L(c_i)} \tag{4.13}$$

such that $0 \leq P({}^{j}x_c) \leq 1$. This probability is theoretically minimised when the voxel does not interfere with any C-space nodes, and is theoretically maximised when a voxel interferes with every C-space node (these maximum/ minimum cases are unlikely in practice). In general, a voxel in the manipulator's workspace will interfere with several C-space nodes. Hence, if a voxel in the manipulator's workspace is unknown, then there are unknown C-space nodes and it follows that there should also be information remaining about C-space.

To enable the pursuit of C-space information, a measure of the geometric data gathered from a viewpoint in terms of the C-space is required. This measurement of C-space information, $\mathbf{H}_2(\mathbf{C})$, can then be used to rank a set of viewpoints based upon the predicted reduction of the remaining C-space information as formulated in $f_2(\vec{Q})$ from Stage One of exploration. The probability of a voxel, if unknown, interfering with C-space nodes, $P({}^{j}x_c)$, can be used to measure the C-space information remaining in a voxel. Then, given the C-space information in a single voxel and the predicted voxels which can be sensed from a viewpoint, the C-space information remaining in an environment, $\mathbf{H}_2(\mathbf{C})$, can be determined.

A voxel with an unknown geometric occupancy state which interferes with very few C-space nodes, and where all nodes are infrequently used, will have $P({}^{j}x_c) \approx 0$. Conversely, a voxel which is unknown and interferes with many frequently used C-space nodes will have $P({}^{j}x_c)$ closer to unity. In order to reduce the number of unknown nodes in C-space, voxels with a probability of C-space interference closer to unity must have their occupancy state determined. The probability is proportional to the 'desirability' of a voxel. In order to incorporate this into an information measurement, $P({}^{j}x_c) = 1$ should maximise the information remaining. The probability of non-interference is the complement $1 - P({}^{j}x_c)$. As previously stated, where there are two outcomes (i.e. interference or non-interference) the binomial information function is maximised when the probability is 0.5, since this maximises Equation 4.10. Therefore, a one-sided uncertainty, $0 < \frac{P({}^{j}x_c)}{2} < 0.5$, is created. The probability for each voxel thus ranges from 0 to 0.5. This can be used to generate an information-remaining measurement for a single voxel about C-space. The maximum information remaining exists when the probability of C-space interference by an unknown voxel is maximised, $\frac{P({}^{j}x_c)}{2} = 0.5$. Then similar to the geometric information (Equation 4.9), the C-space information remaining in each geometric voxel is summed over all voxels as follows

$$\mathbf{H}_2(\mathbf{C}) = \sum_{j=1}^{n_v} -\frac{P({}^{j}x_c)}{2}\log\left(\frac{P({}^{j}x_c)}{2}\right) - \frac{1-P({}^{j}x_c)}{2}\log\left(\frac{1-P({}^{j}x_c)}{2}\right). \tag{4.14}$$

4.4 Material-Type Information Model

The last information measure required is that of the material-type information remaining on all surfaces in an environment, $\mathbf{H}_3(\mathbf{M})$. Knowledge of the surface geometry is a prerequisite to surface material-type identification and is obtained from Stage One of exploration. In Stage Two, the material-type information on the surfaces must be measured in order to select target SLDs, which require material-type identification.

The jth occupied voxel holds the belief, $P({}^{j}x_m)$, of the material-type state, ${}^{j}x_m$. The probabilities of the material-type states are only kept where there are surfaces - it is not necessary to hold a material-type belief about a surface that doesn't exist. Hence, the number of material-type voxels, n_m, is never greater than the total number of voxels in geometric space, n_v (i.e. $n_m \leq n_v$). Once the material-type information remaining in a voxel is measured, then the information available on a SLD (or for the environment) can be obtained by summing the information in the corresponding voxels. Given the material-type information on each SLD, the SLDs can be compared, and a target SLD for material-type identification can be selected. Then a viewpoint (i.e. a manipulator pose) can be determined as shown in Stage Two of the exploration approach. Predictions can also be made about the material-type information remaining after the SLD target is sensed from the viewpoint determined by Stage Two.

4.4.1 Surface Material-type Belief Update

Details of the LRC system are presented in Appendix B, however as introduced in Chapter 3, the roughness and reflectance properties of the surface can be used as the basis for surface material-type identification. A Bayesian approach [166] was used as the LRC's classification technique such that the LRC system produces surface material-type identifications using the intensity and range curves. Each voxel stores the probability $P({}^{j}M = {}^{j}x_m) = P({}^{j}x_m)$, that the jth voxel's material-type state variable ${}^{j}M$ equals ${}^{j}x_m$ for $m \in \{1, \ldots n_t\}$, given the n_t possible materials. $P({}^{j}x_m)$ is initialised as a uniform distribution across each material-type state since there is no surface material-type *a priori* knowledge. For each sensing observation (i.e. a single laser scanner ray) of the jth voxel, the LRC system returns the probability distribution for a material-type state, $P({}^{j}z_m)$. This probabilities distribution is used as evidence to update the material-type belief probability for a voxel.

Bayes' theorem is applied after sensing at each viewpoint. The environment is assumed to be stationary, such that the material-type state in a voxel does not change with time. Since sensing with the LRC contains errors and is often an ambiguous process, the viewpoint observation is modelled as a stochastic function, $P({}^{j}x_m|{}^{j}z_m)$, of the material-type state, ${}^{j}x_m$.

Bayesian probability theory can be used to estimate the most likely material-type, $P({}^{j}x_m|{}^{j}z_m)$, for each voxel given the sensing data, $P({}^{j}z_m)$, the LRC system sensing model, $P(z_m|x_m)$, the LRC sensing model, $P(z_m|x_m)$, and the current voxel belief, $P({}^jx_m)$. The posterior probability is based on conditional and so-called 'prior' probabilities derived from the LRC model about the material-type being measured, $P(z_m|x_m)$. Note how the sensor model is not dependant upon the voxel. The probability distribution returned by the LRC can thus be used as evidence to update the belief of the surface-type in a voxel by applying Bayes' theorem after sensing at each viewpoint, Given these distributions, it is possible to update the probability of the material-type state of a voxel, based on the current belief for the voxel, $P({}^jx_m)$.

$$P({}^jx_m|{}^jz_m) = \frac{P(z_m|x_m)P({}^jx_m)}{P({}^jz_m)} \tag{4.15}$$

Given the distribution over ${}^jx_m$, a single material-type state estimate ${}^j\hat{x}_m$, can be extracted (a decision about the material-type) by using the discrete maximum likelihood estimate methodology [167].

As with the uncertainty in geometric and C-space, the uncertainty of the material-type state, $P(jx_m)$, for the n_t surface-type states, can be used to measure material-type information remaining (similar to Equation 4.6) in the jth voxel,

$$H_3({}^jM) = \sum_{i=1}^{n_t} P({}^jx_m)\log\big(P({}^jx_m)\big). \tag{4.16}$$

The information remaining in an environment, which is represented by n_v voxels can then be calculated by summing the information in each voxel,

$$\mathbf{H}_3(\mathbf{M}) = -\sum_{j=1}^{n_m} H_3({}^jM). \tag{4.17}$$

Equally, the information remaining in a single SLD can be determined by summing the information remaining in the voxels which the SLD represents.

4.4.2 Surface Material-type Identification Demonstration

A material-type identification demonstration with two sets of results is presented here. The demonstration shows the material-type belief updates and the predictions versus the actual information remaining after inspecting (i.e. sensing) a surface.

A number of voxels containing surfaces (and hence a surface material-type belief) on the walls, sides and roof were monitored. The first set of results (Fig. 4.9) show each surface contains material-type information which is presented as different sized circles with a CAD overlay for this particular simple environment. The base of the manipulator is at the origin. The Stage Two exploration approach determined poses so that the sensor was appropriately directed at SLD targets with high information content. At every viewpoint, the information remaining in each region (a set of voxels) is represented by sphere sizes in Fig. 4.9. Larger spheres indicate that there is more information remaining. Bayesian theory occurs subsequent to each sensing viewpoint to update the belief about a voxel's material-type state, such that the information remaining is reduced and the surface material-type becomes more certain.

Fig. 4.9*a* shows the state after the second iteration of Stage Two for material-type identification. Some material-type identification was possible during the geometric exploration at Stage One, and the first two viewpoints of Stage Two. However, there is still a large amount of material-type information remaining. Fig. 4.9*b* shows the information remaining prior to the 13th viewpoint. The information has been reduced significantly compared with Fig. 4.9*a*. According to Equation 4.17 and graphed in Fig. 4.10 the material-type information remaining has approximately halved. Fig. 4.9*c* shows the remaining information is once again reduced. There are several surfaces at the extremities of the environment which cannot be sensed adequately so that the material-type can be identified with the LRC system.

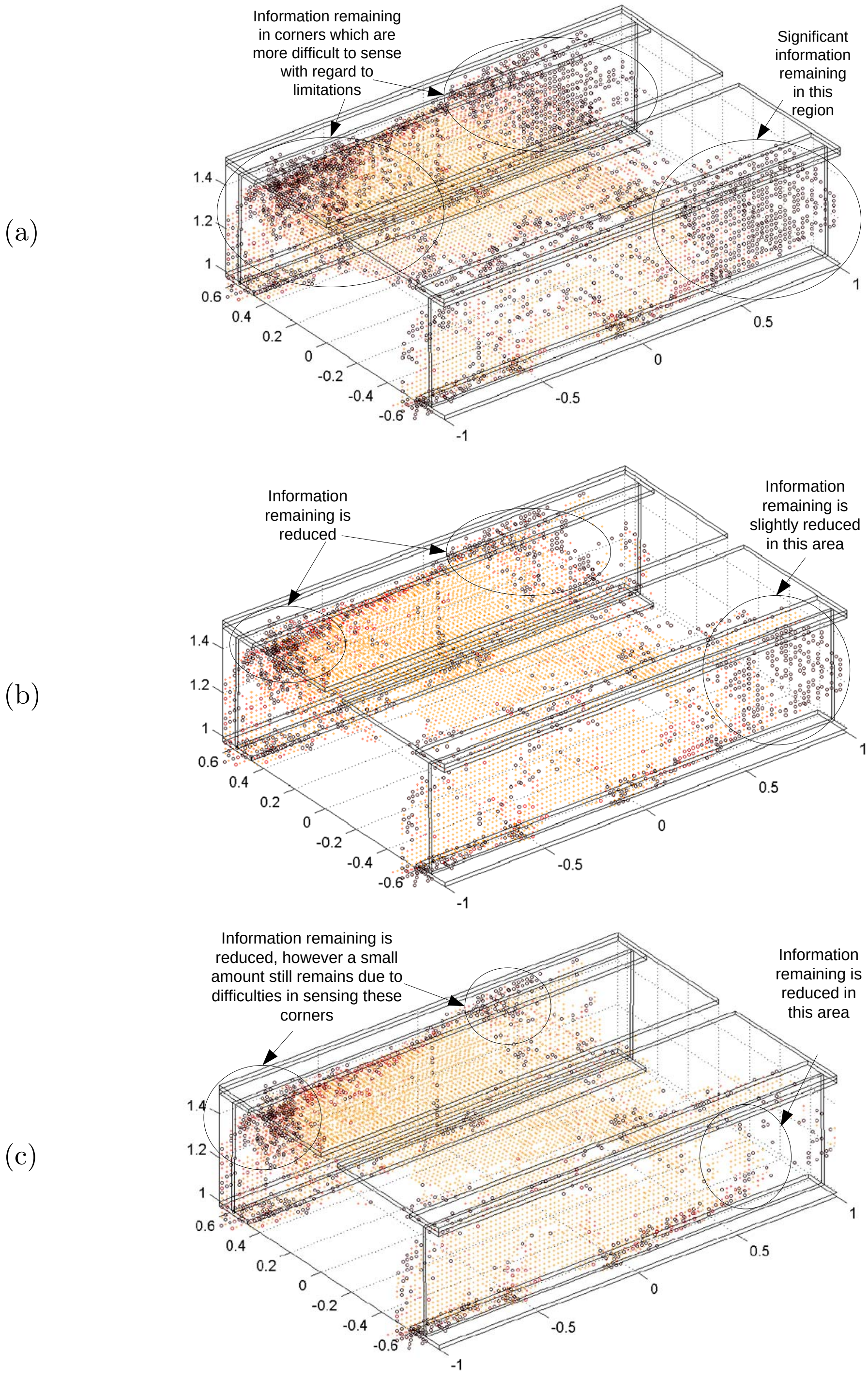


FIGURE 4.9: As the belief about the state of each area changes with each viewpoint, the surface material-type is identified and the information remaining is reduced.
a) At the 2nd iteration, there is a great deal of information;
b) By the 13th iteration the information has been slightly reduced;
c) After the 26th iteration, in the close range inspection areas there is very little material-type information remaining which is possible to sense.

Fig. 4.10 presents the plot of the information remaining from the demonstration compared with the information that was predicted (after viewpoint $(k-1)$th) would remain. It is assumed in Stage Two of exploration that sensing the targeted SLD within the LRC's constraints will enable successful material-type identification to reduce the information remaining on the SLD. Thus, the predicted information remaining $\mathbf{H}_3(\mathbf{M}_p)_{k-1}$, in the predicted state variable vector $\mathbf{M}_p$, is calculated based upon the reduction of information on that SLD, and the assumption that the other SLDs will not be identified. Due to the LRC system sensing constraints, it is generally only possible to identify one SLD target at a time. The predicted information remaining is compared with the measurement of information remaining, $\mathbf{H}_3(\mathbf{M})_k$, after sensing at the kth viewpoint. Fig. 4.10 shows that the prediction is similar to the actual resulting information remaining across all viewpoints in this test case, $\mathbf{H}_3(\mathbf{M})_k \approx \mathbf{H}_3(\mathbf{M}_p)_{k-1}$.

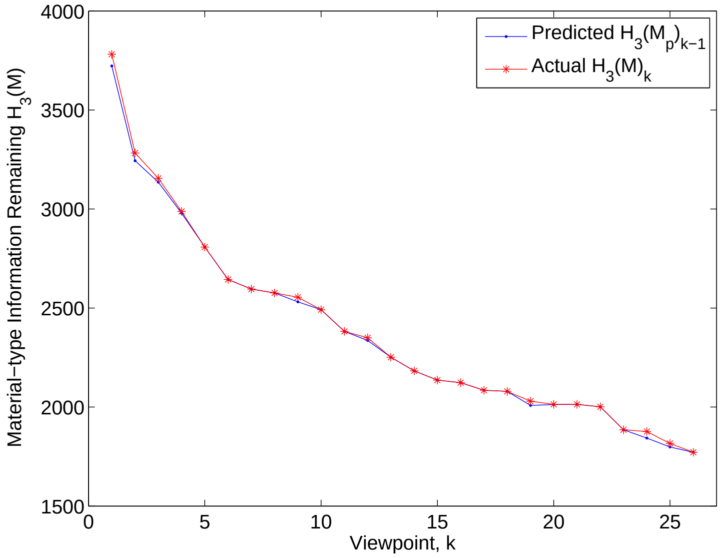


FIGURE 4.10: Remaining material-type information marked with $(\cdot)$ and predicted remaining information marked with $(*)$ while exploring to determine the material-type.

This demonstration verifies how it is possible to reduce the information remaining about a surface's material-type identification by selecting viewpoints which overcome the LRC's constraints. The belief about each voxel's surface material-type state is updated, which results in a reduction of the information remaining. Additionally, the demonstration shows how predictions about the information remaining subsequent to sensing a targeted SLD from a viewpoint are similar to the actual information remaining after sensing.

4.5 Conclusions

The measurement of the information remaining in an environment has been presented in this chapter. An information theory-based approach was taken so that possible exploration viewpoints can be compared based upon the predicted reduction in the information remaining they achieve. The probabilistic belief of the state of the environment is linked with the information measurements remaining. Geometric regions of interest were defined to weight information in different regions, then the relationship between the Euclidean space and C-space was also investigated. The SLD representation was used for geometric information reduction predictions and as targets for material-type identification. At each viewpoint in exploration, surface geometry data is collected and must be carefully represented (i.e. SLDs and segmented maps). The following chapter presents techniques to represent the data collected in exploration efficiently, so as to assist in exploration and in maintenance operation planning.

Chapter 5

Surface Representation and Map Segmentation

Exploration is done for the purpose of developing a map (i.e. surface representation) which can be utilised in applications such as manipulator trajectory planning and coverage planning. The exploration approach discussed in Chapter 3 and in Chapter 4 provides an efficient way for the exploration of unknown complex 3D environments to occur. The data collected in exploration includes geometric information of an environment and material-type information of the surfaces in this environment. The challenge after exploration is the presentation of the collected geometric and material-type data. The application-specific requirement is to build maps which are directly usable for manipulator trajectory planning. An appropriate surface representation is a reduced data set consisting of partially overlapping Scale-Like Discs (SLDs) which are carefully grouped to form map segments. This chapter presents a technique for generating a representation of surfaces based upon two algorithms. The first algorithm, SLD generation, transforms the fused mesh maps into a more compact usable format via Principal Component Analysis (PCA). The second algorithm groups similar SLDs using cluster analysis to aid manipulator trajectory planning such as those used in surface maintenance operations.

5.1 Surface Representation Problem

The result of exploration is a point cloud. The point cloud must be represented in a map which can be used for planning manipulator-based maintenance operations. It has been shown in [63][19][24] and [25] how the manipulator trajectory planning for maintenance operations can make use of small disc-shaped targets (e.g. SLDs) generated from the collected sensor data, a point cloud. To plan the manipulator's pose, it is necessary to have a point to aim for (i.e. a target) and a surface normal. In grit-blasting maintenance, an individual blast spot is observed to be circular [18]. A straight line blasting path will generate a rectangular shape with half-circles at either end, which can be approximated by several partially overlapping discs (Fig. 5.1). It was shown in [25] that grit-blasting coverage calculations can be performed using disc-shaped targets, even in the case of rough surfaces such as in bridge maintenance where there may be rivets. A SLD-based representation can therefore enable the manipulator-based maintenance planning problem to be solved as a task optimisation problem.

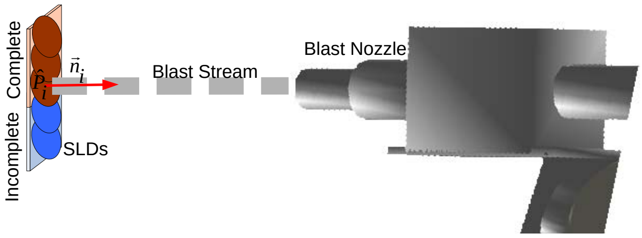


FIGURE 5.1: Grit-blasting maintenance can utilise overlapping disc-shaped targets.

In the case of a large map, the large number of SLDs which can be generated potentially makes manipulator trajectory planning difficult and inefficient. User interaction can assist by grouping the SLDs, although this interaction can be time consuming and ineffective. Therefore, an algorithm is required to cluster the SLDs based upon two considerations: the proximity of SLDs; and the manipulator pose that directs the tool on the end-effector to the desired SLD target. This section formalises the range data collection process. The next section presents an algorithm to generate SLDs which are sized and positioned in a novel way by considering motion plan task requirements to enable segmentation for efficiency and usability in manipulator operation planning. The final section presents an algorithm to cluster these SLDs to realise map segmentation.

5.1.1 Data Collection Process Formalisation

A laser range scanner is mounted on the end-effector of a robot manipulator, and makes horizontal scans along its plane of orientation. Ranges, r_i, to reflecting points on an object's surface are returned in conjunction with the bearings, θ_i, of these points. The angle between the ith and $(i+1)$th rays with angles, θ_i and θ_{i+1}, is the constant angular resolution of the scan. A scan refers to one complete set of range and angle measurements.

Given the manipulator's joint configuration, the instantaneous location and orientation of the laser scanner can be obtained with reference to a 3D coordinate system, located at the base of the robot manipulator. The end-effector's homogeneous transformation matrix, ${}^0\mathbf{T}_f(\vec{Q})$, is from the robot base to the end-effector and is a function of the manipulator's joint angle configuration, $\vec{Q}$. The transformation matrix from the end-effector to the sensor is denoted as ${}^f\mathbf{T}_s$, as detailed in Chapter 3. The location of a point on an object, $[\mathsf{x}_i, \mathsf{y}_i, \mathsf{z}_i]^T$, returned from the ith ray is given by

$$[\mathsf{x}_i, \mathsf{y}_i, \mathsf{z}_i, 1]^T = {}^0\mathbf{T}^f(\vec{Q})^f\mathbf{T}_s\,[0, -r_i \sin\theta_i, r_i \cos\theta_i, 1]^T \tag{5.1}$$

where $\mathbf{p}_i = [\mathsf{x}_i, \mathsf{y}_i, \mathsf{z}_i]$ based upon a single sensor ray in a scan, $[r_i, \theta_i]^T$.

The laser scanner is driven in a tilting movement using q_t, such that the transformation matrix, ${}^0\mathbf{T}_f(\vec{Q})$, varies. Hence, this tilting motion enables the laser sensor to deliver a set of measurements representing the location of points on surfaces of objects surrounding the manipulator. After a tilting movement, the sensor produces a set of points (i.e. a point cloud, P) as

$$\mathsf{P} = \{\mathbf{p}_i\},\ i = \{1, \cdots n\} \tag{5.2}$$

where n is the number of points sensed.

Fig. 5.2 shows a block diagram of the process that generates 3D points in the manipulator base coordinate frame from sensor range data and manipulator configuration data. The adjacency of the range data is maintained so that a surface mesh may be created.

The accurately calibrated manipulator provides time-stamped configuration data at 10Hz. Range data is continuously acquired and time-stamped while the manipulator tilts the sensor with the fifth joint.

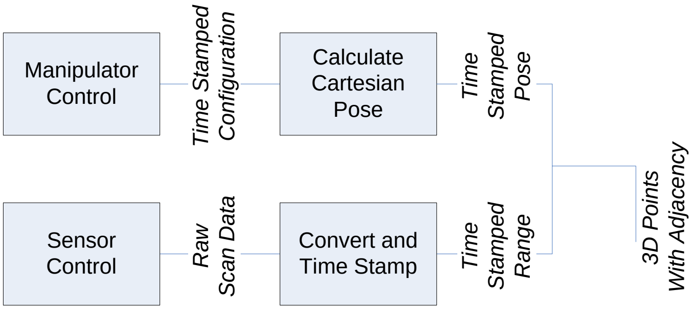


Figure 5.2: A grid of 3D points in the robot base coordinate frame is produced from manipulator configuration and laser sensor range data.

To fuse data from multiple scans into a surface, an adaptive distance field map representation (volumetric technique) has been chosen. The data fusion is based on the technique proposed by Curless et al. [42] and improved by Webb [43] for real-time and online implementation and for being able to handle thin plates and sharp features. The implementation includes a spatial index over multiple signed distance fields, implemented as an octree of small 3D grids. This provides a sparse representation to minimise memory usage and an index for efficient updates. The output of the fusion process module is a mesh map where vertices (i.e. a point cloud), P, can be rapidly queried. Fig. 5.3 shows the point cloud generated by sensing data obtained from multiple viewpoints in a simple environment.

After obtaining the point cloud, P, the goal is to segment or classify the points into n_{ss} distinct point groups PG_j, with $j \in \{1, \dots, n_{ss}\}$ such that

$$\mathsf{P} = \bigcup_{j=1}^{n_{ss}} \mathsf{PG}_j \tag{5.3}$$

where $\bigcup$ stands for set union combining the n_{ss} groups of sub-point clouds. Each group must be constructed such that points in the group can be represented by a SLD. The SLDs should also then be analysed and clustered together so that the environment map is segmented. The following section presents an algorithm to generate SLDs from a point

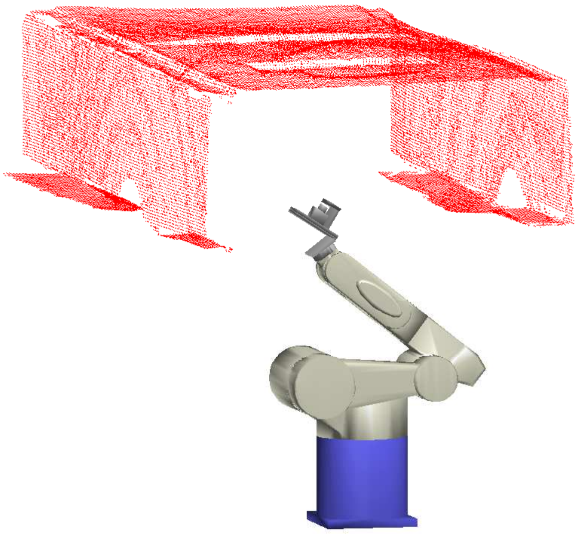


Figure 5.3: Point cloud, P, consisting of $n_p = 80977$ vertices and obtained from laser scans gathered by a robot manipulator tilting the scanner. A surface mesh (not shown) is made up of sets of 3 connected vertices which form faces.

cloud, while the second algorithm defines the connectivity of SLDs so that SLD clustering can occur and the environment map is effectively segmented as required.

5.2 Scale-Like Disc Generation

The generation of Scale-Like Discs (SLDs) is considered as both a data reduction and a dimensionality reduction problem. From a collection of 3D points, it is observed that if a subset of points could form a small planar surface, one of the dimensions vanish or a reduction of dimension is achieved. If the remaining dimensions are utilised, a point cloud (i.e. set of data) could be reduced to a representative surface normal and a 3D centre point (equivalent to a disc). To this end, an approach based on principle components analysis (PCA) is adopted. One of the benefits from using PCA is that a normal vector (describing the orientation of the surface) can be determined as the eigenvector corresponding to the minimum eigenvalue of a point cloud's covariance matrix. A further advantage is that if the centre point is carefully positioned, a surface in the environment can be constructed from a set of partially overlapping discs (SLDs). SLDs can represent complete surfaces including the surface curvature so as to predict sensing occlusions and facilitate manipulator trajectory planning, such as in maintenance-type applications. Additionally, SLDs can be used as material-type identification targets.

The SLD generation algorithm must ensure SLDs created by the PCA data reduction technique are arranged in the required partially overlapping pattern. A flowchart of the SLD generation algorithm is presented in Fig. 5.4

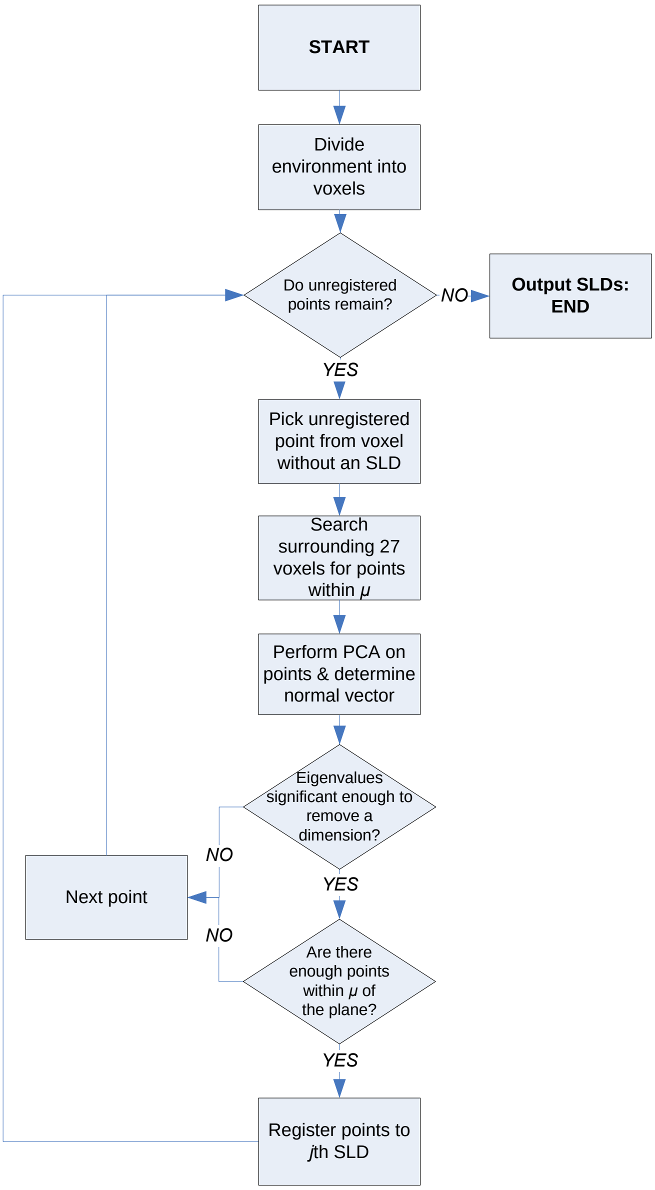
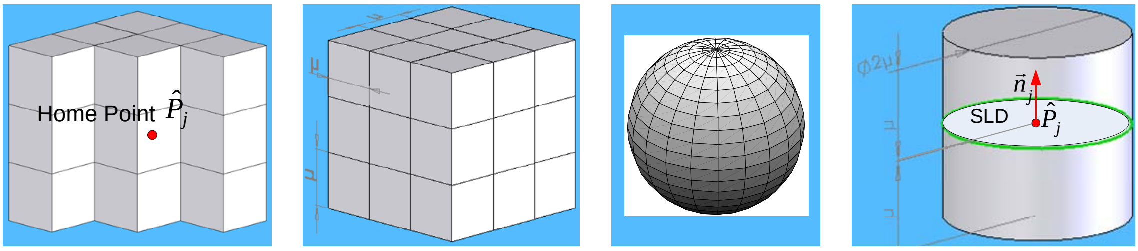


Figure 5.4: Flowchart of the SLD generation algorithm.

Initially, the environment is divided into voxels so that each 3D point is assigned to a voxel. Indexing is necessary to determine which voxel each point is associated with. Indexing can also reduce the time to determine the groups of points which may constitute a SLD. An occupied voxel may contain many points. In order to determine the vector index for each voxel, each point is divided by the voxel size and radius of the SLDs, μ, and rounding to the floor (i.e. $\lfloor \cdot \rfloor$). Thus, if the voxel size $\mu = 10mm$, then the ith point $\mathbf{p}_i = [130, -25, 35]^T$, has a voxel index $[13, -3, 3]^T$. Each point is therefore assigned to a voxel by the 3×1 index vector containing integers. To ensure the required pattern is attained, each occupied geometric voxel is limited to containing only one SLD. The radius of a SLD is also specified as μ, which is the same as the dimensions of each voxel. Hence, two adjacent occupied voxels containing 3D points can each generate a SLD such that the two SLDs will partially overlap.

The next step of the algorithm determines if there is a voxel with an occupied state that does not contain a SLD. Then an unregistered point is randomly selected from this voxel and defined as the 'home point' (Fig. 5.5a) and is denoted as $\hat{P}_j$. Following this, the voxel index is used to search the enclosing 1 and surrounding $3^3 - 1 = 26$ voxels (Fig. 5.5b). A set of points is then found, PG_j, within the specific distance, μ, from the initial home point. This is equivalent to the spherical region $(4/3\pi\mu^3)$ around the home point as shown in Fig. 5.5c.



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

FIGURE 5.5: a) Home point of the centre voxel; b) Calculate distance to points in centre voxel and in the surrounding $3^3 - 1 = 26$ voxels; c) Perform PCA on points within μ of home point to determine a plane; d) Register points within $\pm\mu$ of the plane to the SLD.

PCA must then be performed on each sub-point cloud, PG_j, so a normal of the point cloud, $\vec{n}_j$, can be extracted. The measurements obtained from the laser sensor give a 3D description of the location of a point on an object's surface. Therefore the dimension is

$M = 3$. The goal is to reduce this dimension to $M = 2$, which is in 2D such that the surface is still adequately represented.

Let there be ${}^{j}n_p$ points observed from a subset point cloud, PG_j, returned from the laser sensor and transformed into global coordinates,

$$\mathsf{PG}_j = \{\mathbf{p}_i\},\ i = \{1, \cdots {}^{j}n_p\}, \tag{5.4}$$

The mean, $\overline{\mathsf{PG}}_j$, is first calculated as

$$\overline{\mathsf{PG}}_j = \frac{1}{{}^{j}n_p} \sum_{i=1}^{{}^{j}n_p} \mathbf{p}_i. \tag{5.5}$$

Where $\overline{\mathsf{PG}}_j$ is a 3D vector and the difference between the sub-point and the mean is

$$\mathbf{\Delta} = \mathsf{PG}_j - \overline{\mathsf{PG}}_j \times \mathbf{1}, \tag{5.6}$$

where $\mathbf{\Delta}$ is a $3 \times {}^{j}n_p$ matrix and $\mathbf{1} = [1, \cdots, 1]$ is vector containing ${}^{j}n_p$ values of 1.

A covariance matrix, C, representing the spread of the points is calculated from

$$\mathbf{C} = \frac{1}{{}^{j}n_p} \mathbf{\Delta}\mathbf{\Delta}^T,\ \mathbf{C} \in \mathbb{R}^{3\times3} \tag{5.7}$$

In order to determine the principal components of the matrix, the covariance matrix, C, is used to determine the eigenvectors, V, which diagonalises C as in

$$\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{D} \tag{5.8}$$

where $D_{i,i}$ is equivalent to the ith eigenvalues, λ_i, and where the eigenvectors are $V = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$.

The magnitude of the eigenvalues has a significant role in the representation of a data set. In conventional applications of PCA, the dimensions of data corresponding to larger eigenvalues are selected to represent the data set. Thus, the eigenvalues are further sorted in descending order. A dimensionality reduction is possible in this case if the ratio between the largest and smallest eigenvalue, and the ratio between the middle and smallest eigenvalue is such that the third dimension of the data is not required to describe the sub-point cloud. It is then possible to remove the dimensions corresponding to the third eigenvalues. Therefore, the data can be described as being on the plane represented by the remaining eigenvalues. Thus, the dimensionality of the dataset is reduced from a 3D point cloud to 2D plane.

For the problem considered in this work, a disc must be extracted from a set of laser scanned points in 3D space. The dimension perpendicular to the disc contains less information than the dimensions which are parallel to the plane. This is since the points on the disc are dense along the perpendicular dimension. In contrast to conventional PCA applications, the eigenvector that corresponds to the minimum eigenvalue is selected to represent the normal vector, $\vec{n}$. That is

$$\vec{n} \equiv v_i, \tag{5.9}$$

where v_i is selected based on the corresponding minimum eigenvalue, λ_i. In the case of the 3D points, the 'surface normal' (i.e. the vector perpendicular to the surface) originates from the centre of the disc, $\hat{\mathsf{P}}$ (note that $\hat{\mathsf{P}}$ can be different from the centre of mass of the sub-point cloud). Hence, a single SLD is characterised by $\{\hat{\mathsf{P}}, \vec{n}\}$. All points, P_j, within μ of the home point and μ of the plane (Fig. 5.5d) are registered to the jth SLD. Figure 5.6 presents a single disc which is generated based upon a sub-point cloud.

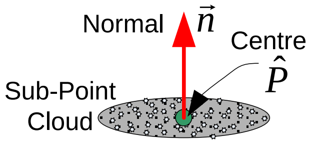


FIGURE 5.6: Single SLD: centre and surface normal, $\{\hat{\mathsf{P}}, \vec{n}\}$, represents a sub-point cloud.

5.2.1 SLD Parameter Tuning and Simulation Results

This section presents simulation results obtained from SLD generation tests. Different sizes of point clouds collected in exploration are used and the parameters for the SLD generation algorithm are investigated. In particular, the results are used to determine the radius, μ, of a SLD, and the minimum number of points which constitute a SLD, n_{min}. As the parameters of μ and n_{min} are varied, the characteristics of SLD can be observed. The results can be used to make comparisons of the number of SLDs generated, the time taken for a given implementation, and the percentage of points registered for the SLDs generated.

In Fig. 5.7 the effects of varying the SLD's radius, μ, are shown. Values of $\mu \leq 50mm$ were investigated. Fig. 5.7a shows how the computational time changes. It is observed that if the size of a SLD is less than $10mm$, then the time to create the SLDs is significant, but if greater than $20mm$ then the time taken is generally less than a second. Fig. 5.7b shows how the number of SLDs, n_{ss}, peaks when the size is $7mm$. This is due to the parameter for the minimum number of points which constitutes a SLD, $n_{min} = 8$. With small values of $\mu < 7mm$, many surfaces are not valid and cannot be generated due to the limited density of point cloud data on the surfaces. A value greater than $\mu = 7mm$ reduces the number of SLD targets. Fig. 5.7c shows how the percentage of points correctly registered to a SLD is increased as μ is increased. This is consistent with Fig. 5.7b where values of $\mu < 7mm$ result in a decrease in SLDs generated. A value of $\mu > 10mm$ is shown to register 100% of points.

The value of μ was tuned empirically based upon the application requirements, however various additional considerations must also be made when selecting μ. The specified laser sensor's error is $\pm 10mm$, which can result in surface inaccuracies or rough surfaces. Therefore, a value of μ larger than $10mm$ can analyse more surrounding data points and reduce the effect of spurious data. For the SLD to be used in maintenance operations such as grit-blasting, the nozzle tool must be considered. A blast spot can cover a radius of between $10mm$ and $40mm$ depending upon the nozzle size [18].

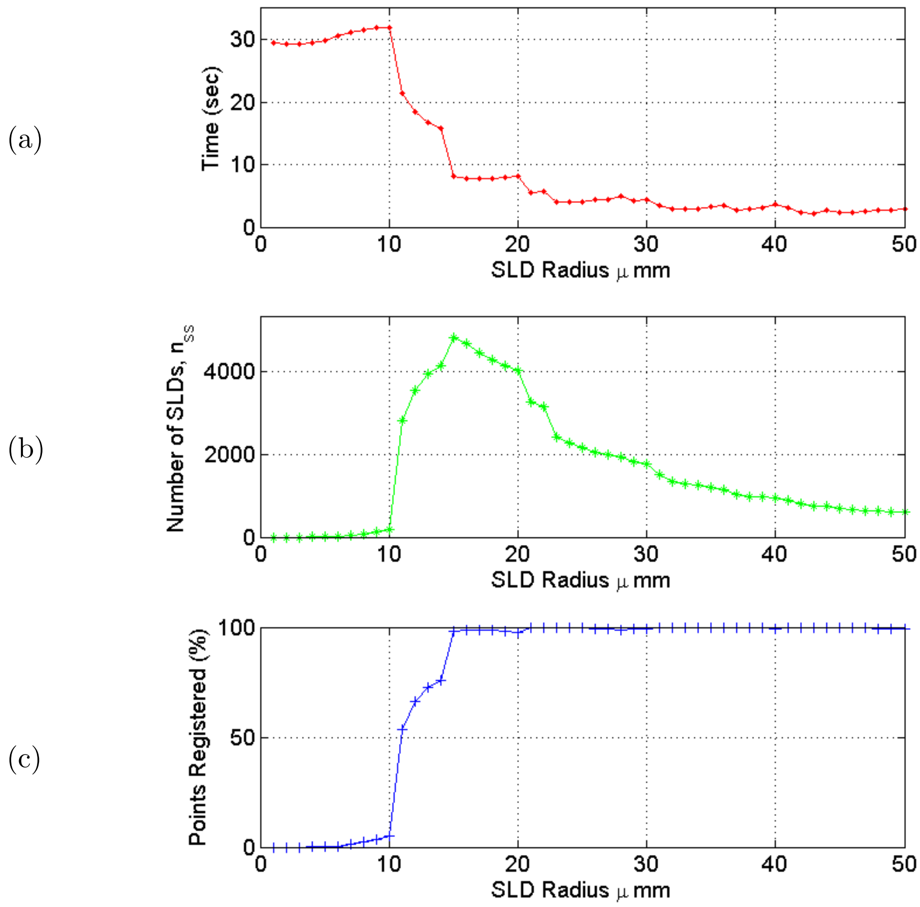


FIGURE 5.7: With respect to changing μ(mm). *a*) Time taken to generate surfaces; *b*) Number of small SLDs created n_{ss}; *c*) Percentage (%) of point registered.

A second experiment was performed which varied the minimum number of points that constitute a point cloud, while fixing $\mu = 25mm$ (equivalent to the middle of the range of the previous experiment). The results are shown in Fig. 5.8. Fig. 5.8a shows how the computational time changes as n_{min} is varied. If $n_{min} > 20$, then the time to create the SLDs gradually begins to increase. Fig. 5.8b shows how the number of SLDs, n_{ss}, is similar until $n_{min} > 20$, then this begins to decrease. Fig. 5.8c shows how the percentage of points begins to decrease at a similar value of n_{ss} to when the first two graphs began to change significantly. This is linked to the density of points on a surface. Hence, if too many points are required to form a SLD, then the algorithm may be unable to successfully generate SLDs. Appendix D presents the extended versions of results in Fig. 5.8 and Fig. 5.7, where both n_{min} and μ are varied.

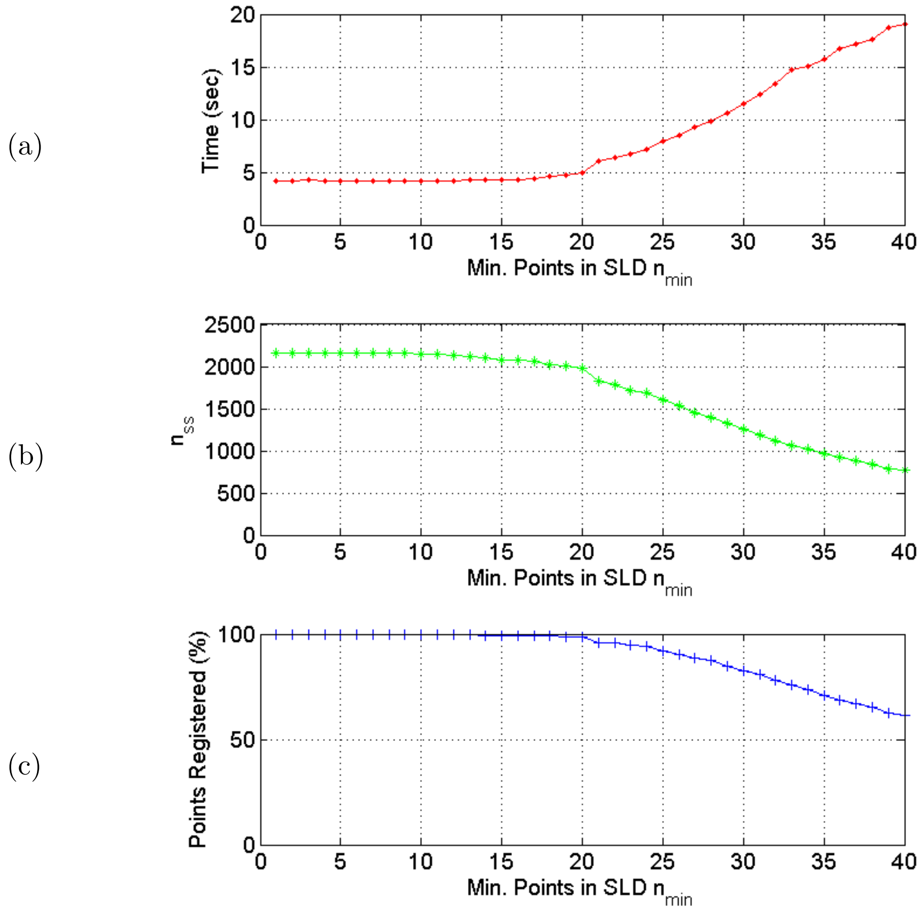


Figure 5.8: With respect to changing n_{min}. a) Time taken to generate surfaces; b) Number of small SLDs created n_{ss}; c) Percentage (%) of point registered.

Fig. 5.9 shows the SLD generation in 50 test cases for surfaces with different point cloud sizes of $1 \times 10^4 \rightarrow 2 \times 10^5$. For this experiment, the parameters of $n_{min} = 20$ points and $\mu = 40mm$ were used. The results in Fig. 5.9a show the overall time spent for the processing over the different number of points. Fig. 5.9b shows the resulting number of planes produced. Note that the percentage of points registered stayed above 99% across the tests and hence is not shown. These results show that the data needed to describe the surface sufficiently is significantly reduced - turning 100000 points into 1000 SLDs. In Fig. 5.9, as the number of points increase, both the time and the number of SLDs increase linearly.

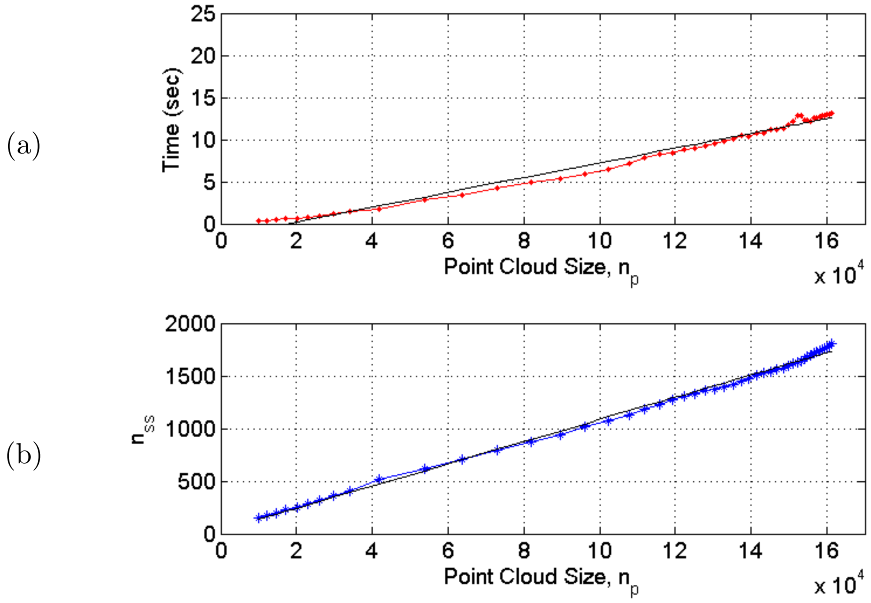
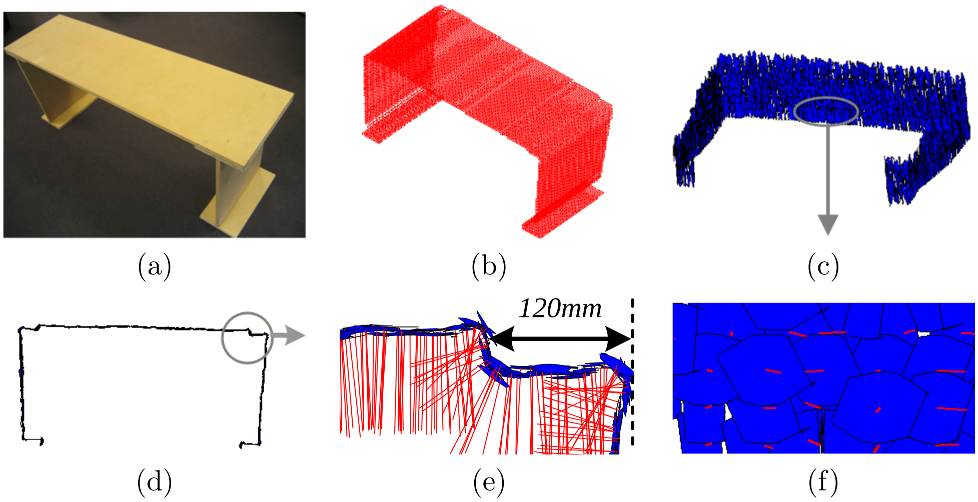


Figure 5.9: Results of SLD generation when increasing the number of points in the point cloud, n_p. a) Time taken (red $\cdot$) and the trend line (black line); b) Number (no.) of planes created (blue *).

Fig. 5.10 presents a test case on a maintenance-like I-beam channel structure including features such as corners, flanges and webs. Several scans are taken from the underside of Fig. 5.10a to generate a point cloud as shown in Fig. 5.10b. The SLD generation algorithm was performed and the outputs displayed: Fig. 5.10d shows the front view with a close-up showing the normals (Fig. 5.10e); and Fig. 5.10c shows a side view of SLDs created with a close-up of the plotted SLD in Fig. 5.10f. The parameters are shown in Table 5.1.

Table 5.1: SLD Generation Test Case Parameters

Parameter	n_p	μ	n_{min}	% registered	Time	n_{ss}
Value	54916	40mm	20 points	99.87	5.5 secs	1636



(a) (b) (c)

(d) (e) (f)

Figure 5.10: *a*) Photo of example structure; *b*) Fused raw data point cloud; *c*) Bottom right view; *d*) Front view of planes created; *e*) Expanded view of the corner; *f*) Expanded view of the front - showing SLDs and the associated normals (red lines)

The final experiment is to show how the SLDs result in a compact data representation for the purposes of efficiency. The values for n_{min} and μ which were determined in the previous section are used for this experiment. Fig. 5.11 shows the 3D data collected from 20 arbitrarily selected manipulator viewpoints. The data is fused together via the surface mapping technique to produce a fine resolution surface mesh [42]. Vertices are extracted and SLDs are generated. Each time the reflector mirror in the laser scanning sensor rotates through one revolution, this is known as a “scan” [40]. For each scan there are up to 342 individual usable rays returned. The maximum number of returns is achieved when all rays are reflected from surfaces within the sensor’s range. The number of scans depends upon the tilt angle, α. The results are graphed in Fig. 5.11 on a log scale. The cumulative number of points collected is shown as $(\cdot)$. The surface mesh collected from a viewpoint is fused to create a mesh map consisting of vertices and faces, although only the number of vertices is shown as $(+)$. The number of SLDs which are generated is shown as $(*)$. Both the mesh map and the SLDs demonstrate how the data converges while the raw data continues to increase. The quantity of data required for SLD (normal, $\vec{n}$, and home point,

$\hat{\mathsf{P}}$) is between one and two orders of magnitude less than the original point cloud data. The SLD generation algorithm presented fulfils the requirement of reducing the data and providing a compact and useful representation of surfaces in an environment. These SLDs are an effective representation of a map of a partially known or unknown environment consisting of predominantly planar surfaces.

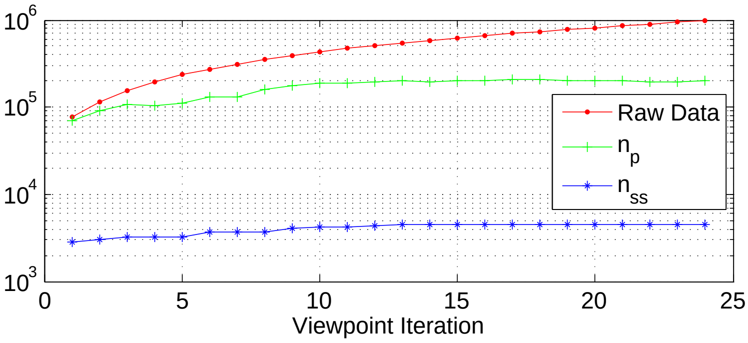


FIGURE 5.11: Data reduction: Number of raw data points collected (.); Mesh vertices, n_p, (+); Number of SLDs, n_{ss}, (*).

5.3 Map Segmentation

One challenge identified when exploring and mapping a complex, partially known or unknown 3D environment is to generate a usable map of the environment. In autonomous steel bridge maintenance by means of an intelligent robotic system, [25][19][26], map segmentation can improve manipulator trajectory planning efficiency. Therefore, the map segmentation algorithm must consider the movement of the manipulator over the surface. If a segment is too large, then the manipulator and the blasting tool may not be able to reach every point of the segment because of the joint limitations. Where there are many SLD targets covering a large segment of the map, it has been observed that it is challenging to keep the end-effector tool directed at the targets without contorting the manipulator. If the segments are too small, the maintenance tool (i.e. a grit-blasting nozzle) must be frequently turned on and off when moving between segments. Therefore, segments of a map must ideally be as large as possible while ensuring that the manipulator (including the maintenance tool) can reach every SLD target in a segment.

As presented in the previous section, the ith SLD consists of a surface normal $\vec{n}_i$ and a centre point $\hat{P}_i$. In order to cluster SLDs into segments, an algorithm based on connectivity of SLDs is devised. A graph is created by linking (i.e. connecting) SLDs based upon a set of rules. These four rules are inferred from: the angular difference between the normals of SLDs; the distance between centre points of SLDs; the average distance between SLD centre points and other SLD planes; and the angular difference between the manipulator poses which enable the manipulator to act (i.e. perform a task) upon a SLD target.

Consider the case shown in Fig. 5.12 where there are five SLDs. Even if SLDs 1 and 5 would not be connected directly, if the manipulator poses for all SLDs were known to be similar, then a manipulator trajectory could be planned from SLD 1 to 5 through SLDs 2, 3 and 4. Fig. 5.12 may then be regarded as a single map segment, assuming that the remaining three aforementioned conditions of SLD connectivity are met. Henceforth, the connectivity of the set of SLDs will be formulated.

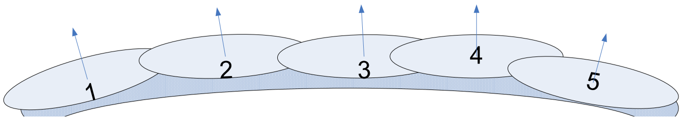


FIGURE 5.12: Five SLDs with normal vectors shown, represent a curved surface in the environment. The connectivity of the SLDs must be determined so they can be clustered into segments for ease of use by the manipulator path planner.

5.3.1 Segmentation Formulation

5.3.1.1 Angle Between Surface Normals

Every SLD has a surface normal which defines its orientation. Using this normal vector, $\vec{n}$, it is possible to determine the angle between neighbouring SLDs. As shown in Fig. 5.13, for SLD $\mathbf{s}_i$ and $\mathbf{s}_j$, the angular difference, $\delta\theta_{ij}$, of their surface normals is calculated as

$$\delta\theta_{ij} = \mathbf{acos}(\vec{n}_i \cdot \vec{n}_j), i = \{1, \cdots, n_{ss}\}, j = \{1, \cdots, n_{ss}\} \tag{5.10}$$

where n_{ss} is the number of small SLDs. The n_{ss} by n_{ss} angular difference matrix can then be formed as $\delta\boldsymbol{\Theta} = [\delta\theta_{ij}]$ with $\delta\theta_{ij} = \delta\theta_{ji}$ and $\delta\theta_{ii} = 0$.

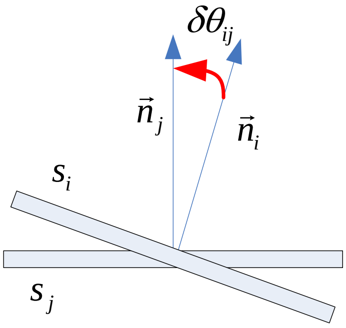


FIGURE 5.13: Angle difference, $\delta\theta_{ij}$, between normals on the ith and jth SLDs.

It is possible to compare this matrix of angles to a constant threshold, τ_θ, which is the maximum allowable angular difference between the normals of any two SLDs, $\mathbf{s}_i$ and $\mathbf{s}_j$. If $\delta\theta_{ii} \leq \tau_\theta$, a link between SLDs i and j is built. This will give a binary connection graph, ${}^{\delta\theta}\mathcal{B} = \{{}^{\delta\theta}b_{ij}\}$, which is an $n_{ss} \times n_{ss}$ matrix of binary values with ${}^{\delta\theta}b_{ij} = 0$. That is

$$
{}^{\delta\theta}b_{ij} = \begin{cases} 1, & \delta\theta_{ij} \leq \tau_\theta \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}
$$

5.3.1.2 Distance Between Centres of SLD

Denoting the position of centres of two SLD's as $\hat{\mathsf{P}}_i$ and $\hat{\mathsf{P}}_j$, the Euclidean distance vector between them (Fig. 5.14) is found by

$$
\delta d_{ij} = \|\hat{\mathsf{P}}_i - \hat{\mathsf{P}}_j\|, \tag{5.12}
$$

where $\|\cdot\|$ is the Euclidean distance.

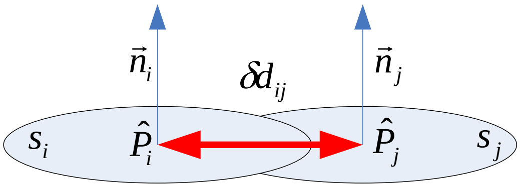


FIGURE 5.14: Distance between centres of the ith and jth SLDs is $\delta d_{ij} = \|\hat{\mathsf{P}}_i - \hat{\mathsf{P}}_j\|$.

This can also be transformed into a binary connectivity graph, ${}^{\delta d}\mathcal{B} = \{{}^{\delta d}b_{ij}\}$, such that

$$ {}^{\delta d}b_{ij} = \begin{cases} 1, & \delta d_{ij} \leq \tau_d \\ 0, & \text{otherwise.} \end{cases} \tag{5.13} $$

5.3.1.3 SLD Centre-to-Plane Distance

The normal from the centre of the jth SLD intersects with the ith SLD plane at $\hat{\mathsf{P}}_{j\rightarrow s_i}$. Conversely, the normal from the centre of the ith SLD intersects with the jth SLD plane at $\hat{\mathsf{P}}_{i\rightarrow s_j}$. The average distance between the points of intersection and the respective centres of the SLDs is denoted as δp_{ij} and calculated as

$$ \delta p_{ij} = \frac{\left\| \hat{\mathsf{P}}_{j\rightarrow s_i} - \hat{\mathsf{P}}_j \right\| + \left\| \hat{\mathsf{P}}_{i\rightarrow s_j} - \hat{\mathsf{P}}_i \right\|}{2} \tag{5.14} $$

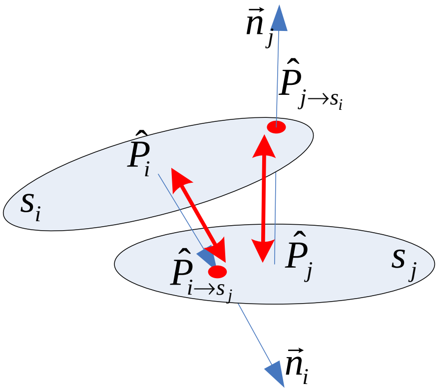


FIGURE 5.15: Showing the point of intersection between the normal of the ith SLD and the jth SLD's plane, $\hat{\mathsf{P}}_{i\rightarrow s_j}$, and the point of intersection between the jth SLD's normal and the ith SLD's plane, $\hat{\mathsf{P}}_{j\rightarrow s_i}$.

The average of two SLD centre-to-plane distances have to be within a limiting threshold, τ_p. The binary graph is constructed from ${}^{\delta p}\mathcal{B} = \{{}^{\delta p}b_{ij}\}$, where

$$ {}^{\delta p}b_{ij} = \begin{cases} 1, & \delta p_{ij} \leq \tau_p \\ 0, & \text{otherwise.} \end{cases} \tag{5.15} $$

5.3.1.4 Difference Between Manipulator Poses

For the ith SLD there is an associated manipulator pose $\vec{Q}_i$. The joint angles for a manipulator pose are determined for each SLD target using an optimisation approach similar to the one used in material-type identification in Stage Two of exploration [25]. The method for manipulator trajectory planning is outside the scope of this thesis but is briefly outlined in Appendix A. The angular difference between joints used in a manipulator pose for SLDs i and j is another measure for map segmentation. When the angular difference for each joint is small, then the manipulator movement required from SLD i to SLD j is also small, which is a required consideration in map segmentation. The joint difference between poses $\vec{Q}_i$ and $\vec{Q}_j$ is formed as

$$\delta Q_{ij} = \left| \vec{Q}_i - \vec{Q}_j \right| \tag{5.16}$$

where $|\cdot|$ denotes the angular difference between two manipulator poses such that $\delta Q_{ij} = [\delta q_{1,ij}, \delta q_{2,ij}, \delta q_{3,ij}, \delta q_{4,ij}, \delta q_{5,ij}, \delta q_{6,ij}]^T$.

The binary graph ${}^{\delta q}\mathcal{B} = \{{}^{\delta q}b_{ij}\}$, is constructed as

$${}^{\delta q}b_{ij} = \begin{cases} 1, & \delta q_{k,ij} \leq \tau_{q_k}, k \in \{1, \dots, 6\} \\ 0, & \text{otherwise.} \end{cases} \tag{5.17}$$

5.3.2 Map Segmentation by Means of SLD Clustering

Based upon the four binary graphs, one single connectivity binary graph (i.e. an $n_{ss} \times n_{ss}$ binary matrix) is generated.

$$\mathbf{B} = \left\{ {}^{\delta\theta}\mathcal{B} \wedge {}^{\delta d}\mathcal{B} \wedge {}^{\delta p}\mathcal{B} \wedge {}^{\delta q}\mathcal{B} \wedge (\neg \mathbf{I}) \right\} \tag{5.18}$$

where $\wedge$ is the intersection between the binary graphs such that $[1] \wedge [0] = [0]$ and $[1] \wedge [1] = [1]$. Also, $\neg\mathbf{I}$ is the logical negation of the identity truth matrix, which forms an n_{ss} by n_{ss} matrix of ones (true) with the diagonal as zeros (false). Therefore, the self-connectivity is disabled by this Boolean combinatorial process. This is notionally similar to penalising self links, and consequently $\mathbf{B}_{i.i} = 0$ and so is never linked in the graph. It is from this combined graph that searches will be performed to generate the required geometric map segments.

The links in the graph, $\mathbf{B}$, in effect represent the connection between two SLDs. Generally the graph is expected to consist of several unconnected sub-graphs. A search through the graph will use a breadth-first search methodology and thus produces clusters of connections corresponding to sub-graphs, where each SLD has a link to at least one other SLD in the sub-graph. These clusters of SLDs then constitute a segment of the map. The number of SLD clusters, n_s, is equivalent to the number of segments in the geometric map. Algorithm 1 presents how the connectivity between each SLD and the SLDs which they are linked to, is checked recursively until there are no more unregistered SLDs in the sub-graph. Following this, a new segment is formed starting from an unregistered SLD in another sub-graph, and then the search begins again.

Algorithm 1 Clustering SLDs into map segments

```
 1: Set the number of segments, $n_s$, to zero
 2: Enqueue a random unregistered root
 3: Increment $n_s$
 4: for each node in the current queue do
 5:    Dequeue a node and examine it
 6:    Check other nodes connected via combined binary graph $\mathbf{B}$
 7:    Enqueue these nodes to the stack of the $n_s$th segment
 8:    if there are no more nodes to dequeue then
 9:       Segment is complete
10:       Increment $n_s$
11:    end if
12: end for
13: if all $n_{ss}$ small SLDs have been registered to segments then
14:    FINISH
15: else
16:    Repeat from Step 2
17: end if
```

5.3.3 Map Segmentation Demonstration

This section presents a demonstration of map segmentation based upon real data collected during exploration. In this case a number of vertices, $n_p = 10637$, were extracted from an existing mesh map. Using the points which are inside the manipulator's work envelope (as detailed in Chapter 4), the SLD generation algorithm created $n_{ss} = 433$ SLDs (Fig. 5.16). Then the segmentation algorithm was implemented. The parameters used in this demonstration are shown in Table 5.2

TABLE 5.2: Map Segmentation Demonstration Constraint Parameters

Parameter	**Purpose**	**Value**
μ	Radius of the SLDs	40mm
n_{min}	Ensure SLDs contain enough points	8 points
τ_θ	Min. angle between surface normals	$5 \times \frac{\pi}{180}$ rads
τ_d	Min. distance between SLD centres	200mm
τ_p	Min. average SLD centre-to-plane dist.	50mm
$\tau_{\mathbf{q}}$	Min. angle between manipulator poses	$8, 10, 14, 20, 20, 20°$

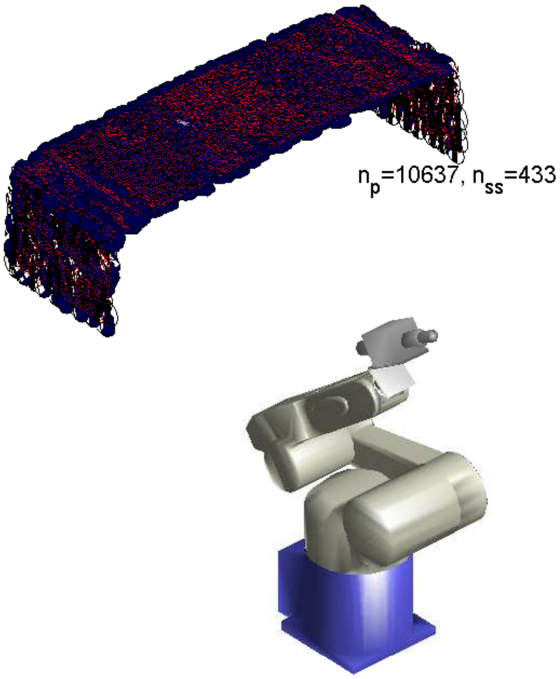
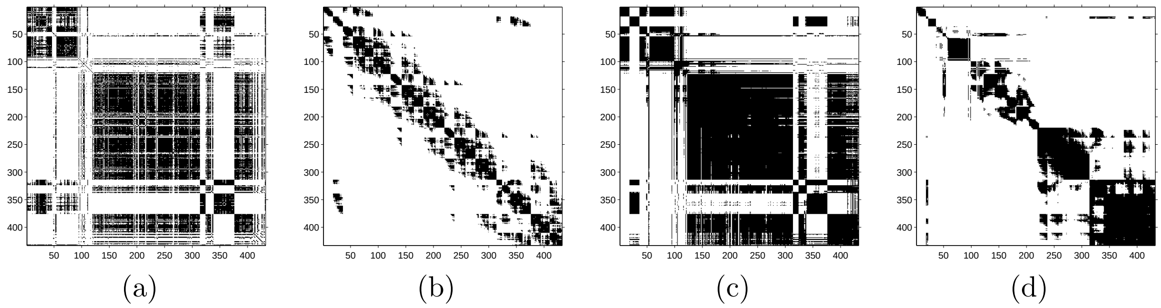


FIGURE 5.16: The point cloud, P, with $n_p = 10637$ vertices represented by $n_{ss} = 433$ small SLDs. These SLDs are overlayed on the original point cloud as discs. The manipulator is also shown for scale.

The individual binary graphs (${}^{\delta\theta}\mathcal{B}, {}^{\delta d}\mathcal{B}, {}^{\delta p}\mathcal{B}, {}^{\delta q}\mathcal{B}$) are shown in Fig. 5.17. From this figure it can be seen that the clusters of connectivity are different for the four graphs. Note how the binary graphs shown are symmetrical about the diagonal (i.e. $\mathcal{B}_{i,j} = \mathcal{B}_{j,i}$). For the surfaces on the same plane, ${}^{\delta p}\mathcal{B}$, in Fig. 5.17*c* there are three main clusters since the example contains three predominant flat surfaces. Similarly, the angular difference between surface normals in Fig. 5.17*a*, shows three main clusters. The distances between the centres of the SLDs (Fig. 5.17*b*) only have local connectivity, which is logical for this proximity constraint. Conversely, the C-space constraint (Fig. 5.17*d*) results in a different pattern, meaning the poses of the manipulator may be similar for completely different SLDs. In Fig. 5.17*d* there is some distinct clustering at the bottom right, between nodes of higher node index. The overall connectivity binary graph is shown in Fig. 5.18.



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

FIGURE 5.17: Binary connectivity graphs for the connectivity of SLDs: *a*) Angle between normals, ${}^{\delta\theta}\mathcal{B}$; *b*) Distance between centres, ${}^{\delta d}\mathcal{B}$; *c*) Average distance from an SLD centre to another SLD plane, ${}^{\delta p}\mathcal{B}$; *d*) Manipulator pose C-space distance, ${}^{\delta q}\mathcal{B}$.

Using the map segmentation algorithm, five clusters (map segments) are found. Two segments are shown in Fig. 5.19. The first segment is made up of $n_{ss} = 150$ small SLDs (Fig. 5.19*a*). These SLDs represent $n_p = 4098$ points on the top right of the manipulator as shown in Fig. 5.19*b*. Fig. 5.19*c* shows the histogram of joint angles for the manipulator poses which direct the tool at the end-effector at the SLD targets. These histograms show that the poses for the segment are similar. As shown in Fig. 5.19*b* and Fig. 5.19*c*, this map segment consists of SLD targets clustered according to their similarities in geometry and manipulator joints. The second segment of $n_{ss} = 115$ small SLDs which replaced the $n_p = 2902$ points from a point cloud is shown in Fig. 5.19*d, e* and *f.* It can be seen from these two segments that segments are not necessarily regular geometric shapes due to the consideration of pose selection being in C-space. However, the segments are still logical in terms of both the proximity and orientation of the SLDs. Also, the SLDs are clustered into segments based upon the similarity of manipulator poses.

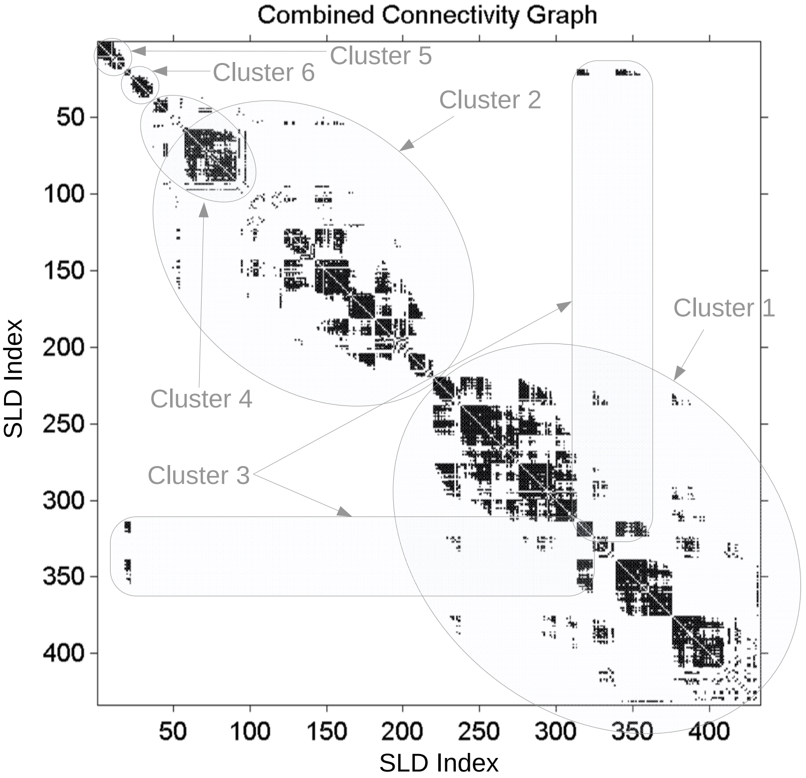
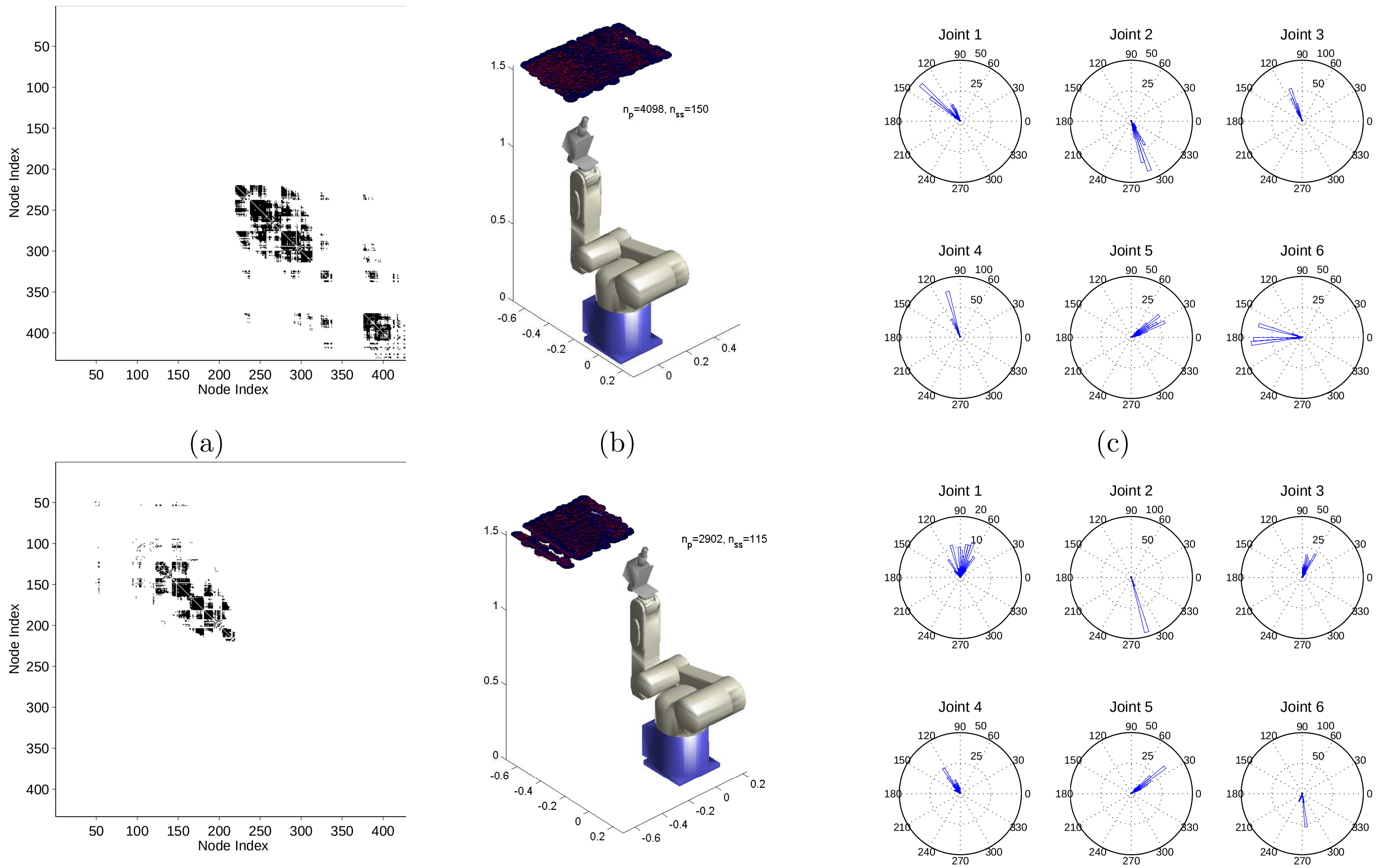


Figure 5.18: Overall connectivity graph between SLDs, **B**. The graph is binary (black = connection, white = no connection), symmetric, and nodes clusters are visible.



(a) (b) (c)

(d) (e) (f)

FIGURE 5.19: *a*) The clustered connectivity for the first segment; *b*) The first segment in relation to the manipulator; *c*) Angular histogram for each joint of the poses for the first segment. This shows that manipulator poses are similar. *d*) The connectivity for the second segment; *e*) The second segment in relation to the manipulator; *f*) Angular histogram for each manipulator joint. Note that the angular histograms are different to (*c*).

5.4 Discussion and Conclusions

A Scale-Like Disc generation and environment segmentation technique has been presented to represent surface data collected from exploration. The SLD generation algorithm has been shown to convert over 10000 unordered points into approximately 400 SLDs. This process takes one second in the demonstrated implementation. After the completion of exploration, several thousand SLDs can be clustered to form map segments. Assuming that manipulator poses are available for each SLD target, the segmentation algorithm occurs in less than a second.

The SLD generation algorithm occurs in addition to mesh fusion techniques and does not order the SLDs in a particular sequence. The environment segmentation is based upon the proximity and orientation of the SLDs and the proximity of the associated manipulator pose. The environment segmentation is limited by the time taken to obtain manipulator poses from a pose selection module ($\approx 10ms$ per pose).

Sensor data gathered from exploration is probabilistically fused into a surface mesh, and then the vertices are compacted into the exploitable form of SLDs. These SLD targets are clustered in a novel way by considering task requirements to enable map segmentation. SLDs and the segments can then be effectively utilised to fulfil the requirements for exploration and planning tasks. The SLD representation and map segmentation was shown to enable data reduction, exploration information predictions, material-type identification and manipulator trajectory planning so manipulator-to-surface actions can be planned.

Chapter 6

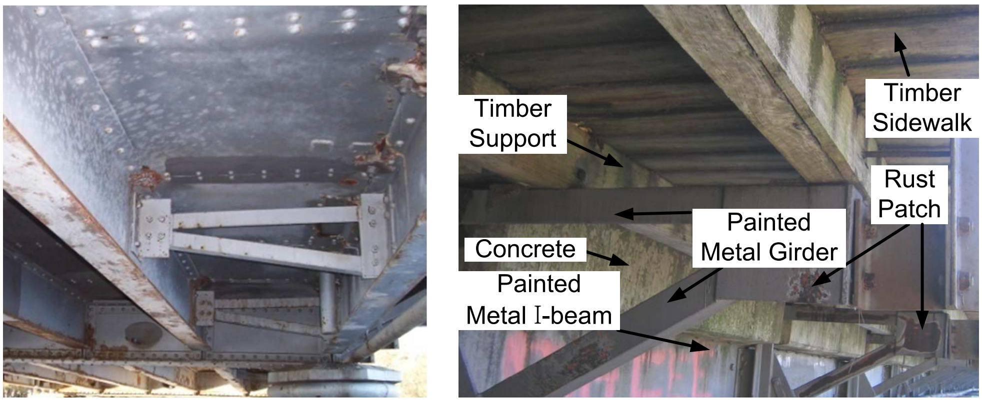
Case Study: Exploration and Map Building of Complex Steel Bridge Structural Environments

The developed exploration approach, algorithms and models are integrated into a system named the Autonomous Exploration to Build A Map system (AXBAM). The AXBAM system is tasked with the exploration and mapping of real-world, structural environments which are not structured for the robot application. The AXBAM system serves to verify the presented exploration approach. Three experiments are designed. This chapter presents the experimental results, including the details of the objective functions used in the two-stage exploration. The multiple objective functions in Stage One, based on the geometry and C-space information remaining and the safety and efficiency of manipulator movements, allow for the selection of a manipulator pose out of a solution space. Stage Two is commenced immediately after Stage One and focuses on the material-type identification of surfaces. Resulting maps of the environment have been successfully used in trajectory planning for a steel bridge maintenance robot.

6.1 Experiment Setup

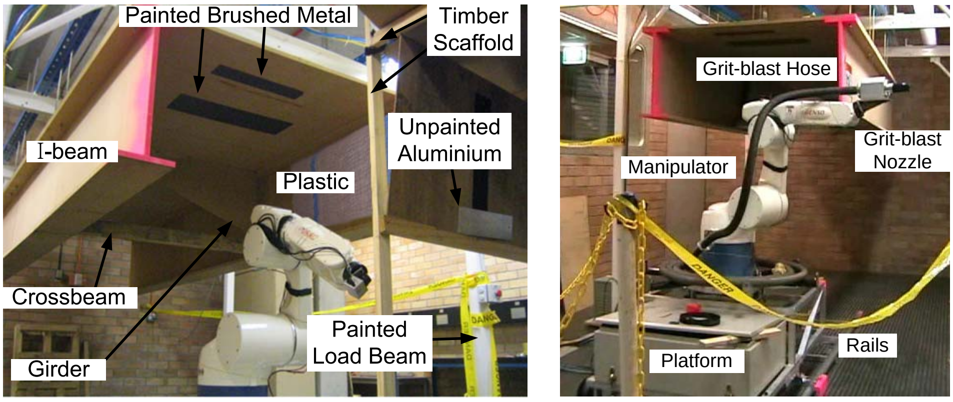
6.1.1 The Environment and The Robot

The application is in an environment of a steel bridge that requires grit-blasting maintenance. A 1 : 1 scale replica of a single channel underneath the bridge shown in Fig. 6.1*a* has been built and suspended in a laboratory (Fig. 6.2*a*). In the most basic case the environment consists of I-beams running parallel with a 'roof' connecting them. There are often crossbeams or girders in the structure. Typical materials in the bridge include: plastic to encase the scaffolding; painted / rusted metal; timber; and concrete (Fig. 6.1*b*).



(a) (b)

Figure 6.1: Environment set-up with several materials highlighted.
a) The "Unwins Bridge" channel; *b*) Various material-types in the bridge.



(a) (b)

Figure 6.2: *a*) The replica bridge maintenance environment and; *b*) The prototype grit-blasting robot and platform.

Fig. 6.2*b* shows that the robot manipulator is placed on a platform which can move on a rail track. Although the rail allows the movement of the robot along the I-beam channel, this extra degree of freedom is not available to the AXBAM system. The platform is stationary during exploration and grit-blasting due to the concern of stability and safety of the platform. During exploration the manipulator is equipped with the sensor tool instead of the grit-blasting nozzle.

A Denso VM-6083 manipulator arm and a Hokuyo URG-04LX Scanning Laser Range Finder (i.e. the sensor tool) are included in the set-up. The sensor tool is mounted on the end-effector of the manipulator. This is interchangeable with the grit-blasting tool. The computer used is a dual core P1.66GHz "Mac mini". The implementation code is written in a combination of C++ and MATLAB. Manipulator joint positions, time stamps and laser scanner sensor data are all processed online by a single computer. Further details of the robotic system are presented in Appendix C. Initially, the whole platform is placed in a start point and the robot does not have any knowledge of the environment except that the initial starting pose is safe. The robot starts to scan the environment by moving joint 5 from the initial pose (viewpoint 0).

6.1.2 AXBAM System Implementation

The AXBAM system is implemented according to the flowchart shown in Fig. 6.3. Initially, the sensor-equipped manipulator is placed into an unknown environment such that only one safe scan can be taken. At each subsequent viewpoint, the available map is analysed for future viewpoint selection. In each stage of exploration, a viewpoint selection optimisation problem is solved. In Stage One, information predictions are performed based upon the state of the surfaces, the occupancy map, and the information remaining. This is combined with the planning of manipulator trajectories from viewpoint to viewpoint. At each viewpoint, with the acquisition of new data, the map is updated and then reutilised in the selection of the next viewpoint. This process continues until the geometric information remaining is no longer decreasing significantly. The output of Stage One is SLDs that represent the surfaces in the environment. Some information on the material-types is also collected. Stage Two of exploration (Fig. 6.3 *right*) determines viewpoints which can direct sensing efforts at the SLDs so that the required information on the material-types of surfaces can be collected within the sensor's limited field of view. Thus, the material-type information remaining is gradually reduced.

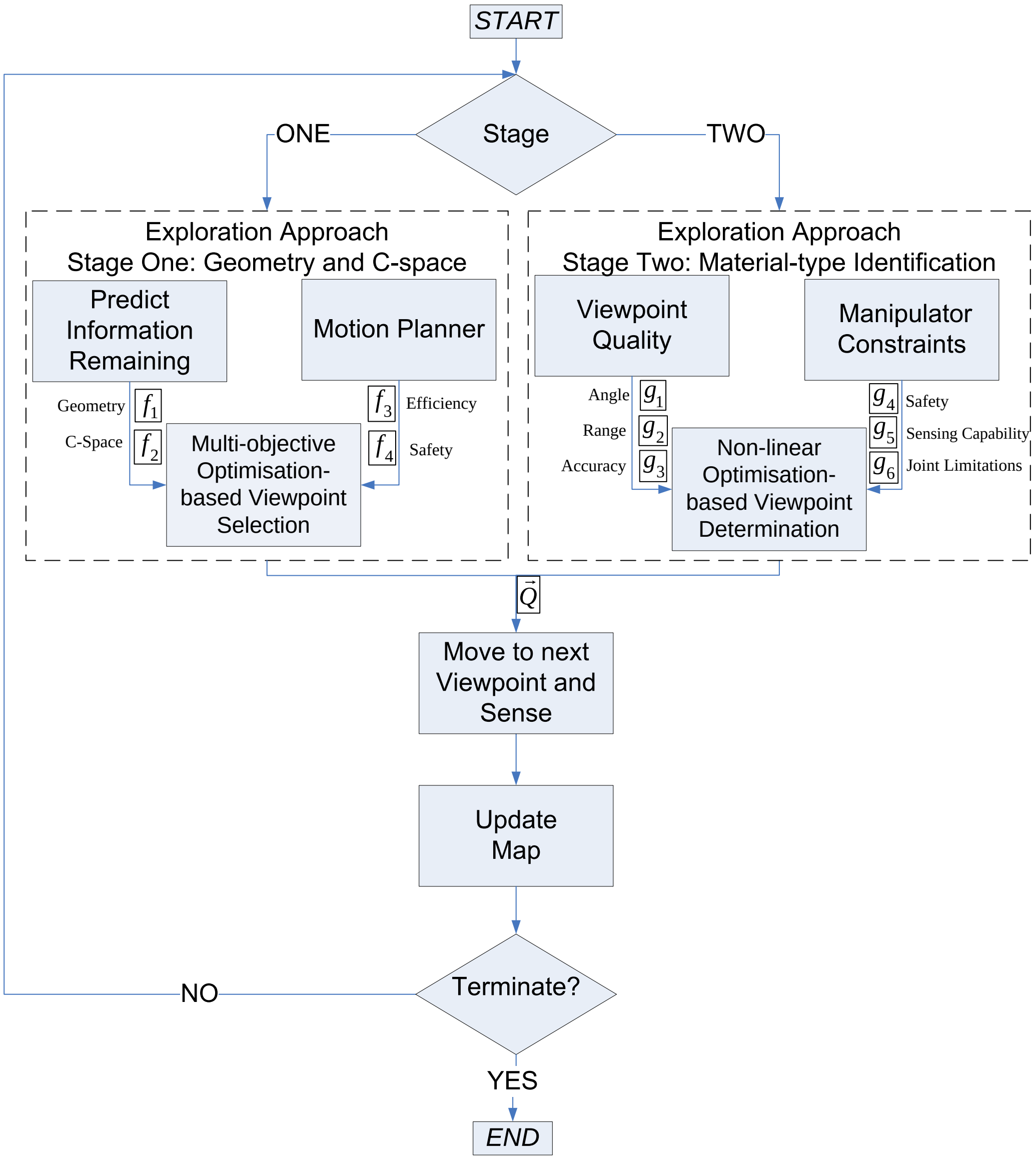


FIGURE 6.3: Flow Chart of the AXBAM System. *left*) Stage One exploration: Geometric Mapping; *right*) Stage Two exploration: Material-type identification.

The parameters used in the experiments are shown in Table 6.1. The initial geometric and C-space information remaining prior to viewpoint 0 for all experiments is denoted as $\mathbf{H}_1(\mathbf{X})_0$ and $\mathbf{H}_2(\mathbf{C})_0$. Prior to exploration there are no known surfaces so the material-type information, $\mathbf{H}_2(\mathbf{C})_0$, equals 0 and is not shown in the table. In the Stage Two parameters, 'LM' refers to the Levenberg-Marquardt algorithm's termination parameters for the maximum number of optimisation iterations and the gradient tolerance.

TABLE 6.1: Experiment Parameters

Stage One					
$\mathbf{H}_1(\mathbf{X})_0$	$\mathbf{H}_2(\mathbf{C})_0$	$\{c_{w1}, c_{w2}, c_{w3}\}$	n_Q	n_v	α
4.7×10^3	22	$\{0.6, 0.35, 0.05\}$	1571	41070	60°
Stage Two					
n_t	ϕ_{max}	$\{d_{min}, d_{max}\}$	ρ_{min}	$\{q_{i,min}, q_{i,max}\}$	LM
5	42°	$\{250, 800\}mm$	$15mm$	Appendix C	$\{50, 1e-4\}$
Mapping					
μ	n_{min}	$\tau_{\mathbf{q}}$	τ_d	τ_p	τ_θ
0.04	8	$\{8, 10, 14, 20, 20, 20\}^\circ$	$200mm$	$50mm$	5°

6.2 Evaluation Criteria

As exploration is conducted in two stages, the experimental results will be presented as such. The first section will show how the objectives in Stage One of exploration are met, and that the geometric surface map is produced as required. Stage Two then shows how the geometric map is used in exploration for surface material-type identification. The evaluation criteria for the results are defined for the two stages in the following two subsections.

6.2.1 Stage One: Geometric Mapping

The factors used to judge the success of the geometric mapping process are as follows:

1. The geometry of surfaces and the areas of freespace within an environment of interest must be discovered. The purpose of the first objective function, f_1, is to minimise the amount of geometric information remaining after sensing the environment from each viewpoint. The results from experiments (the values of f_1) will be used to evaluate the information remaining and efficiency of the Stage One exploration.
2. The second objective is to minimise the C-space information remaining so as to enable efficient manipulator movements. The manipulator's C-space information for the n_Q viewpoints will be used for evaluating the exploration approach.
3. A manipulator path to each of the viewpoints and the time taken to move the manipulator have a significant effect on exploration efficiency. The third objective function, f_3, aims to minimise the movement of the 6DOF manipulator from viewpoint to viewpoint and therefore is also used to evaluate the Stage One exploration.
4. The safety of manipulator movements is another vital consideration. Safe movements are those that can avoid any possible collisions of the manipulator with obstacles or with unexplored space. The risk of collision is presented as the fourth objective function, f_4. The smallest algebraic distance from the ellipsoid (which cover the manipulator links) to each unsafe point is determined. If an ellipsoidal bounding field around the manipulator at a viewpoint has its smallest algebraic distance less than a threshold, it means a collision will occur or an unexplored space will be intersected with.

The AXBAM system is therefore judged based on the values of the objective functions, computational efficiency and the quality of the output map.

6.2.2 Stage Two: Material-type Identification

Surface material-type information reduced by means of the LRC is achievable if the appropriate objectives and constraints are considered. The factors used to evaluate the success of viewpoints determined for surface material-type identification are as follows:

1. The laser scanning sensor's centre ray must meet the targeted object's surface within a certain angle of incidence, ϕ_{max}. Therefore, the value of the objective function g_1, will evaluate how well the sensor is orientated with respect to the surface.

2. The range to the SLD target must fall between the sensing range limitations, d_{min} and d_{max}. The objective function value, g_2, will be used to evaluate the adherence to the range limitation.

3. The third objective function, g_3, must be minimised to ensure the centre laser ray is close (i.e. within ρ_{min}) to the desired specific target's centre, and therefore the value of this function is also an evaluation criteria.

4. The safety of the manipulator is incorporated as the fourth objective function, g_4. Provided that the smallest algebraic distance is less than the threshold, the manipulator is safe and hence the viewpoint is valid.

5. The 3D Field Of View is significantly affected by the fifth joint. Therefore, the freedom of movement of this joint (greater than $q_{t,max}$) which allows the joint to tilt the sensor is evaluated using the fifth objective function, g_5.

6. A pose corresponding to a viewpoint requires that the manipulator's joints are constrained to within the physical angular limitations $q_{i,min}$ and $q_{i,max}$. The sixth objective function, g_6, will thus be used to evaluate whether or not the manipulator pose is valid.

7. The material-type of surfaces must be discovered. Subsequent to sensing at a viewpoint which has been determined, results from experiments (the value of $\mathbf{H}_3(\mathbf{M})$) will be used to evaluate the material-type information remaining.

The AXBAM system is therefore judged based on the values of the objective functions in Stage Two and the reduction of the material-type information remaining in the environment.

6.3 Experiment One: Exploration in a Simple Environment

The first experiment is conducted for a relatively simple case of a steel bridge structural environment which contains various obstacles made from different materials (Fig. 6.4). The aim is to explore and map the environment using the presented approach. As shown in Fig. 6.4, the environment consists of an I-beam channel running lengthways into the image, with two painted steel scaffold members on either side, a matt-finish painted steel plate on the left-hand I-beam web, blast hose, containment plastic connected to the right-hand web of the I-beam, and a timber channel roof above the robot.

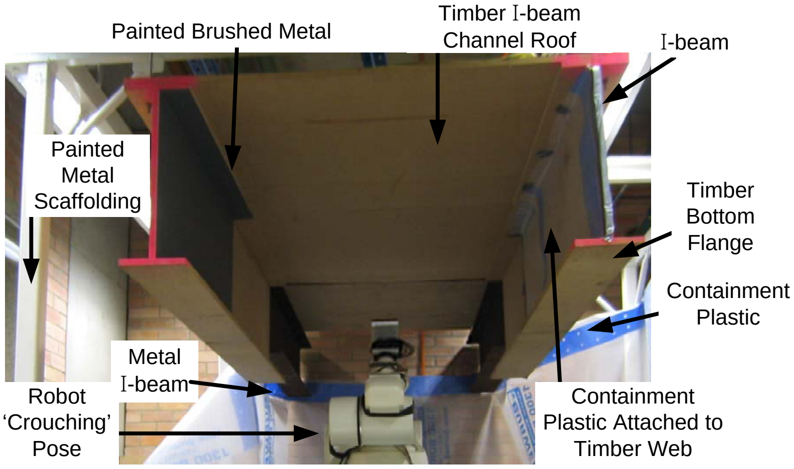


Figure 6.4: The environment to be explored in the first experiment.

In this experiment, after the initial scan (i.e. the viewpoint 0) shown in Fig. 6.4, AXBAM went through nine exploration viewpoints. Moving between viewpoints and the scan at each viewpoint takes approximately 95% of the time (2-3 minutes). The remaining 5% (5-10 seconds) of the time is required for all computations, including exploration prediction and updating the map. The following section shows the results of the two-stage exploration approach.

6.3.1 Stage One of Exploration: Geometric Mapping

Fig. 6.5 shows viewpoints 2 to 7 where sensing occurs. After viewpoint 0 (Fig. 6.4), the subsequent viewpoints gradually reduced the geometric and C-space information remaining in the environment using minimal manipulator movement. Viewpoint 1 is still in a crouched pose similar to viewpoint 0. This allows sensing of the regions immediately above the base of the manipulator.

Viewpoint 2 (Fig. 6.5*a*) directs the sensor and the manipulator on the right-hand side of the I-beam, while viewpoint 3 (Fig. 6.5*b*) is to the left-hand side. As shown in the C-space information remaining objective function, f_2, (Fig. 6.6*b*) these viewpoints are able to discover a great deal of information. In viewpoints 4 to 6 (Fig. 6.5*c*-*e*), because of the obstacles (marked in Fig. 6.4) above the manipulator (i.e. the roof and I-beams of the bridge structure) small manipulator movements are used during exploration.

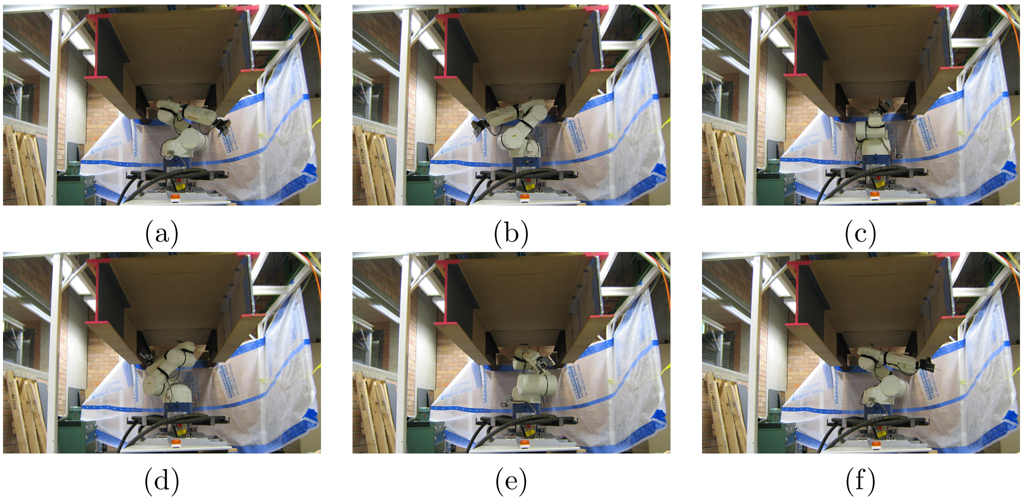
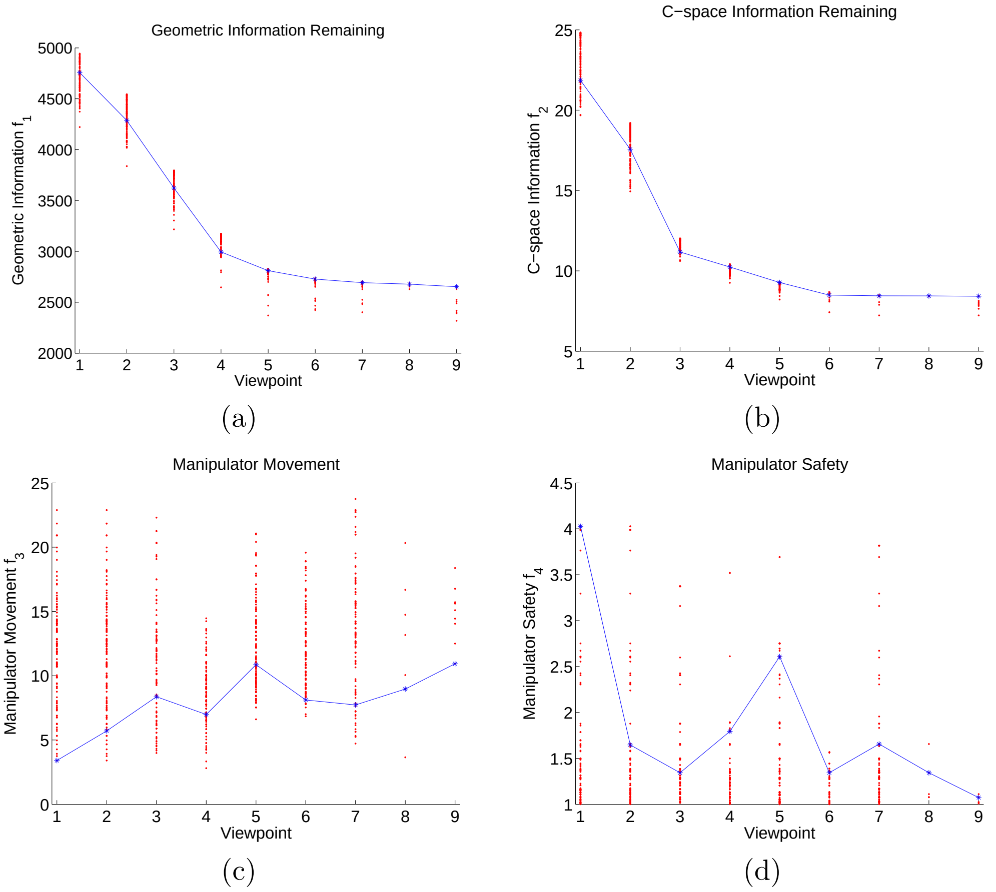

(a) (b) (c)
(d) (e) (f)

Figure 6.5: *a*-*f*)Viewpoints 2 to 7 for exploration.

Once the C-space information is reduced the manipulator is able to move to viewpoint 7 (Fig. 6.5*f*) and 8 to look at the underside of the flange of each I-beam. Fig. 6.6 presents the results of the four objective functions obtained in this experiment. The information remaining is reduced with the first three viewpoints contributing most significantly (Fig. 6.6*a* and Fig. 6.6*b*).

Fig. 6.6 shows that multiple viewpoint candidates exist at each iteration of Stage One. The objective function results (i.e. the viewpoint candidates) are shown as red dots ($\cdot$) for each of the nine viewpoints. The viewpoint chosen by the multi-objective optimisation in Stage One is shown as a blue star (*). It is clear how the geometric and C-space information remaining is reduced at each viewpoint (Fig. 6.6*a* and *b*). However, the information remaining reaches a level where it is predicted that it will not be significantly reduced. This is due to sensing occlusions caused by objects in the environment. There is still information within the pre-defined maximum workspace which the robot can neither reach nor sense. Therefore, once the information remaining is no longer being reduced, it is acceptable to terminate Stage One. Fig. 6.6*c* and Fig. 6.6*d* show how the viewpoints are also selected based upon the objective to minimise the manipulator movements, and the objective to maximise the manipulator safety, respectively.



(a) (b)

(c) (d)

FIGURE 6.6: Stage One objective functions: *a*) to *d*) for (f_1, f_2, f_3, f_4). Each viewpoint has up to 1000 possible candidates, (red $\cdot$) with one selected (blue *).

The output map is shown in Fig. 6.7. The number of points in the mesh map, n_p, is approximately 145000. The number of SLDs which are generated within the manipulator's workspace, n_{ss}, is 4088. Based on visual inspection compared with Fig. 6.4, the map is sufficiently complete and conforms to the shape of the environment. There is an map of the surfaces outside the working envelope which still must be considered (shown in red as a point cloud in Fig. 6.7).

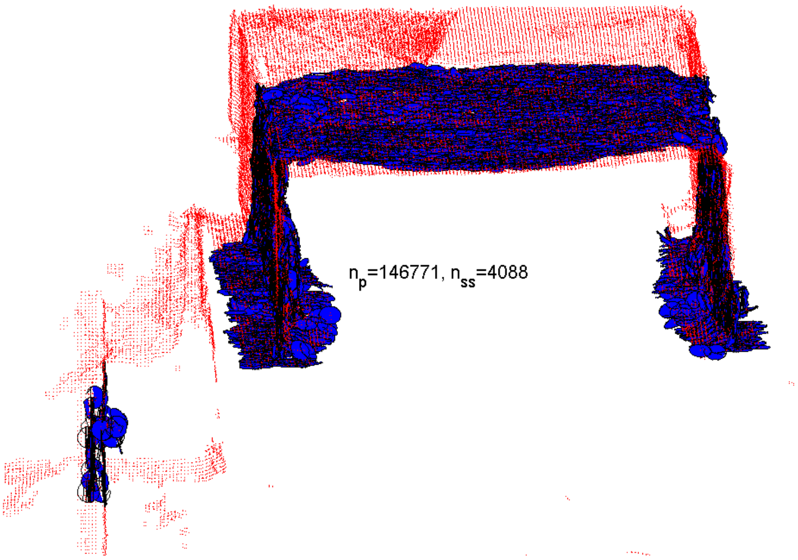
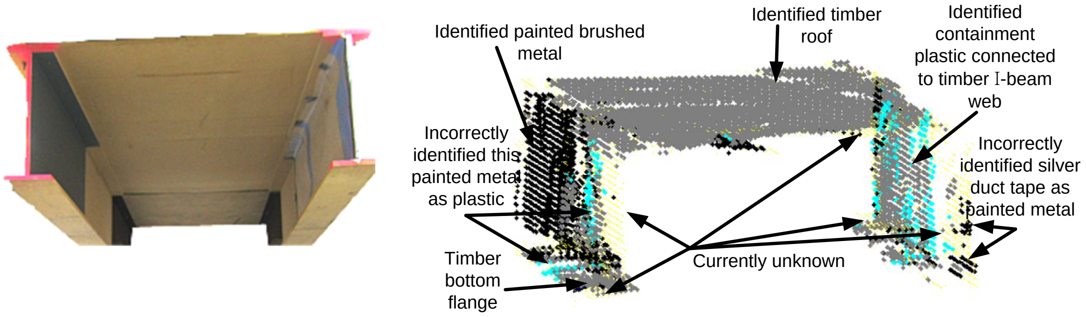


Figure 6.7: The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations.

Fig. 6.8 shows that the timber (in grey) on the roof has been identified (in black) in Stage One. Some areas of painted metal web (in black) on the left are also identified. The identified areas of the containment plastic on the right web are shown in cyan, while unknown material-types are shown in yellow. The plastic containment material, attached to the timber is only partially successful as it is translucent. The misidentification of this containment plastic is generally timber.



(a)

(b)

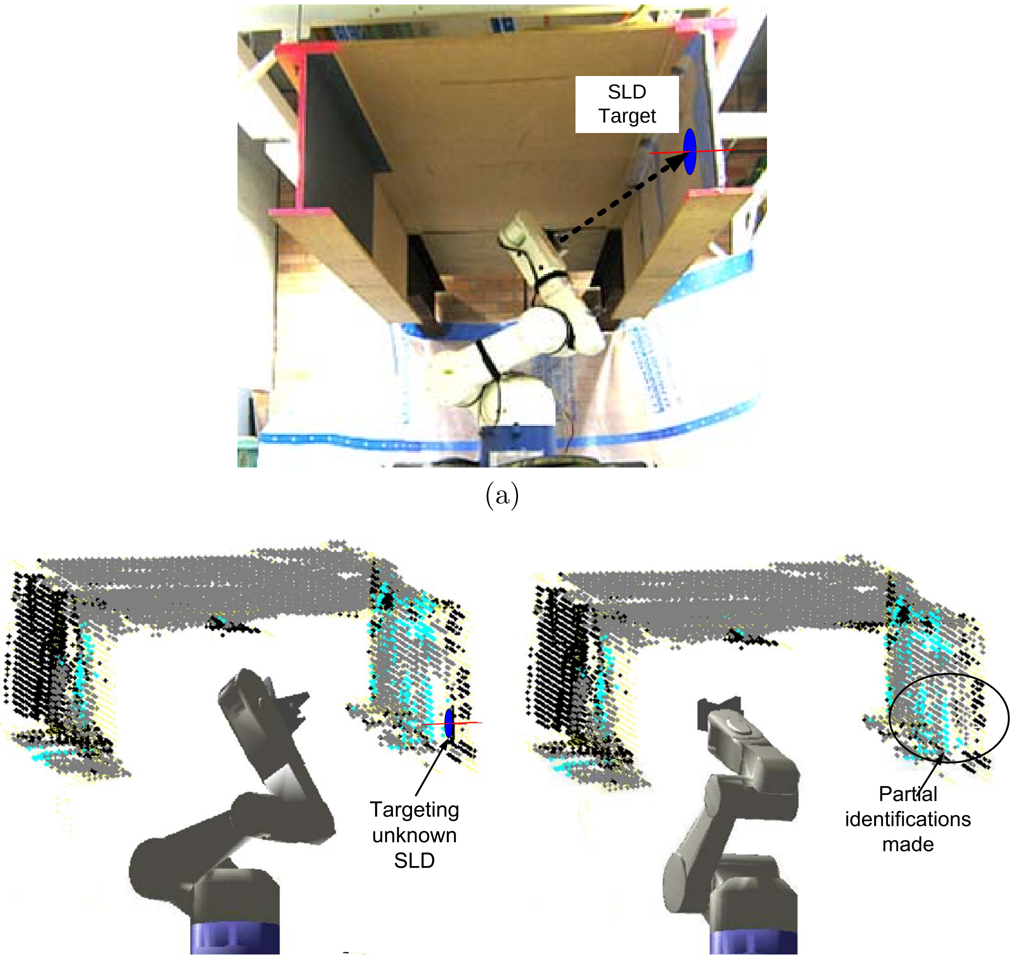
FIGURE 6.8: *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow.

6.3.2 Stage Two of Exploration: Material-type Identification

Stage Two begins once the geometric map of the environment is adequately built and SLDs are generated to represent the geometry of surfaces. During Stage One some incidental material-type information was collected (Fig. 6.8), however there is still material-type information remaining in the environment because the focus of Stage One exploration is geometric mapping. Thus, the sensor must be positioned carefully to inspect surfaces and collect the required information for material-type identification.

This section shows how the material-type information of the surfaces is collected by the Stage Two exploration. The Stage Two objective functions that are used to determine a viewpoint are shown to be minimised via the optimisation process. The material-type information remaining on each Scale-Like Disc is used to rank the SLDs, then the most desirable SLD target which must be identified is selected, as presented in Chapter 3. At each viewpoint, the manipulator pose is determined by optimising the objective functions using the Levenberg-Marquardt (LM) algorithm, then the material-type information of the map is updated based on the identification results.

Fig. 6.9*a* shows a viewpoint where the manipulator is placed at a pose so as to allow the right-hand I-beam web to be sensed. The targeted SLD is highlighted in Fig. 6.9*b*. The result is that partial identifications have been made as shown Fig. 6.9*c*. On some surfaces in the environment the tape holding the containment plastic to the I-beam on the right is incorrectly identified as painted metal because of the similarity of the surface profiles. Additionally, the containment plastic is difficult to identify because in some places the timber behind is visible. Therefore, the containment plastic surface is commonly misidentified as timber.



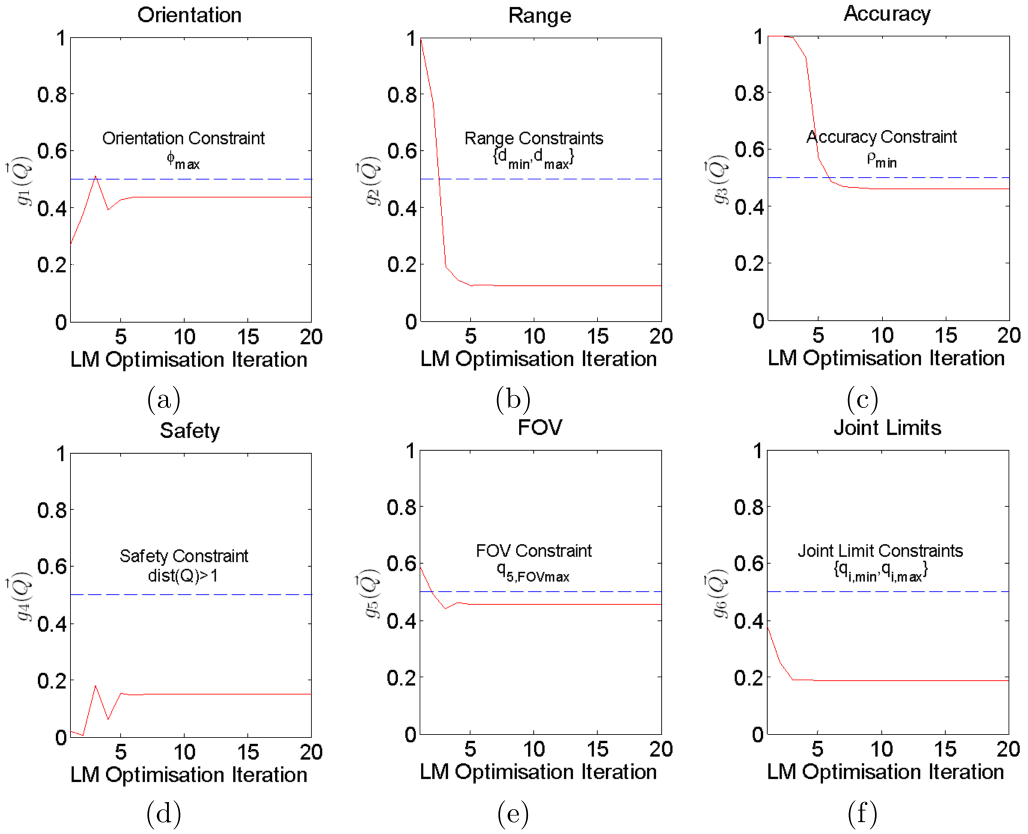
(a)

(b)

(c)

Figure 6.9: *a*) Photo of the robot in the second viewpoint of Stage Two; *b*) The second viewpoint indicating the targeted SLD which lies on the edge of the containment plastic and the silver tape which is misidentified as painted metal; *c*) The third viewpoint (i.e. after the second viewpoint identified the material-type of the surface).

Fig. 6.10 shows with a red line the values of each of the objective functions, $g_i(\vec{Q})$ for $i \in \{1, \ldots, 6\}$, as the LM optimisation is used to determine the pose in Fig. 6.9*b*. The optimisation is used to minimise the Stage Two objective functions and ensure all constraints (i.e. the LRC and manipulator constraints - the blue line) are met. For the first three LM optimisation iterations, $g_2(\vec{Q})$, $g_3(\vec{Q})$ and $g_5(\vec{Q})$ are not within their respective constraints. This means that the range between the scanner and the surface is not within the constraints d_{min} and d_{max}. It also means that the distance between the point of intersection of the scanner's central ray and the SLD is greater than ρ_{min}, and that joint 5 is not able to appropriately tilt in order to provide the required Field of View (FOV), $q_{t,max}$. As the optimisation continues, all objective functions except $g_3(\vec{Q})$ are minimised to be within their constraints. After seven LM optimisation iterations, all the constraints are met.

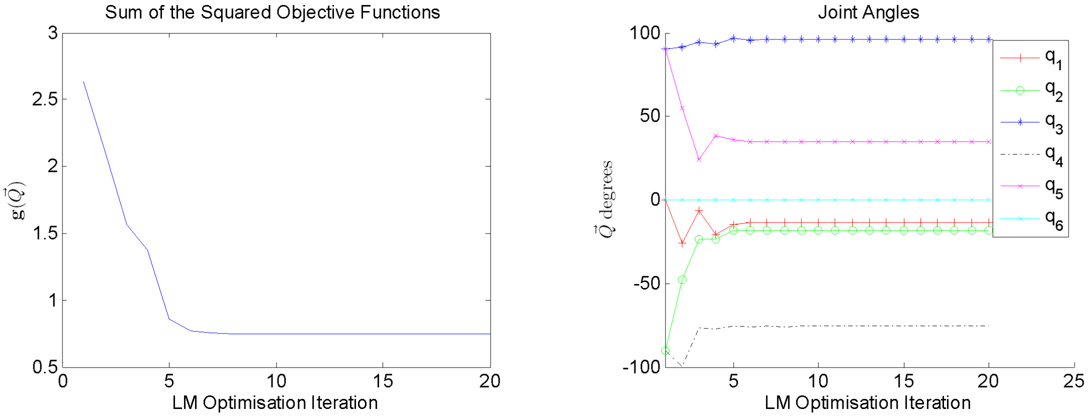


(a) (b) (c)

(d) (e) (f)

Figure 6.10: Stage Two viewpoint 2 constraints and objective functions: *a*) angle to surface normal $g_1(\vec{Q})$; *b*) distance to surface $g_2(\vec{Q})$; *c*) accuracy $g_3(\vec{Q})$; *d*) manipulator safety $g_4(\vec{Q})$; *e*) 3D field of view $g_5(\vec{Q})$; *f*) manipulator's joint limits $g_6(\vec{Q})$.

Fig. 6.11a shows the sum of the squared objective functions, $\mathbf{g}(\vec{Q})$, which is actually being minimised. After the 20 iterations, the LM optimisation is terminated due to the improvement gradient reaching the specified tolerance level (10^{-4}). This results in a single viewpoint being determined. The optimisation in this case takes $100ms$. The change in the manipulator joints, $\vec{Q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T$, over the course of the optimisation is shown in Fig. 6.11b.



(a)

(b)

FIGURE 6.11: a) The sum of the squared objective functions, $\mathbf{g}(\vec{Q})$, for viewpoint 2. b) The manipulator joints over the 20 LM optimisation iterations so as to determine the second material-type identification viewpoint in Stage Two.

Fig. 6.12 shows how the material-type information remaining, $\mathbf{H}_3(\mathbf{M})$, is reduced as Stage Two of exploration occurs at nine viewpoints. At the second viewpoint, an additional surface is discovered on the I-beam flange, requiring its material-type to be identified. Although the material-type information remaining is decreased on the target SLD, it is increased due to the new surface discovered. The result of which is a net increase in the material-type information remaining on all surfaces in the environment. The process of selecting a SLD and determining a pose so that the manipulator can place the sensor at an appropriate viewpoint continues through nine viewpoints. After nine viewpoints, Stage Two is terminated when a manipulator pose cannot be determined to sense any SLD with significant material-type information remaining.

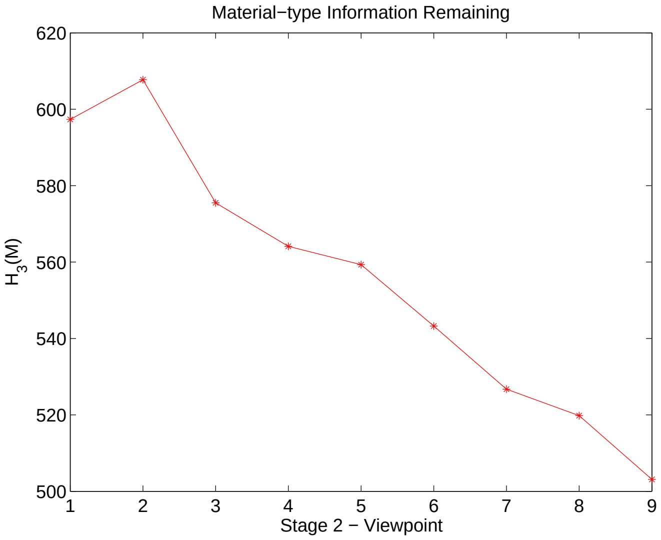


FIGURE 6.12: Material-type information remaining after each viewpoint, $\mathbf{H}_3(\mathbf{M})$.

6.4 Experiment Two: Exploration in Complex Environment

The second experiment is conducted in a more difficult version of the steel bridge environment due to an additional crossbeam directly behind the manipulator. As shown in Fig. 6.13, the environment consists of the I-beam channel, two painted steel plates on the left and right-hand I-beam web, containment plastic at the back and sides, a timber roof, and a timber crossbeam. Once again the initial scan is from viewpoint 0. AXBAM completed eleven exploration viewpoints to generate a complete geometric map.

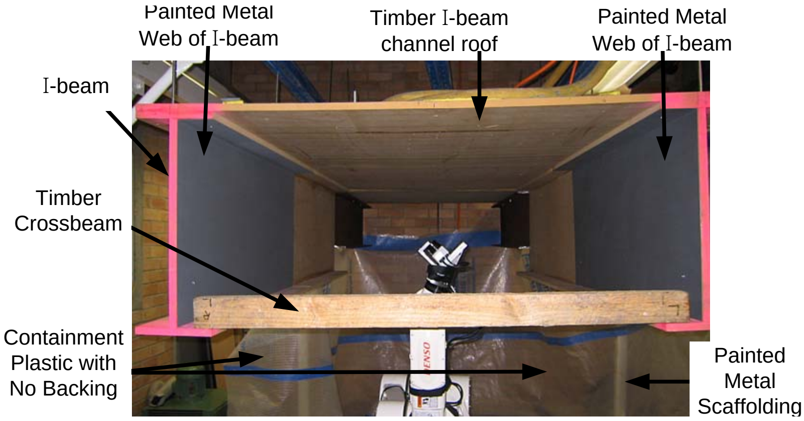


FIGURE 6.13: The environment for the second experiment.

6.4.1 Stage One of Exploration: Geometric Mapping

Fig. 6.14 shows the first six viewpoints where sensing occurred. Fig. 6.15 presents the results of the objective functions. From this, it is clear that in terms of the information remaining (Fig. 6.15*a* and *b*), the first six viewpoints reduce both the C-space and geometric information more significantly than the remaining five viewpoints in Stage One exploration.

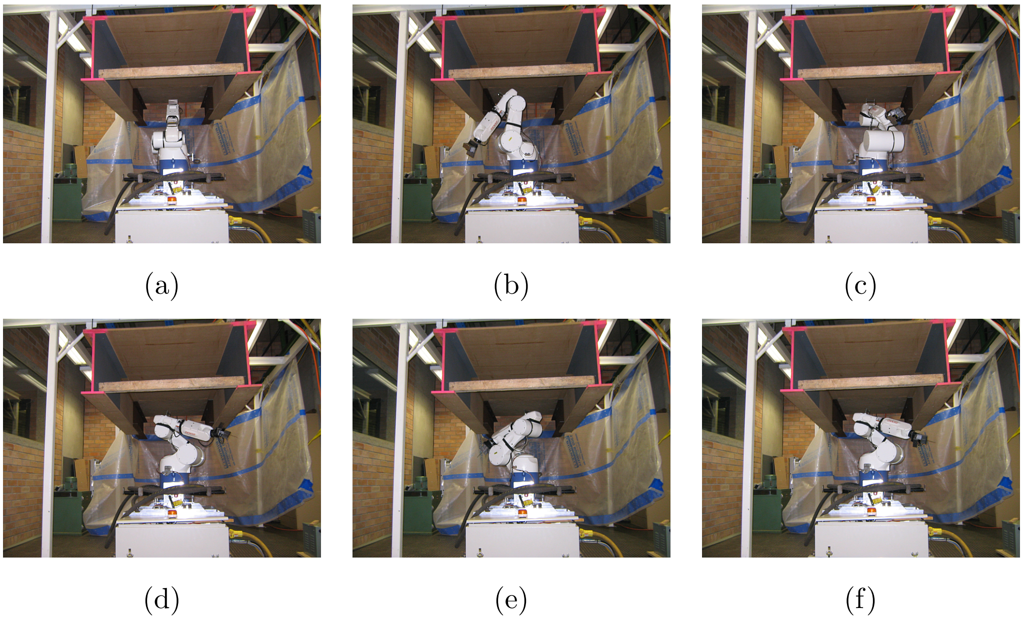
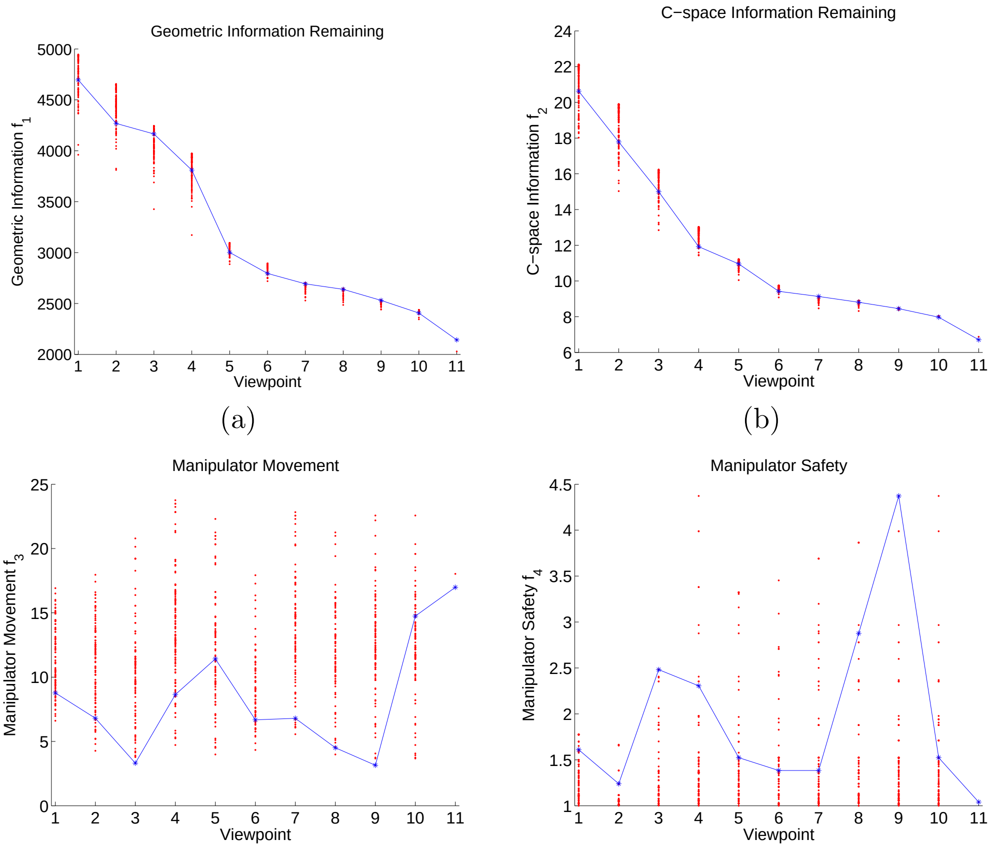

(a) (b) (c)
(d) (e) (f)

FIGURE 6.14: The first six viewpoints used for Stage One exploration.

In viewpoints 2, 3, 4 and 5 (Fig. 6.14*b, c, d* and *e*) the viewpoints collect the information surrounding the base (i.e. left, right back, right, left front). Fig. 6.15*a* and Fig. 6.15*b* show that both the geometric information and C-space information remaining are reduced as a result of these viewpoints. The reason for the different rates of reduction is because of the different positioning of occluding obstacles. Viewpoint 6 (Fig. 6.14*f*) is predicted to significantly reduce the freespace, however instead of sensing freespace, the side wall of the bridge is discovered. Therefore, there is only a small reduction in the geometric information remaining (Fig. 6.15*a*).

Fig. 6.15 shows the objective function results in red dots $(\cdot)$, with the viewpoint selected by the multi-objective optimisation as blue stars(*). Once again, as well as being selected based upon the reduction in information remaining, the viewpoints are selected by minimising the manipulator's movements (Fig. 6.15*c*) and maximising the safety of the manipulator (Fig. 6.15*d*). At viewpoint 8 and 9, the f_1 and f_2 are similar for a large group of $\vec{Q}$. The selection of these viewpoints is thus based upon the rank of the viewpoint in terms of manipulator movement, f_3, and manipulator safety, f_4.



(a) (b)

(c) (d)

FIGURE 6.15: Viewpoints with all contending possibilities, (red $\cdot$) and the viewpoint selected (blue *). *a*) to *d*) The Stage One objective functions, 1 to 4.

Experiment two requires more viewpoints to terminate because of the additional obstacle (timber crossbeam) which reduces the number of valid viewpoints to select from.

The output map is shown in Fig. 6.16. The number of points in the mesh map, n_p, is slightly more than 150000. The number of SLDs which are generated within the manipulator's workspace, n_{ss}, is larger than in the first experiment, with 4367 SLDs created. Based upon visual inspection, the map is not as complete as the first experiment, particularly on the roof. There is also an erroneous surface created directly above the timber crossbeam due to spurious sensing data. In general however, the SLDs conform to the shape of bridge structure.

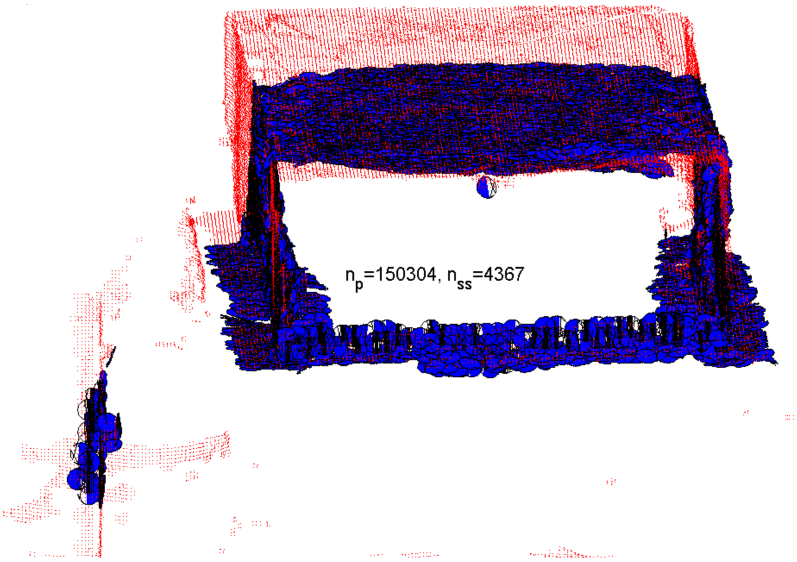
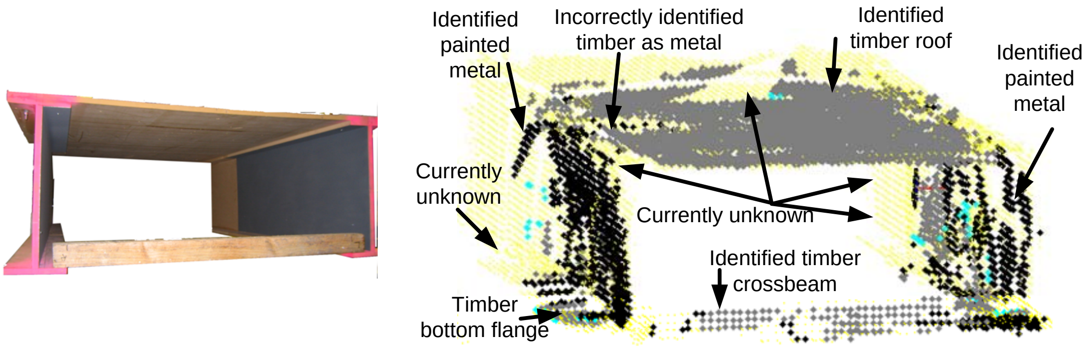


FIGURE 6.16: The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations.

Fig. 6.17 shows the material-type identification state subsequent to Stage One. Most areas of the timber on the roof are already identified, particularly directly above the manipulator, and a significant proportion of the painted metal on the left and right web of the I-beam have been identified. Due to the increased difficulty of the environment, less material-type identifications have been possible in this experiment than in experiment one. Once again the timber is shown in grey, the painted metal in black, the containment plastic in cyan and the unknown material-types are shown in yellow.



(a)

(b)

FIGURE 6.17: *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow.

6.4.2 Stage Two of Exploration: Material-type Identification

Stage Two of exploration begins once the geometric map of the environment is adequately determined. In Stage One, a small percentage of the material-type information has been gathered. However, there is still a significant amount of material-type information remaining in the environment, which requires Stage Two of the exploration approach.

The information remaining on SLDs is determined, and then a SLD target is selected for sensing. Stage Two objective functions are minimised by means of the LM optimisation process so that an appropriate manipulator pose is determined. Fig. 6.18 shows a sequence of three viewpoints starting from the 12th Stage Two viewpoint. These figures allow the material-type identification to be visualised.

The focus of the first eleven viewpoints (not shown) has been the identification of the material-type of surfaces around the fringes of the roof and the web on either side. There are already a significant number of surfaces which have been identified since Fig. 6.17. Fig. 6.18*a* shows the 12th viewpoint and the SLD which is on the underside of the timber flange of the I-beam. Fig. 6.18*b* shows the actual manipulator sensing from the determined viewpoint. Fig. 6.18*c* and *d*, show how the material-type is partially identified on the underside of the left-hand flange due to the 12th viewpoint. It also shows the SLD target and the 13th viewpoint which is determined by LM optimisation as shown in Fig. 6.19. Fig. 6.18*e* shows the results of the 14th viewpoint. Fig. 6.21 shows that by sensing at viewpoints 12, 13 and 14, there is a reduction in the material-type information remaining.

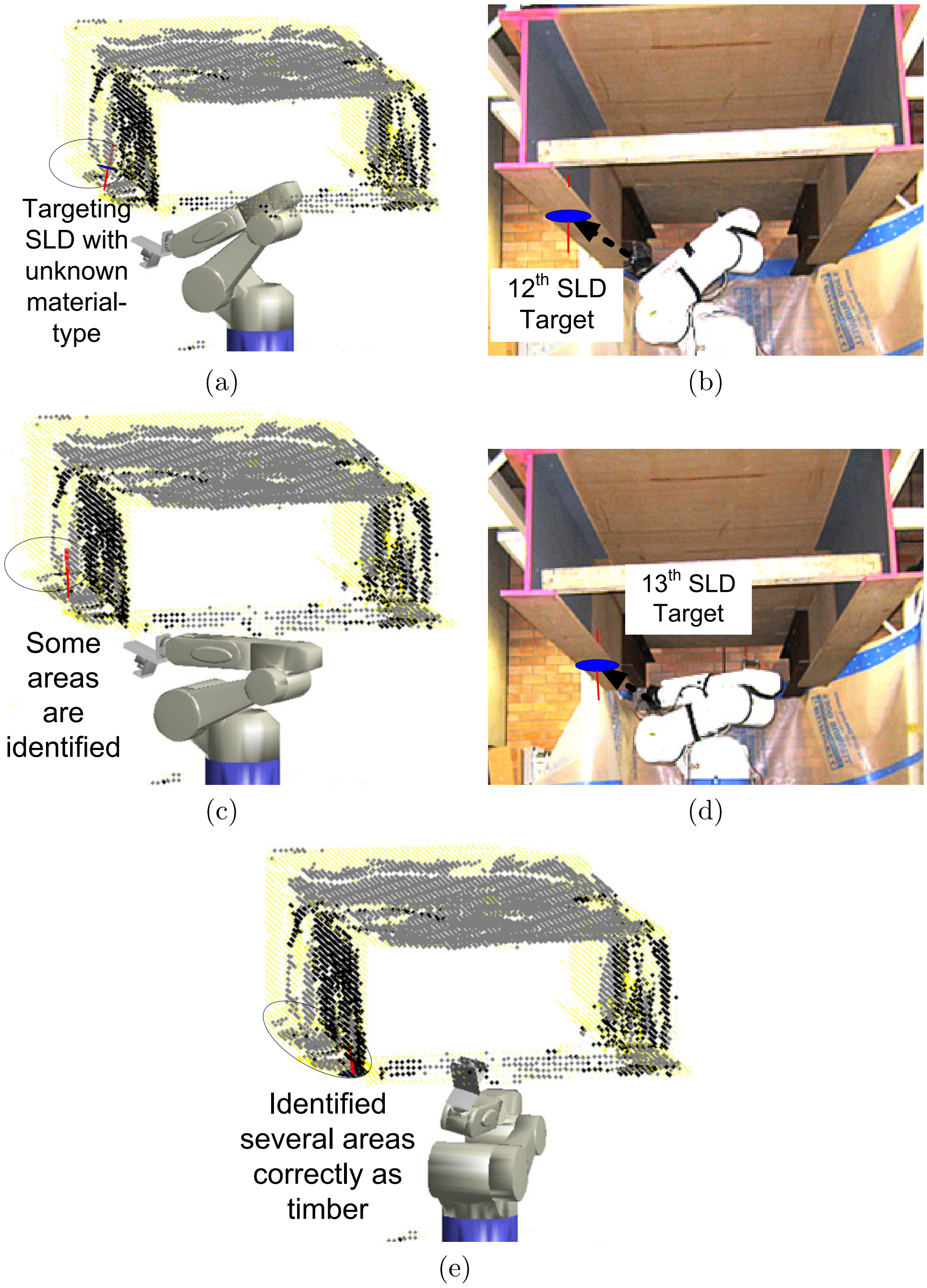
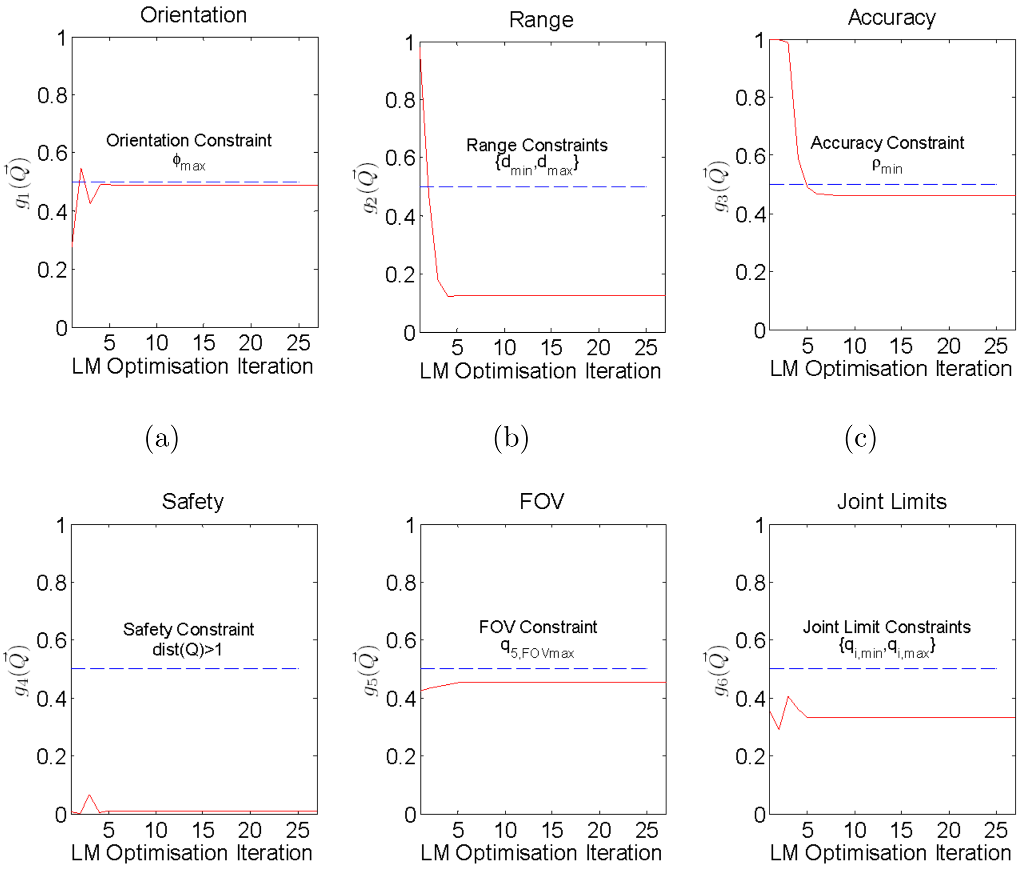


FIGURE 6.18: *a*) and *b*) Stage Two viewpoint 12; *c*) and *d*) Viewpoint 13; and *e*) Viewpoint 14 with material-type maps.

Fig. 6.19 shows the optimisation to determine the 13th viewpoint in Fig. 6.18*c*. Fig. 6.19 shows in red each of the objective functions, $g_i(\vec{Q})$ for $i \in \{1, \ldots, 6\}$. For the first several LM optimisation iterations, $g_2(\vec{Q})$ and $g_3(\vec{Q})$ are not within the desired constraints (blue lines). In this case, the range from the sensor to the SLD is greater than d_{max}, and the distance between the point of intersection of the scanner's central ray and the SLD is greater than the accuracy constraint, ρ_{min}. After the LM optimisation completes six iterations, all objective functions including $g_3(\vec{Q})$ are within their constraints. This is similar to the previous experiment, indicating that the third constraint (to ensure the central ray is within $\rho_{min} = 15mm$ of the desired target SLD centre) can be difficult to satisfy.

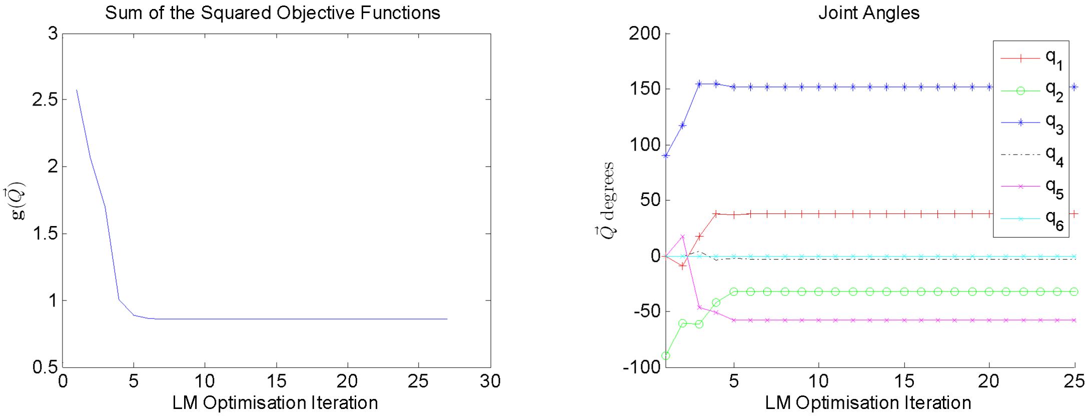


(a) (b) (c)

(d) (e) (f)

FIGURE 6.19: Determining the viewpoint 13 in Stage Two of exploration including the constraints and objective functions: *a*) angle to surface normal $g_1(\vec{Q})$; *b*) distance to surface $g_2(\vec{Q})$; *c*) accuracy $g_3(\vec{Q})$; *d*) manipulator safety $g_4(\vec{Q})$; *e*) 3D field of view $g_5(\vec{Q})$; *f*) manipulator's joint limits $g_6(\vec{Q})$.

Fig. 6.20a shows the sum of the squared objective functions, $\mathbf{g}(\vec{Q})$ which is actually being minimised. The LM optimisation is terminated after 27 iterations and the 13th viewpoint is determined. Fig. 6.20b shows how the manipulator joints, $\vec{Q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T$, change from their initial values over the course of the optimisation.



(a) (b)

Figure 6.20: a) The sum of the squared objective functions, $\mathbf{g}(\vec{Q})$ for each viewpoint in Stage Two; b) The manipulator joints over 27 LM optimisation iterations so as to determine material-type identification viewpoint 13 in Stage Two.

Fig. 6.21 shows for the second experiment how the material-type information remaining, $\mathbf{H}_3(\mathbf{M})$, is reduced as Stage Two of exploration occurs at 15 viewpoints. At viewpoint 9 an additional surface is discovered on the underside of the timber crossbeam. This additional surface is large and thus the material-type information remaining on all known surfaces is increased slightly (Fig. 6.21). The process of selecting a SLD and determining a pose so that the manipulator can place the sensor at an appropriate viewpoint continues through 15 viewpoints. The process is terminated after 15 viewpoints when a manipulator pose could not be determined to sense a SLD with significant material-type information remaining.

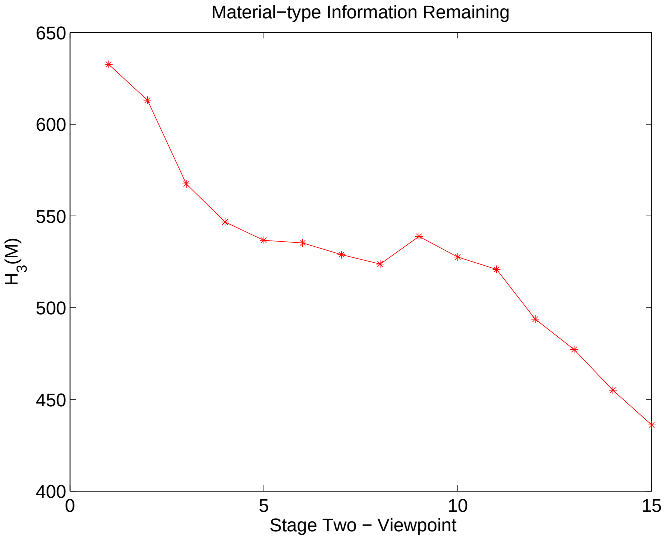


FIGURE 6.21: Material-type information remaining after each Stage Two exploration viewpoint, $\mathbf{H}_3(\mathbf{M})$.

6.5 Experiment Three: Exploration in a Second Complex Environment

The third experiment is conducted in a more challenging environment (Fig. 6.22) than experiments one and two. The environment contains more obstacles, so there are fewer valid viewpoints.

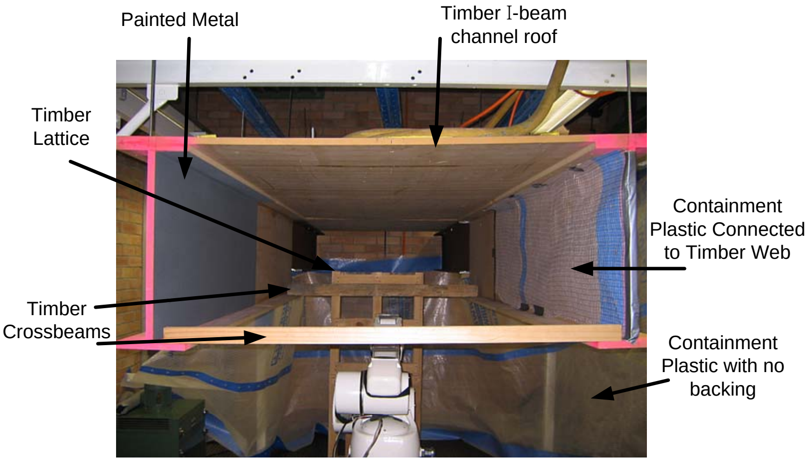


FIGURE 6.22: The environment in the third experiment.

The bridge structure environment for this experiment is an amalgamation of the environments in experiments one and two, with additional obstacles. A timber crossbeam is placed both in front and behind the manipulator, and a timber lattice is connected to the back crossbeam. Therefore, there are obstacles in front, behind to either side and above the manipulator. This results in a difficult environment to explore and map.

6.5.1 Stage One of Exploration: Geometric Mapping

In this experiment, eleven viewpoints are used for exploration, which is equal to the most used in the three experiments presented. The first six viewpoints are shown in Fig. 6.23. These reduce the geometric and C-space information more significantly than the final five viewpoints (Fig. 6.24*a* and *b*). Fig. 6.24 shows the objective functions for Stage One of exploration. The geometric and C-space information remaining is reduced at every viewpoint. Experiments one, two and three can be compared after the ninth viewpoint. The C-space information remaining in experiment three is reduced further than in the previous two experiments. The reason is that there are so many obstacles in the environment which cause collisions that there isn't a significant number of unknown voxels (i.e. geometric space) which affect the freedom of movement of the manipulator. The geometric information remaining after Stage One is similar in both experiments two and three.

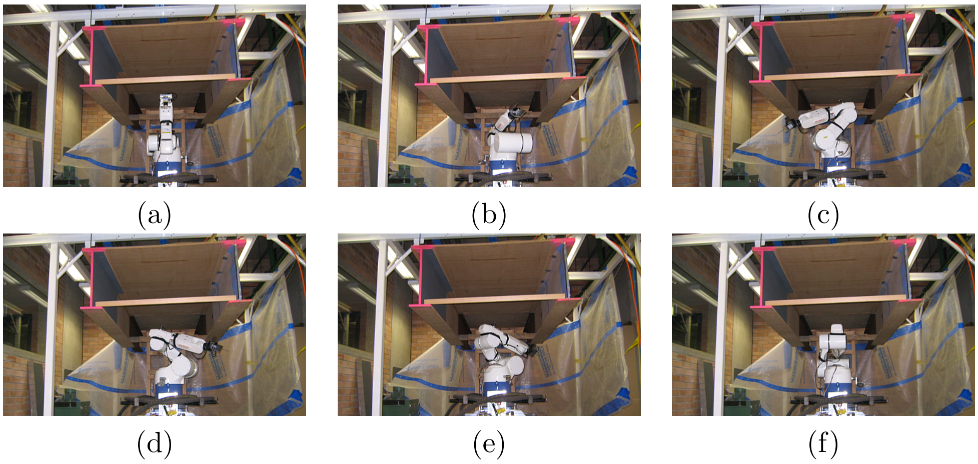
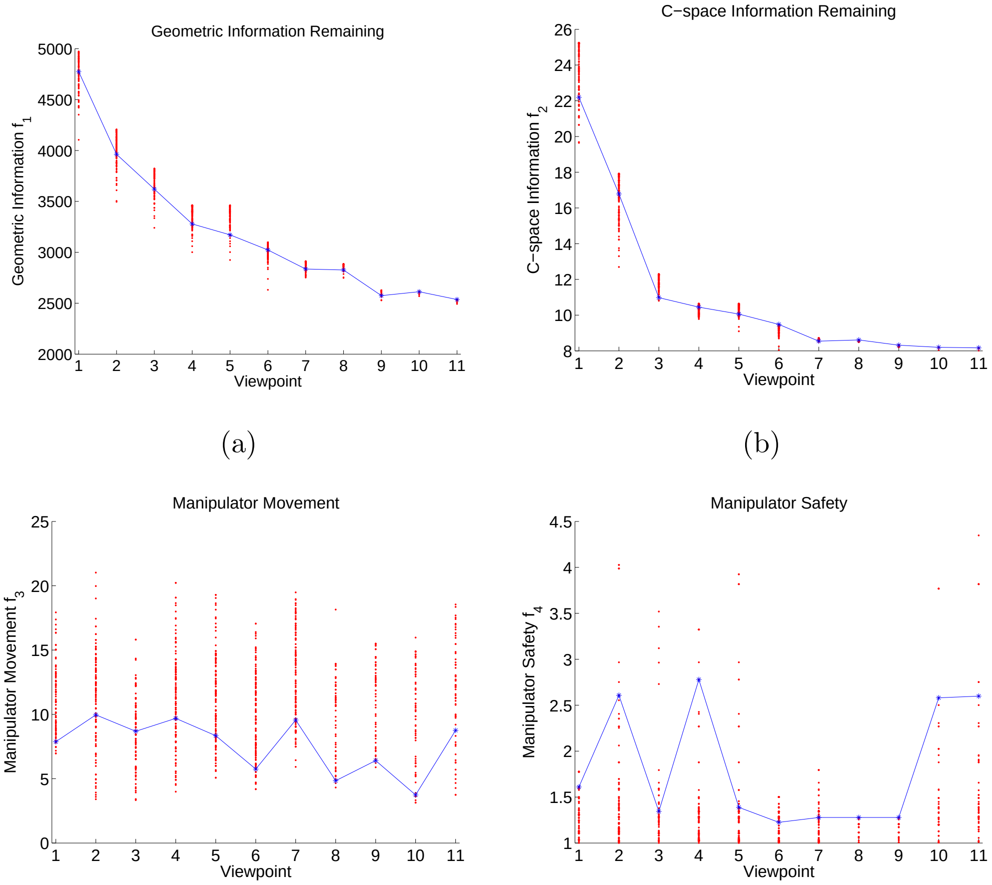

(a) (b) (c)
(d) (e) (f)

FIGURE 6.23: Exploration viewpoints 1 to 6. In this experiment eleven viewpoints are required, although after six viewpoints, the information reduction is not significant.



(a) (b)

(c) (d)

FIGURE 6.24: Viewpoints with all contending possibilities, (red ·) and the viewpoint selected (blue *). *a*) to *d*) The Stage One objective functions 1 to 4.

The output map is shown in Fig. 6.25. The size of the point cloud, n_p, is approximately 155000 which is 3% more than the second experiment and approximately 7% more than the first experiment. The number of SLDs which are generated within the manipulator's workspace, n_{ss}, is slightly larger than in the first experiment, with 4449 SLDs generated. This is expected due to the additional obstacles in the environment. Based upon visual inspection, the map is relatively complete with the two crossbeams and the lattice clearly visible.

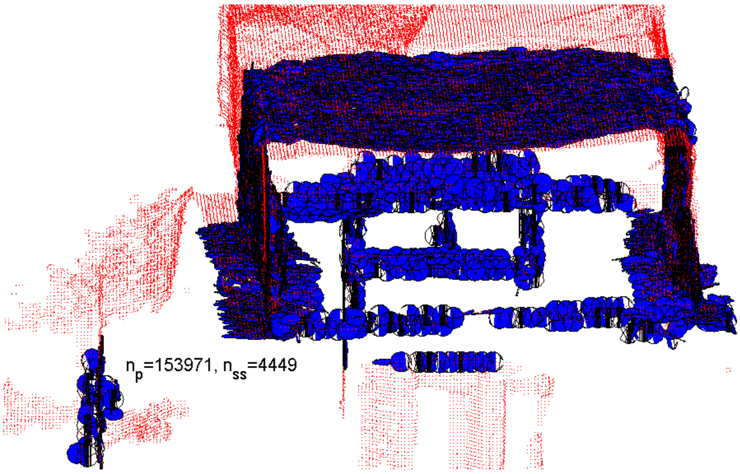
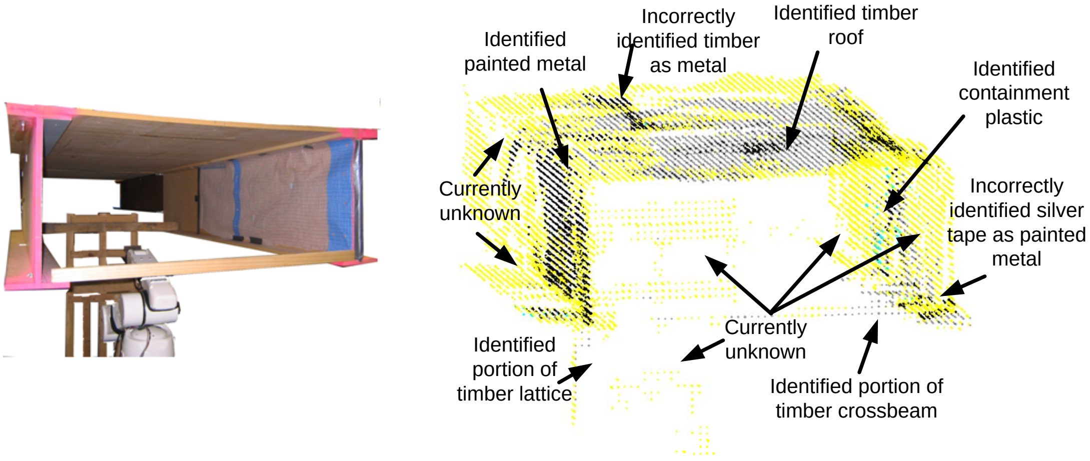


FIGURE 6.25: The output map of mesh vertices and the SLDs which are within the manipulator's work envelope for maintenance operations.

Fig. 6.26 shows the material-type identification state subsequent to Stage One. The timber on the roof has been identified directly above but not to the front or back, while the painted metal on the left web of the I-beam has also been identified in some areas. Once again the timber is shown in grey, the painted metal in black, the containment plastic in cyan, and the unknown material-types are in yellow.



(a)

(b)

FIGURE 6.26: *a*) The target environment; *b*) The material-type identification state after Stage One. Materials: timber = grey, metal = black, containment plastic = cyan, unknown = yellow

6.5.2 Stage Two of Exploration: Material-type Identification

As shown in Fig. 6.26, Stage One of exploration has allowed some incidental material-type identifications. However, there is still material-type information remaining in the environment which requires that the sensor be positioned and orientated so that the surfaces can be identified. The material-type information remaining on the surfaces is reduced as the Stage Two objective functions are optimised, so a manipulator pose is determined. The material-type information remaining after sensing and processing the sensor data with the LRC, is shown for each Stage Two viewpoint in Fig. 6.27.

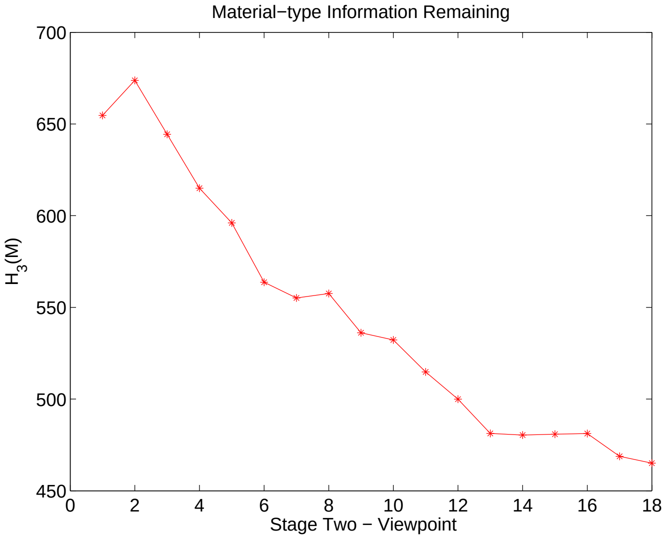
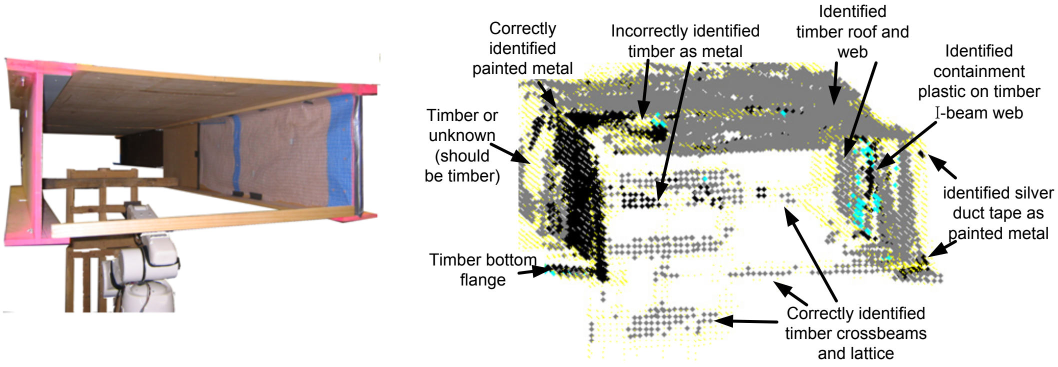


FIGURE 6.27: Material-type information remaining after each Stage Two exploration viewpoint, $\mathbf{H}_3(\mathbf{M})$.

The information is reduced more slowly than in experiment one and two. At viewpoint 2 of Stage Two and again on viewpoints 8 and 16, additional surfaces were discovered on the lattice. Due to the complex shape of the lattice, more viewpoints are required to sense all the surfaces. When the new surfaces are discovered after viewpoints 2, 8, and 16, the surface material-type information remaining in the environment increases. This is despite the information reduction achieved by sensing each SLD target from the determined viewpoint. In this experiment, Stage Two was terminated manually after 18 iterations. Fig. 6.28 shows the surface material-type identified map after the 18 Stage Two viewpoints. The material-type (timber) of the crossbeams and the lattice is identified when compared with the map prior to Stage One (Fig. 6.26). Also, a significant amount of the roof is identified correctly as timber. There is a small area on the roof to the left which is incorrectly identified as metal instead of timber. The identification of the containment plastic is generally poor due to it being translucent and mounted on the timber I-beam. In many cases the timber behind is identified instead.



(a)

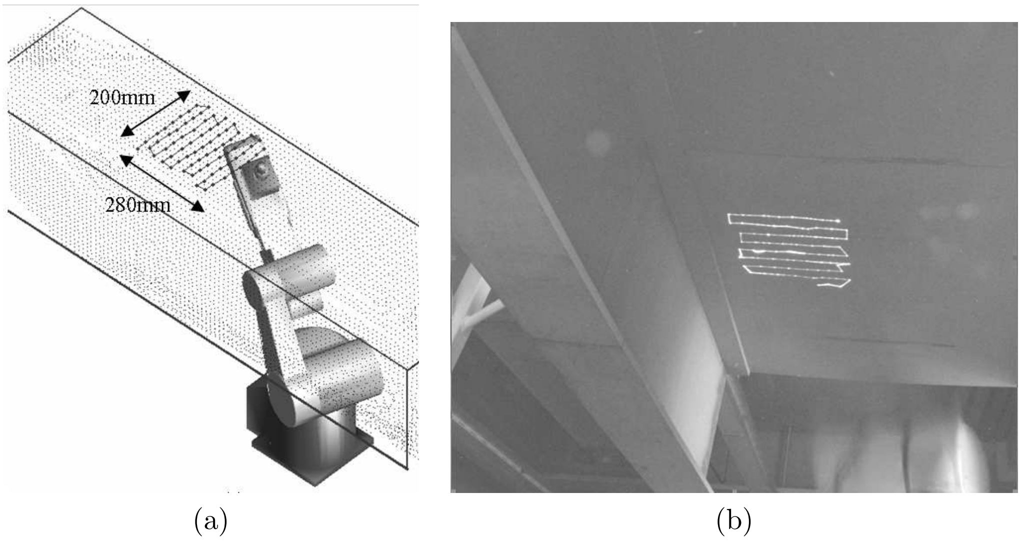
(b)

FIGURE 6.28: *a*) The target environment; *b*) Material-type map after the 18th viewpoint.

6.6 Map Utilisation Discussion

Two maintenance operation demonstrations are presented to show how the generated geometric maps can be utilised. Both demonstrations are based on the segmented SLD-based map built by AXBAM in the experiments for exploration and mapping.

Fig. 6.29 shows the first demonstration, which simulates the grit-blasting maintenance operation. The 6DOF Denso VM-6083 robot manipulator equipped with a hose and blasting nozzle is fitted with a low-power laser pointer. The laser pointer is placed inside the blast nozzle so that a grit-blasting pattern can be simulated. The centres of the SLDs are shown in Fig. 6.29*a* as dark points. A grit-blasting pose-selection and sequencing algorithm [25] (detailed in Appendix A) is used to determine manipulator poses and manipulator trajectories which allow for smooth movement over the generated map.



(a) (b)

FIGURE 6.29: *a*) Grit-blasting plan created from a SLD-based segmented map. *b*) Actual trajectory of the grit-blast spot over the roof of the bridge environment using a laser pointer and a long exposure camera.

The manipulator trajectory plan is executed for the segment on the timber roof. The trajectory path is recorded by a still image capture using a long exposure camera. As the laser pointer is on, the long exposure camera is able to capture the sequence of movements over the surface of the structure. Fig. 6.29*b* shows how the simulated blast spot is directed at a sequence of SLDs in the segment. The demonstration shows how the simulated grit-blasting of the map (Fig. 6.29*a*) is similar to the simulated grit-blasting of the bridge structure (Fig. 6.29*b*).

(a) (b)

FIGURE 6.30: A manipulator conducting grit-blasting of a metal I-beam surface based on a map autonomously generated by AXBAM.

Fig. 6.30 shows a second maintenance operation demonstration that utilises a map built, is conducted on a metal I-beam to the left of the manipulator. In this demonstration, live grit-blasting of the metal I-beam web is performed using the autonomous prototype steel bridge maintenance system (Appendix D). A significant potion of the paint and rust is removed successfully from the I-beam web. Issues with continuous air and grit supply led to some areas of patchy completion in Fig. 6.30*b*. A thorough investigation of the prototype grit-blasting maintenance system is outside the scope of this research and is left for future work. This demonstration shows that a geometric map generated by AXBAM through the two-stage exploration approach presented can be utilised in some maintenance operations.

This chapter has presented the results of three exploration and mapping experiments and two brief maintenance operation demonstrations. The two-stage exploration and mapping approach, algorithms, techniques and a deployable system (AXBAM), have been evaluated using a replica of a steel bridge structure. The geometry and material-type of surfaces of 3D complex structural environments have been acquired, whilst discovering and avoiding obstacles within the environment. The algorithms and approaches presented in this research can thus be used to acquire and represent the geometry and material-type properties of surfaces of complex real-world structural environments, such as steel bridges.

Chapter 7

Conclusions

This thesis has presented a two-stage exploration approach to robot manipulator-based exploration and mapping of complex 3D environments. This approach enables an industrial robot manipulator system placed in a complex environment to autonomously acquire information about the environment. An information-theory based modelling technique has been presented to measure the information of an environment. The relevant information pertains to the geometry and material-types of a three-dimensional (3D) environment and the Configuration space (C-space) of a manipulator. The information remaining measurements are used to select sensing viewpoints at which to position and orientate a laser range finding sensor. The exploration approach presented is divided up into two stages, which were completed sequentially. A surface representation and map segmentation technique is also presented to represent a 3D environment using scale-like discs (SLDs) which can be clustered to allow task-specific map segmentation. In exploration and mapping, safe and efficient robot manipulator movements are considered when positioning the sensor in a partially known environment. The exploration and mapping approach consists of several novel algorithms and techniques which have been verified via simulations, then implemented into a system so as to perform experiments in a steel bridge maintenance application scenario.

7.1 Summary of Contributions

7.1.1 A Novel Two-stage Manipulator-based Exploration Approach

The two-stage approach is shown to be a novel way to solve the manipulator-based exploration problem. This approach first explores and builds the surface geometry and the manipulator's C-space map, and then identifies the surfaces' material-types. Each stage consists of objectives which focus on the reduction of the information remaining about the geometry, C-space and surface material-type, and the maximisation of safety and efficiency of a manipulator's trajectories. The geometry and C-space information is effectively reduced by optimising the objective functions. The problem is modelled as a multi-objective optimisation problem, such that a set of contending viewpoints can be ranked and the best viewpoint chosen from the set. Thus, the exploration approach enables an appropriate viewpoint to be selected based upon the relevant objectives, so as to comprehensively explore an environment by means of a manipulator equipped with a laser range scanning sensor.

7.1.2 Pose Selection Including Sensing Constraints

A manipulator pose selection optimisation approach is presented based upon the constraints of a Laser Range Classifier (LRC) system. The surface geometry (encapsulated in the SLD representation) is used to formulate objective functions which address the material-type identification constraints and the limitations and safety of a manipulator. This pose selection problem is solved via a non-linear optimisation approach. The outcome is a novel formulation of the pose selection problem, which incorporates sensing constraints and manipulator parameters.

7.1.3 Information Theory-based Measurement Models

An information theory-based modelling technique was developed to measure the remaining information about different aspects of an environment. The information includes the environment geometry, manipulator C-space and the surface material-type. The geometric information measurements include importance weights in specific regions of the environment, particularly in regions of interest close to surfaces. It was demonstrated that the geometric information of an environment is related to the industrial manipulator's C-space information, and that this information can be measured. The material-type information in voxels is measured such that a target SLD could be selected for further surface inspection and exploration. The exploration approach thus seeks to reduce the information remaining which is equivalent to minimising the uncertainty of the state of the environment.

7.1.4 Surface Representation and Map Segmentation

A surface representation and map segmentation technique has been developed which is composed of two complementary algorithms for representing the range data gathered in exploration concerning the geometry of the environment. The representation is in the form of overlapping disc-shaped targets that can be used for material-type identification, and can assist trajectory planning for maintenance operations. The Scale-Like Disc algorithm generates partially overlapping SLDs, using principal component analysis to reduce the high density surface mesh geometric data into a more compact and effective representation. The SLDs are clustered into a number of surface map segments. The map segmentation algorithm is based upon analysis of the SLD's proximity and a manipulator's task-specific tool-to-surface interaction trajectory. The map segments can assist manipulator-based tasks, such as maintenance operations, to be performed on certain surfaces.

7.1.5 Practical Contribution

The thesis has also shown a practical contribution through the integration and implementation of the devised exploration approach and algorithms. The integrated system, AXBAM, is an engineering solution for environment map building with a Denso VM-6083 6DOF industrial manipulator equipped with a Hokuyo URG-04LX laser range sensor. Experiments have been conducted in complex, static and structural environments where autonomous maintenance operations must be performed. Three experiments in environments of varying complexity have been conducted and the experimental results have demonstrated efficiency, robustness and effectiveness of the developed exploration approach and algorithms.

7.2 Discussion of Limitations

Autonomous manipulation in complex 3D environments is one of the challenges for robotics research. This thesis has addressed the challenging issue of exploring and mapping complex 3D environments with an autonomous exploration system (AXBAM) being presented. However, fully realising an autonomous system which generates an awareness of an environment so as to act upon it, or interact with it, is still a challenging development exercise.

The proposed exploration approach is limited to the use of a laser scanner that has a relatively high resolution and high accuracy and returns range and intensity data. Another type of sensor, such as a camera, may not be suitable since the developed information models rely on quality range measurements in order to predict the information remaining after sensing. The sensor must also be able to return intensity data corresponding to the range values [41], as it is necessary to determine an indication of the reflectivity and roughness of surfaces in order to perform material-type identification. The LRC system used may not always determine accurate surface material-type identifications, even when the sensor is positioned and orientated appropriately. The sensing time required to generate detailed maps is another limiting factor for the implementation. If a higher scan-rate sensor was available, it may allow the manipulator to tilt the sensor more quickly, and hence generate maps more efficiently.

The position of the base of the manipulator is assumed to be fixed securely and the manipulator is assumed to give accurate joint position information. Therefore, there are no manipulator or sensor position uncertainty problems which have been covered by this research. The algorithms presented require the sensor to be accurately positioned by an industrial manipulator. The viewpoint selection is therefore limited to a 6DOF manipulator. However, if the objective functions were modified, then it may be possible to extend the algorithms to a higher DOF manipulator.

The environments where the simulations and experiments were conducted were assumed to consist predominantly of planar surfaces and to be static. In the case where an environment contains irregular geometry (e.g. trees or rubble) or when interaction with the environment is not required, then an alternative surface representation may be appropriate. The map segmentation technique uses manipulator-based task-specific action considerations (i.e. directing a grit-blast tool and stream at a surface). However, other types of interaction with surfaces may require a different representation approach.

7.3 Future Work

The exploration approach and the implemented system have been shown to be capable of solving a few manipulator-based exploration and mapping challenges. However, there are a number of key areas which still require further work. These areas are beyond the scope of this current thesis, but it is hoped that these topics will inspire future work in this interesting area of field robotics.

In terms of pursuing improved map representations, a possible extension is the development and/or use of algorithms so that image data from a camera could improve material-type identification. The SLD generation technique could also be used to provide a 'canvas' for the textured images. It is attractive for the mesh fusion technique to be integrated with the SLD generation algorithm, since this could improve the data at corners and edges through the statistical analysis of the surface normals. Further simulations and experimental investigations are required to improve the maps generated for surface interaction planning. The segmentation technique presented depends upon the surface interaction pose selection module. Future research is needed to devise methodologies which improve segmentation for actions onto a surface. Also, in terms of the application domain, research into grit-blasting coverage could be investigated and potentially integrated with the map segmentation technique.

Apart from assumptions made about the predominantly planar nature of surfaces in the environment, this thesis only incorporates a limited amount of *a priori* knowledge about the surrounding environment. In many scenarios, an attractive extended solution to the environment mapping could consist of 'recognising' higher-level objects (e.g. I-beams, girders, crossbeams, scaffolding) from a database. It would then be advantageous, and possibly more efficient, if each object had manipulator-based actions linked to them. This has the potential to improve an autonomous system's ability to map and interact with the environment. Assumptions about the static nature of the environment could also be discarded with an attempt to explore with the manipulator in a dynamic environment. In a dynamic case, the voxels could be in permanent flux, such that they are constantly shifting state from known (obstacles or freespace) back to unknown.

Currently the uncertainty of sensing positions is not included in the exploration approach. Exploration and mapping while moving the mobile manipulator platform, is still a challenging on-going research issue. When a manipulator is mounted on a mobile robot (e.g. a moving base) then extra degrees of freedom become available. However, when the manipulator base is not static, the position of the sensor may not be known accurately. Thus, the uncertainty associated with the sensing position must be incorporated. Future work should extend the presented approach so that the platform can be moved to multiple positions along the rail and perform exploration and mapping. This type of solution can be particularly desirable in applications where the manipulator would enable more detailed 3D mapping (e.g. in search and rescue). In this case, separate localisation algorithms need to be developed to expand the exploration algorithm to deal with robot movement.

Future work should look at integrating the two-stage exploration approach presented with a carefully devised manipulator movement planner and collision detection algorithm. One interesting research topic could be to improve manipulator-based exploration by developing skin-type sensors for collision avoidance during movement. This would be useful where the map is uncertain or incorrect, or where there is inconsistent sensor data returned. Potentially this could be done by adapting the sensing field of a capacitive-type sensor and using multiple sensors in order to provide entire manipulator encompassing sensing zones to determine the proximity to obstacles. Such technology is envisaged to aid in the exploration approach as it can reduce the uncertainty around the manipulator. Further research into these challenges should be performed in order to further improve the capabilities of manipulator-based robotic systems working in unknown, unstructured and/or dynamic environments.

Appendix A

Grit-blasting Pose Selection and Planning

Appendix A is an extract from our published works [25] and [19], on an efficient trajectory planning approach for autonomous robots in complex bridge environments. This work experimentally verified the maps generated by the approach used in this thesis (i.e. implemented as AXBAM), and the efficacy of utilising the target making algorithm as the basis of a surface interaction (i.e. grit-blasting) optimisation technique. This Appendix is intended as supplementary material for Scale-Like Disc (SLD) generation, map segmentation and the usage of the maps acquired during exploration experiments.

During exploration, prior to the grit-blast planning process, the surface geometry and material-types are determined. Following this, SLDs are generated using the PCA-based algorithm on the point cloud. This reduces the complexity of the grit-blasting task by simplifying the number of parameters required to represent a segmented surface, and providing manageable sets of targets on the surface. The sensing tool is replaced by the maintenance tool, and task-specific manipulator pose selection is performed. Once the poses have been determined and the map segmentation algorithm from this thesis is run, then the sequence in which the surfaces are to be grit-blasted is determined using the Genetic Algorithm (GA) optimisation technique.

A.1 Manipulator Pose Selection for Grit-blasting

A gradient-based pose selection optimisation algorithm was developed using least squared criteria similar to Stage Two from Chapter 3. This enables the joint configuration to be found for the grit-blasting application. The objective of pose selection is to determine a valid joint configuration solution that satisfies the specified pose selection criteria including distance away from the desired target, difference in orientation from current manipulator configuration, and joint limitations. Rapid joint configuration discovery is significant for onsite bridge maintenance planning due to the low repeatability of tasks. Therefore, for practical purposes, the pose selection algorithm is designed towards the rapid discovery of a valid joint configuration for a given SLD target. The grit-blast task-specific values, i.e. blast stream length and blast angle, are also considered in the pose selection criteria so as to satisfy the assumption of fixed blast effect. These parameters for the objective functions are shown in Fig. A.1.

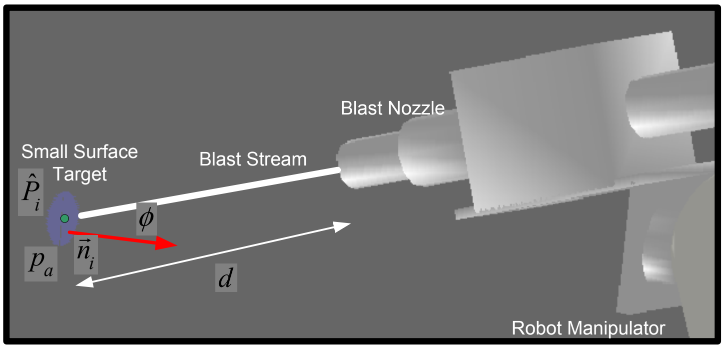


Figure A.1: Parameters which must be optimised to achieve successful grit-blasting maintenance results.

The grit-blasting stream meets an object's surface with an angle of incidence. This angle affects the effective nature of the maintenance and the angle formed with the target surface, ϕ, and must be less than a maximum angle, ϕ_{max}.

$$\min_{\vec{Q}} g_1(\vec{Q}) = \frac{1}{1 + e^{-(\phi - \phi_{max})}} \tag{A.1}$$

The length of an effective grit-blast stream, d, must be between d_{min} and d_{max}.

$$\min_{\vec{Q}} g_2(\vec{Q}) = 1 - \frac{1}{1+e^{-(d-d_{min})}} + \frac{1}{1+e^{-(d-d_{max})}} \tag{A.2}$$

The grit-blasting stream is highly dangerous and destructive, hence it must be aimed at a specific target surface accurately. The centre of blast stream hits the surface at a point vector, $\mathbf{p}_a$, while the centre of the desired ith SLD target is $\hat{P}_i$. The minimum allowable distance tolerance between $\mathbf{p}_a$ and $\hat{P}_i$ is ρ_{min}

$$\min_{\vec{Q}} g_3(\vec{Q}) = 1 - \frac{1}{1+e^{-(\|\mathbf{p}_a-\hat{P}_i\|-\rho_{min})}} \tag{A.3}$$

Implicitly, the safety of the manipulator, which has the maintenance tool affixed to the end-effector, must be maintained through this pose selection optimisation. Collision avoidance is implemented using the ellipsoidal bounding fields around each manipulator link based upon the combination of research [126] and [128] similar to Stage One in Chapter 3. The ith manipulator joint is once again enclosed by ellipsoidal virtual bounding fields, centred at $\mathbf{p}_{c,i}$, and with parameters $[a_{e,i}, b_{e,i}, c_{e,i}]$. Since an obstacle (or unexplored voxel), $\mathbf{p} \in \mathsf{P}$, within an ellipsoid has an algebraic distance less than 1, the minimum constraint for the sigmoid is 1. For each of the six joints, q_i in $\vec{Q}$, the corresponding ellipsoid's algebraic distance, $dist(\vec{Q})$, to all obstacles and unknown voxels is returned using the minimum algebraic distance to points from a pose. Then this can be used to alter the fourth objective sigmoid-wrapped cost function.

$$\min_{\vec{Q}} g_4(\vec{Q}) = 1 - \frac{1}{1+e^{-\left(dist(\vec{Q})-1\right)}} \tag{A.4}$$

For a viewpoint, the manipulator joints are described by the vector of joints, $\vec{Q}$, which must fall within the manipulator's physical limitations: a positive maximum angle, $q_{i,max}$, and negative minimum, $q_{i,min}$, for the $i \in \{1, \dots 6\}$ joints . This objective function, $g_5(\vec{Q})$, has combined the six pairs of joint constraints into one function as follows

$$\min_{\vec{Q}} g_5(\vec{Q}) = \sum_{i=1}^{6} \left(1 - \frac{1}{1+e^{-(q_i-q_{i,min})}} + \frac{1}{1+e^{-(q_i-q_{i,max})}}\right)^2 \tag{A.5}$$

Similar to Stage Two of the exploration approach, the problem was once again formulated so that it can be solved by the Levenberg-Marquardt (LM) algorithm [162]. LM is an established iterative technique which can be used to locate the minimum of the sum of squares of non-linear real-valued function, $\mathbf{g}(\vec{Q})$, as shown in [163] where the desired value $\hat{g}_i$, for all $i \in \{1, \dots 5\}$ cost functions is once again zero (i.e. $\hat{g}_i = 0$).

A.2 GA-based Optimisation of Target Point Sequencing

A GA is applied for the optimisation of the initial path and motion plan. The GA optimisation objectives include the travel distance and joint movement. A variation of the linear aggregating function [168] is chosen for the lower computational load. Path planning steps are given in Fig. A.2

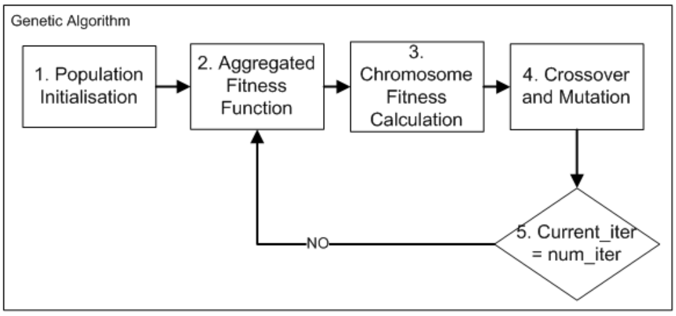


Figure A.2: Optimisation by GA with normalised parameters.

1. Population Initialisation: The algorithm is initialised with a fixed population size of n_{cr} chromosomes. In the applied context of path plan optimisation, a chromosome is defined as a fixed size array containing target points where the jth target SLD, is centred at $\hat{P}_j = [\mathsf{x}_j, \mathsf{y}_j, \mathsf{z}_j]^T$. The initial path plan derived during the blast target point generation process is applied as the initial chromosome in the first generation of the GA. The purpose of the initial path plan is to provide an educated guess for the search direction of the GA.

2. Fitness Function: The fitness function, $g^i_{j,j+1}$, of the ith chromosome from the j point to the $(j+1)$th point is defined as the weighted sum of the travel distance/path length and the joint movement: Equation A.6.

$$g^i_{j,j+1} = \alpha d^i_{j,j+1} + \beta m^i_{j,j+1} \tag{A.6}$$

where α and β are weighting co-efficients which are applied to normalise the travel distance and joint movement. The travel distance, d, is defined as the Euclidean distance travelled between the two target points, $\hat{P}_j$and $\hat{P}_{j+1}$:

$$d^i_{j,j+1} = \|\hat{P}_{j+1} - \hat{P}_j\| \tag{A.7}$$

The joint movement, m, is defined as the sum of angles of the robot's six joints q_1 to q_6, $l \in [1, 6]$ required to move from point $\hat{P}_j$ to point $\hat{P}_{j+1}$.

$$m^i_{j,j+1} = \sum |q_{l,j+1} - q_{l,j}| \tag{A.8}$$

3. Chromosome Fitness Calculation: The fitness (f^i) value of each chromosome is the summation of the aggregated fitness values calculated from Equation A.6 for the movement along the chromosome target point index sequence. The aim of the GA is to minimise the fitness function.

$$f^i = \sum_{j=1}^{n_{ss}-1} g^i_{j,j+1} \tag{A.9}$$

4. Crossover and Mutation: Successive populations are generated using genetic operators including; reproduction, crossover and mutation. The new population is generated from a mating pool of size $\frac{n_{cr}}{2}$, created through a tournament selection process where chromosomes from the current population are randomly paired and fitness compared. $\frac{n_{cr}}{4}$ of the best fitness chromosomes from the mating pool are selected for reproduction using a similar tournament selection process. Crossover is performed by randomly pairing parent chromosomes selected from the mating pool to generate the $\frac{n_{cr}}{4}$ child chromosomes. The remaining $\frac{n_{cr}}{2}$ chromosomes of the new population are created through the mutation of the reproduced and child chromosomes. Mutation occurs either as the interchange of the values from two randomly

selected array slots or by inverting all values within the range of the two randomly selected array slots. The type of mutation used is randomly selected.

5. Termination Condition Checking: If the generation limit has not been reached, then proceed to step 2. Else terminate the algorithm.

Adaptive weighting factor values are calculated by evaluating the fitness distribution in the population for every generation. The weighting factors α and β are defined as a ratio value, calculated from the distribution means μ_T and μ_J.

$$\begin{aligned} \mu_T &= \tfrac{1}{n_{cr}} \textstyle\sum_{i=1}^{n_{cr}} \sum_{j=1}^{n_{ss}-1} d_{j,j+1}^{i} \\ \mu_J &= \tfrac{1}{n_{cr}} \textstyle\sum_{i=1}^{n_{cr}} \sum_{j=1}^{n_{cr}-1} m_{j,j+1}^{i} \\ \alpha &= \tfrac{\mu_T}{\mu_T+\mu_J} \\ \beta &= \tfrac{\mu_J}{\mu_T+\mu_J} \end{aligned} \tag{A.10}$$

This results in an efficient approach for robotic path planning with the implementation of GA for plan optimisation. The approach has been demonstrated to provide a solution that can be practically executed to perform grit-blasting tasks. It has the ability to generate blasting target points and generate a near-optimal path for any map segment made of SLDs, and then select robot joint poses to ensure a safe efficient path.

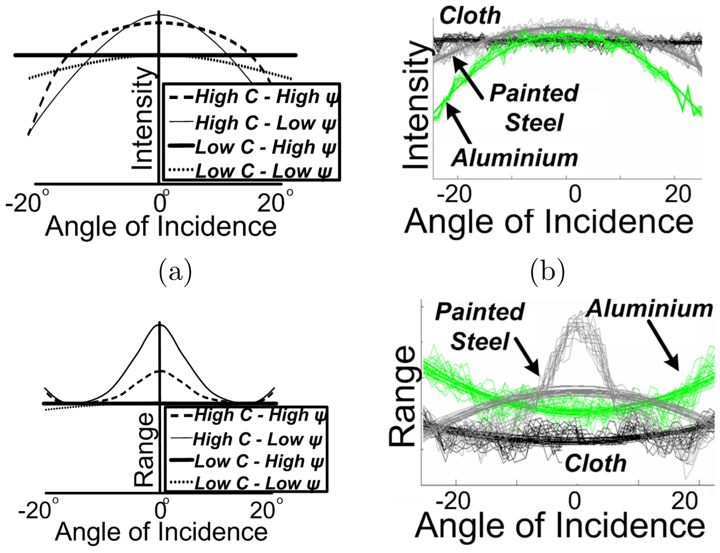
Appendix B

Laser Range Classifier for Material-type Identification

Appendix B briefly introduces the workings and constraints of the Laser Range Classifier (LRC), which we published in [5], and have used to identify the material-type of surfaces. In general, laser range scanners such as the Hokuyo URG-04LX [40] selected for this thesis, operate by emitting light and receiving reflections off the surface, scaling them based on their return intensity and then measuring phase shift. This enables the range to an object's surface to be determined. It has been shown, [5], that surface material-type identification is achievable in some specific applications through the study of the physical properties that affect the light reflected from a surface. The system which processes the range/ intensity data (which the laser range scanner can be especially configured to return [41]) to identify the surface material-type is named the Laser Range Classifier (LRC).

In order to use a laser range finding sensor to identify the material-type of a surface, a method for determining the relevant physical properties (i.e. surface roughness, ψ, and reflectance, C_r) of the surface is vital. The Hancock [4] model incorporates the "specular component" and "diffused light component" to account for changes in reflection (i.e. the intensity), based on the incidence angle. The model was tested across varying angles of incidence to determine the effect of surface roughness and reflectance on intensity and range measurement error. As shown in Fig. B.1, surface roughness and reflectance (ψ

and C_r, respectively) have a unique effect on the shape of the range and intensity curves. Therefore, the shape of the curve is representative of the values of these parameters. Fig. B.1*a* and Fig. B.1*c* show the Hancock model results for intensity and range returns verses the empirical results in Fig. B.1*b* and Fig. B.1*d*. A more reflective surface increases the magnitude of the return intensity for the central region of the curve; the increase is non-linear across the angles of incidence within the region, with the largest ratio of increase being at 0°. The more reflective surface also has the effect of significantly reducing the magnitude of the return intensities in edge regions. Conversely, an increase in surface roughness results in an increase in the return intensities in the edge regions, but has negligible effect on the centre region (i.e. when the angle of incidence relative to the surface normal is 0°). Thus, if a curve exists, an estimation of the parameters ψ and C_r is possible.



(a) (b)

(c) (d)

FIGURE B.1: *a*) and *b*) The effect of surface roughness (ψ) and reflectance (C_r) on the *reflected light's intensity* illustrating how different parameter combinations produce uniquely shaped return intensity curves. *a)* Via Hancock model [4]; and *b)* via experiment [5]. *c) and d)* The effect of ψ and C_r on the *range measurement error* with the *c*) Hancock model; and *d*) experimental data.

For such curves to be constructed, intensity readings from a range of angles of incidence are required. As a laser scanner is being used to gather the intensity data, this is inherently the case [41]. However, this method relies on the assumption that the surface being scanned is flat within the angles of incidence ±20°. A second-order polynomial fit can be applied to represent the range and intensity data (Fig. B.1*b* and Fig. B.1*d*). Although a second-order polynomial fit does not provide an accurate representation of the curve for all cases (e.g. painted steel in Fig. B.1*d*), the typically unstable nature of the higher-order co-efficients makes analysis problematic. In order to minimise information loss due to using a second-order fit, the extent to which the polynomial fails to accurately represent the curve (the mean-squared error of the residuals - MSE) is calculated. A low MSE indicates that the surface material-type can be identified with high-confidence. Fig. B.1 shows the central region is predominately affected by the reflectance. An increase in the angle of incidence to the plane results in an intensity level drop across the data set. Therefore, to increase the LRC's robustness, the centre ray of the scanner should be close to perpendicular to the surfaces so that the angle of incidence is kept within desirable bounds. For the Hokuyo URG-04LX, the $\pm 20°$ sensing field and the density of returned data required to perform correct polynomial fitting means that a targeted surface must be within $\approx 500mm$ of the sensor. For a scanning resolution of $0.36°$, the sensing area is theoretically constrained to a disc (radius$\approx 60mm$) on a planar surface. Thus, in order for the LRC to successfully identify the surface material-types, the sensing viewpoint must be carefully considered.

The LRC's classification encapsulates the roughness and reflectance properties of a surface and uses them as the basis for a Bayesian approach to surface material-type identification [166]. Likelihoods are calculated from probability density functions (of the four parameters: co-efficient of the second-order term in the polynomial and MSE for both the intensity and range data) derived from training data. The environment is assumed to be stationary, such that the surface material-type state in a voxel does not change with time. The LRC system is therefore able to identify surface material-types using the intensity and range curves, provided that the laser ray's angle of incidence with the surface is near perpendicular, and the other sensing constraints are considered.

Appendix C

A Robotic System for Steel Bridge Maintenance

Appendix C describes the hardware that is used by the overall maintenance system and also by the AXBAM system. The robotic maintenance system design was published in [26]. The bridge maintenance robotic system consists of a 6DOF industrial robot, a moving platform, a sensor package equipped with a laser range finder, cameras, a capacitive sensor network, and a small desktop computer mounted onboard. A relatively large robot is needed in order to handle the blasting nozzle reaction force. The whole system is placed on the floor of a fully enclosed scaffold. The system must be able to work in three modes: manual, semi-autonomous and autonomous. Manual mode is used for system testing and cleaning the most difficult sections that cannot be handled automatically. Semi-autonomous mode allows the system to automatically blast a defined region, e.g. a given face of a girder, chosen by a remote operator based on the map which is obtained through autonomous exploration and map building. Autonomous mode is where identifying the areas to be blasted, grit-blasting path planning and actual grit-blasting of the steel structure occurs automatically. The safety of the system is ensured by strategies and technologies for real-time collision detection and avoidance.

As most of the steel bridges do not have complete up-to-date CAD drawings due to the age of the bridges and possible changes made to the bridges, the bridge structure environment is initially unknown or partially known by the robotic system. As shown in this thesis, gathering the environment map is vital. Planning of the blasting nozzle's path and robot motion is conducted after the map of the environment is built and the areas to be blasted have been identified. Path and motion planning must attempt to maximise the grit-blasting coverage completeness, minimise the movement of the robot manipulator and the support platform and maximise efficiency and productivity. Challenges of planning also include partitioning a large surface (as covered by this thesis) and planning for blasting edges of structure members to prevent the adjacent non-metal members being damaged. Due to the complexity of the grit-blasting environment and the large robot size relative to the environment, collision avoidance is a very important issue. Collision avoidance is included in various stages of the autonomous operation including exploration, mapping and performing maintenance actions. This is achieved in real operation through a collision control strategy (including sensors and algorithms) to avoid potential collisions due to robot malfunction or any sudden changes of the environment.

C.1 Hardware

C.1.1 Industrial Manipulator

The Denso robot manipulator Denavit and Hartenberg parameters and limitations are shown in Table C.1. From this it is clear that the manipulator has a large range of movement and it is possible to position a sensor mounted on the end-effector within the workspace with a variety of different configurations.

TABLE C.1: Denavit and Hartenberg [6] parameters, joint types and limits for the Denso VM-6083 manipulator arm, *Note 1*): all joint types are revolute, *Note 2*) the rotation angle parameter θ of all joints is 0.

Link	Twist α_i	Length a	d Offset	State at Zero	Min (deg)	Max (deg)
1	$-\frac{\pi}{2}$	0.18	0.475	0	-170	170
2	0	0.385	0	$-\frac{\pi}{2}$	-90	135
3	$\frac{\pi}{2}$	-0.1	0	$\frac{\pi}{2}$	-80	165
4	$-\frac{\pi}{2}$	0	0.445	0	-185	185
5	$\frac{\pi}{2}$	0	0	0	-120	120
6	0	0	0.084	0	-360	360

C.1.2 Laser Range Finding Sensor

The relevant parameters of the laser range finding scanning sensor are shown in Table C.2. This sensor is lightweight and highly accurate within the necessary workspace range. The sensor is mounted on the end-effector. The 2D laser can be tilted through an angle using either the 5th or 6th manipulator joint to achieve a 3D FOV. External maintenance interaction will require a simple change of head to the blasting nozzle.

Table C.2: Hokuyo URG-04LX Scanning Laser Range Finder relevant specifications. The sensor is lightweight, low power, short range with high accuracy and high resolution.

Light source	Semiconductor laser $\Lambda = 785nm$
Power source (current)	5VDC (< 500mA)
Resolution	1mm
Angular Resolution	$\approx 0.36°$
Accuracy	10mm for objects $20mm \rightarrow 1000mm$, 1% for objects $1000mm \rightarrow 4000mm$
Weight	160g

C.1.3 Computing Hardware

All testing of algorithms and implementation was performed on the production system PC: an off-the-shelf dual core P1.66GHz Mac-mini. Code is written in C++ and MATLAB with MS Windows COMs components handling inter-thread communication. The computer controls the platform's air supply for pneumatic motors and receives positional information from platform encoders and fixed way stations. The computer also controls the air and grit supply used for the maintenance application.

Appendix D

SLD Generation: Additional Parameter Tuning Results

Appendix D extends upon the SLD generation parameter tuning results presented in Chapter 5 of this thesis. In this case the number of points input to the algorithm, n_p is fixed and both n_{min} and μ are varied. The results (i.e. time, n_{ss} and %) are shown in Fig. D.1. Fig. D.1*a* shows the changes to the time taken to generate SLDs. Fig. D.1*b* shows the number of SLDs created n_{ss}, and Fig. D.1*c* shows the percentage (%) of point registered.

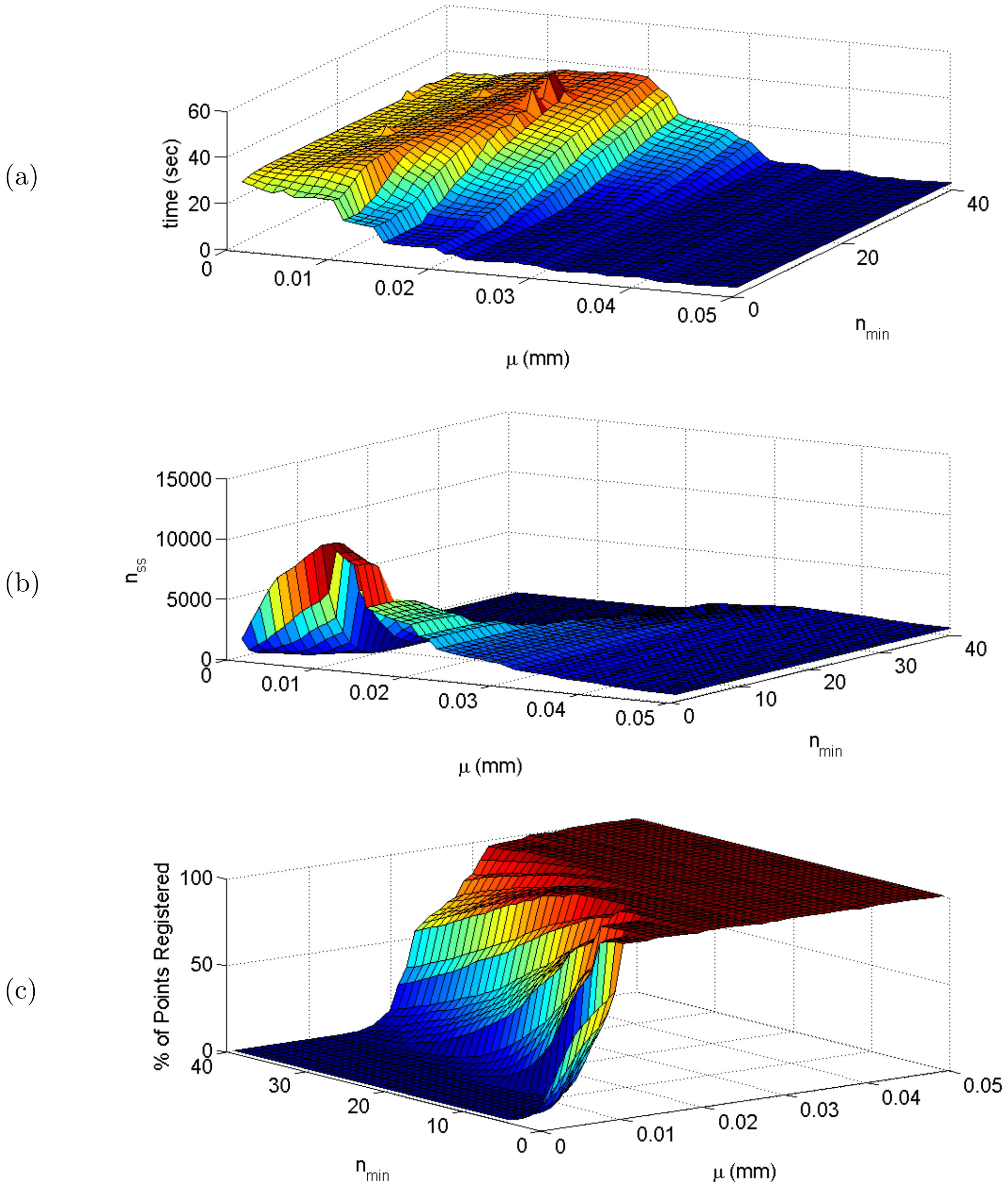


FIGURE D.1: With respect to changing n_{min} and μ *a*) The time taken to make surfaces; *b*) Number of small SLDs created n_{ss}; *c*) The percentage (%) of points registered.

Bibliography

[1] D. Ferguson, A. Morris, D. Haehnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, W. Whittaker, W. Burgard, and S. Thrun. An autonomous robotic system for mapping abandoned mines. In *Proc. of the Conference on Advances in Neural Information Processing Systems*, pages 79–91, Vancouver, 2003.

[2] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini. Roboscan: an automatic system for accurate and unattended 3D scanning. In *Proc. 2nd International Symposium on 3D Data Processing, Visualization and Transmission*, pages 805– 812, Thessaloniki, 2004.

[3] Y. Yu. *An Information theoretical incremental approach to sensor-based motion planning for eye-in-hand systems.* PhD thesis, Simon Fraser University, 2000.

[4] J. Hancock. *Laser Intensity-Based Obstacle Detection and Tracking.* PhD thesis, Carnegie Mellon University, 1999.

[5] N. Kirchner, D.K. Liu, T. Taha, and G. Paul. Simultaneous material type classification and mapping data acquisition using a laser range finder. In *Proc. Intellegent Technology in Robotics and Automation*, pages 124–129, Sydney, 2007.

[6] M. Spong and M. Vidyasagar. *Robot Dynamics and Control.* Wiley, 1989.

[7] Engineers Australia. Another $2 billion for Queensland infrastructure. *Engineers Australia General Edition*, 77(11):25, 2005.

[8] C.H Hare. Protective coatings for bridge steel. Technical Report 0-309-04421-9, Transportation Research Board, 1987.

[9] N. Kirchner, G. Paul, and D. K. Liu. Bridge maintenance robotic arm: Mechanical technique to reduce the nozzle force of a sandblasting rig. In *Proc. 1st International Symposium on Digital Manufacturing*, pages 12–18, Wuhan, China, 2006.

[10] B. Joode, C. Verspuy, and A. Burdorf. Physical workload in ship maintenance: Using the observer to solve ergonomics problems. Technical report, Erasmus University of Rotterdam, 2004.

[11] S. Lahiri, C. Levenstein, D. Nelson, and B. Rosenberg. The cost effectiveness of occupational health interventions: Prevention of silicosis. *American Journal of Industrial Medicine*, 48(6):503–517, 2005.

[12] B. Ross, J. Bares, and C. Fromme. A semi-autonomous robot for stripping paint from large vessels. *The International Journal of Robotics Research*, 22(7-8):617–626, 2003.

[13] W. Schmitz. Robotic paint stripping of large aircraft - a reality with the flashjet coatings removal process. In *Proc. Aerospace Coatings Removal and Coatings*, pages 1–10, Colorado Springs, 2003.

[14] D. Mcguire. Method of robotic automobile paint stripping (patent app. no. 6287389), Nov. 2001.

[15] W. Yan, L. Shuliang, X. Dianguo, Z. Yanzheng, S. Hao, and G. Xueshan. Development and application of wall-climbing robots. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1207–1212, Detroit, 1999.

[16] S. Moon and L. Bernold. Vision-based interactive path planning for robotic bridge paint removal. *Journal of Computing in Civil Engineering*, 11(2):113–120, 1997.

[17] Clemco Industries. *Abrasive Blasting Safety Practices.* Clemco Industries, Washington, 2002.

[18] Clemco Industries Corporation. *Blast Off: Your Guide to Safe and Efficient Abrasive Blasting.* Clemco Industries, Washington, 1994.

[19] W.K. To, G. Paul, N.M. Kwok, and D.K. Liu. An integrated approach to planning for autonomous grit-blasting robot in complex bridge environments. In *Proc.*

*The Fourth Virtual International Conference of the EU-funded FP6 I*PROMS Network of Excellence on Innovative Production Machines and Systems*, pages 313–318, Wales, 2008.

[20] N. Kirchner. *Exploiting Laser and Capacitive Ranging Sensors' Behaviour to Yield Mission-Critical Material Type Identifications In Real-World Environments.* PhD thesis, University Of Technology, Sydney, 2008.

[21] G. Paul, D. K. Liu, N. Kirchner, and S. Webb. Safe and efficient autonomous exploration technique for 3D mapping of a complex bridge maintenance environment. In *Proc. 24th International Symposium on Automation and Robotics in Construction*, pages 99–104, Kochi, India, 2007.

[22] G. Paul, D. K. Liu, and G. Dissanayake. Autonomous robot manipulator exploration and mapping system for bridge maintenance. *Robotics and Autonomous Systems, Under review - submitted April*, 2008.

[23] G. Paul, N. Kirchner, D. K. Liu, and G. Dissanayake. An effective exploration approach to simultaneous mapping and surface material-type identification of complex 3D environments. *Journal of Field Robotics, Special Issue on Three-Dimensional Mapping*, 26(11-12 SI):915–933, 2009.

[24] G. Paul, D. K. Liu, and N. Kirchner. An algorithm for surface growing from laser scan generated point clouds. In T. Tarn, S. Chen, and C. Zhou, editors, *Robotic Welding, Intelligence and Automation*, pages 481–491. Springer-Verlag, Berlin, 2007.

[25] W. K. To, G. Paul, N. M. Kwok, and D. K. Liu. An efficient trajectory planning approach for autonomous robots in complex bridge environments. *International Journal of Computer Aided Engineering and Technology*, 1(2):185–208, 2009.

[26] D.K. Liu, G. Dissanayake P. B. Manamperi, G. Fang, N. Kirchner, G. Paul, S. Webb, P. Chotiprayanakul, and J. Xie. A robotic system for steel bridge mainteance: Research challenges and system design. In *Proc. Australasian Conference on Robotics and Automation ACRA*, Canberra, 2008.

[27] N. Kirchner, D. K. Liu, T. Taha, and G. Paul. Capacitive object ranging and material type classifying sensor. In *Proc. Intellegent Technology in Robotics and Automation*, pages 130–135, Sydney, 2007.

[28] N. Qaddoumi, M. Abu-Khousa, and T. H. Ibrahim. Quantitative rust-under-paint detection utilizing near-field microwave nde techniques. In *Proc. 16th World Conference on Nondestructive Testing*, Montreal, 2004.

[29] H. L. Kung, S. R. Bhalotra, J. D. Mansell, D. A. B. Miller, and J. S. Harris. Standing-wave fourier transform spectrometer based on integrated mems mirror and thin-film photodetector. *IEEE Journal of Selected Topics in Quantum Electronics*, 8(1):98–105, 2002.

[30] R. Soon, N. R. Mysoor, and S. R. Carnes. Miniature magnetic resonance spectrometers. In *Proc. 16th Digital Avionics Systems Conference, DASC, AIAA/IEEE*, volume 1, pages 14–23, CA, USA, 1997.

[31] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, 2003.

[32] H. Yu, Y. Wang, and J. Peng. An occupancy grids building method with sonar sensors based on improved neural network model. In *Proc. 4th international symposium on Neural Networks: Advances in Neural Networks*, pages 592–601, Nanjing, China, 2007.

[33] J. S. Gutmann, M. Fukuchi, and M. Fujita. A floor and obstacle height map for 3D navigation of a humanoid robot. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 1066–1071, Barcelona, Spain, 2005.

[34] J. M. Saez and F. Escolano. A global 3D map-building approach using stereo vision. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1197–1202, New Orleans, 2004.

[35] K. Deguchi and T. Nakagawa. Active and direct acquisition of 3D map in robot by combining motion and perceived images. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 2, pages 1692–1697, Sendai, Japan, 2004.

[36] R. Bajcsy, R. Enciso, G. Kamberova, L. Nocera, and R. Sara. 3D reconstruction of environments for virtual collaboration. In *Proc. Fourth IEEE Workshop on Applications of Computer Vision, WACV*, pages 160–167, New Jersey, 1998.

[37] G. Ruz, P. Estevez, and C. Perez. A neurofuzzy color image segmentation method for wood surface defect detection. *Forest Products Journal*, 55(4):52–58, 2005.

[38] I. Stamos and P. K. Allen. Integration of range and image sensing for photo-realistic 3D modeling. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1435–1440, San Francisco, 2000.

[39] S. Thrun, C. Martin, L. Yufeng, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 20(3):433–443, 2004.

[40] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori. Development of ultra-small lightweight optical range sensor system. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1078–1083, Alberta, Canada, 2005.

[41] H. Kawata. Communication protocol specification (intensity output mode), drawing no. c-42-3320-c. Technical report, Hokuyo Automatic Co., Ltd, Dec 5th 2006.

[42] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Computer graphics proceedings, annual conference series*, volume 2006, pages 303–312, New Orleans, 1996. Association for Computing Machinery SIGGRAPH.

[43] S. S. Webb. *Belief Driven Autonomous Manipulator Pose Selection for Less Controlled Environments.* PhD thesis, University of New South Wales Australia, 2008.

[44] Z. Teng, H. Y. Feng, and A. Azeem. Generating efficient tool paths from point cloud data via machining area segmentation. *The International Journal of Advanced Manufacturing Technology*, 30(3-4):254–260, 2006.

[45] M. Varsta and P. Koikkalainen. Surface modeling and robot path generation using self-organization. In *Proc. 13th International Conference on Pattern Recognition*, volume 4, pages 30–34, Vienna, Austria, 1996.

[46] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*, pages 1–35. Morgan Kaufmann, San Francisco, 2002.

[47] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[48] H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2): 61–74, 1988.

[49] P. Stepan, M. Kulich, and L. Preucil. Robust data fusion with occupancy grid. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(1):106–115, 2005.

[50] K. H. Ko, T. Maekawa, and N. M. Patrikalakis. Algorithms for optimal partial matching of free-form objects with scaling effects. *Graphical Models*, 67(2):120–148, 2005.

[51] H. Zender, O. Martinez Mozos, P. Jensfelt, G.-J.M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

[52] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[53] S. Vasudevan and R. Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56 (6):522–537, 2008.

[54] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.

[55] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, 1987.

[56] G. M. Nielson. On marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):283–297, 2003.

[57] K. Levinski and A. Sourin. Interactive function-based artistic shape modeling. In *Proc. IEEE First International Symposium on Cyber Worlds*, pages 521–528, Tokyo, 2002.

[58] B. Barber and H. Huhdanpaa. Qhull manual. Technical report, The Geometry Center, 1995.

[59] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, Los Angeles, 1992.

[60] H. Woo and T. K. Dey. Updating 3D triangular mesh models based on locally added point clouds. *The International Journal of Advanced Manufacturing Technology*, 30 (3-4):261–272, 2006.

[61] D. H. Laidlaw, K. W. Fleischer, and A. H. Barr. Partial-volume bayesian classification of material mixtures in mr volume data using voxel histograms. *IEEE Transactions on Medical Imaging*, 17(1):74–86, 1998.

[62] J. W. Weingarten, G. Gruener, and R. Siegwart. Probabilistic plane fitting in 3D and an application to robotic mapping. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 1, pages 927–932, New Orleans, 2004.

[63] T. R. Ren, N. M. Kwok, D. K. Liu, and S. D. Huang. Path planning for a robotic arm sand-blasting system. In *Proc. International Conference on Information and Automation, ICIA*, pages 1067–1072, Hunan, China, 2008.

[64] L. I. Smith. A tutorial on principal component analysis. Technical report, 2002.

[65] A. Godil, S. Ressler, and P. Grother. Face recognition using 3D surface and color map information: Comparison and combination. In *Proc. SPIE's symposium on Biometrics Technology for Human Identification*, Orlando, FL, 2004.

[66] J. L. Crowley, F. Wallner, and B. Schiele. Position estimation using principal components of range data. In *Proc. IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[67] X. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Application*, 7(2):115–122, 1994.

[68] J. Peters and X. Wu. Optimized refinable surface enclosures. Technical report, University of Florida, 2000.

[69] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi. Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage. *IEEE Transactions on Automation Science and Engineering*, 6(1):107–120, 2009.

[70] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Journal of Robotics and Automation*, 17(3):229–241, 2001.

[71] K. M. Chow, A. B. Rad, and Y. L. Ip. Enhancement of probabilistic grid-based map for mobile robot applications. *Journal of Intelligent and Robotic Systems*, 34 (2):155–174, 2002.

[72] J. Minguez and L. Montano. Nearness diagram (ND) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1): 45–59, 2004.

[73] K. Okada, S. Kagami, M. Inaba, and H. Inoue. Plane segment finder: algorithm, implementation and applications. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 2120–2125, Tokyo, 2001.

[74] M. Yguel, O. Aycard, and C. Laugier. Efficient GPU-based construction of occupancy girds using several laser range-finders. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 105–110, Beijing, 2006.

[75] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics.* MIT Press, Cambridge, Mass., 2005.

[76] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-based laser scan matching for outdoor 6D SLAM. In *Proc. 28th Annual German Conference on AI, KI 2005: Advances in Artificial Intelligence*, pages 304–319, Koblenz, Germany, 2005.

[77] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[78] H. Moravec. Dense 3D perception for broad mobile robot applications. Technical report, Carnegie Mellon University, 2003.

[79] D. M. Cole and P. M. Newman. Using laser range data for 3D SLAM in outdoor environments. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 1556–1563, Orlando, 2006.

[80] N. Fairfield, G. Kantor, and D. Wettergreen. Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 3575–3580, Orlando, 2006.

[81] I. V. Isler. *Algorithms for Distributed and Mobile Sensing.* PhD thesis, University of Pennsylvania, 2004.

[82] H. Surmann, A. Nuchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198, 2003.

[83] M. Montemerlo, D. Haehnel, D. Ferguson, R. Triebel, W. Burgard, S. Thayer, W. Whittaker, and S. Thrun. A system for three-dimensional robotic mapping of underground mines. Technical report, Carnegie-Mellon Univ Pittsburgh, School Of Computer Science, 2002.

[84] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard. Autonomous terrain mapping and classification using hidden Markov models. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 2026–2031, Barcelona, 2005.

[85] S. Larsson and J. A. P. Kjellander. Motion control and data capturing for laser scanning with an industrial robot. *Robotics and Autonomous Systems*, 54(6):453–460, 2006.

[86] L. Haichao, G. Hongming, and W. Lin. Supervisory control of telerobotic system for remote welding. In *Proc. IEEE International Conference on Integration Technology, ICIT*, pages 603–608, Shenzhen, China, 2007.

[87] A. M. Dollar and R. D. Howe. Towards grasping in unstructured environments: optimization of grasper compliance and configuration. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 4, pages 3410–3416, Las Vegas, Nevada, 2003.

[88] K. Maruyama, K. Oka, R. Takase, Y. Kawai, T. Yoshimi, H. Takahashi, and F. Tomita. 3D localization of partially buried object in unstructured environment. In *Proc. 19th International Conference on Pattern Recognition, ICPR*, pages 1–4, Florida, 2008.

[89] M. C. Martin and H. P. Moravec. Robot evidence grids. Technical report, The Robotics Institute, Carnegie Mellon University, 1996.

[90] V. Isler, S. Kannan, and K. Daniilidis. Local exploration: online algorithms and a probabilistic framework. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1913–1920, Taipei, Taiwan, 2003.

[91] K. M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1160–1165, Nice, France, 2008.

[92] L. Yu, L. Xiaoyong, and Z. Shiqiang. Combined coverage path planning for autonomous cleaning robots in unstructured environments. In *Proc. 7th World*

Congress on Intelligent Control and Automation, WCICA, pages 8271–8276, Chongqing, China, 2008.

[93] S. Garrido, L. Moreno, and D. Blanco. Exploration of a cluttered environment using voronoi transform and fast marching. *Robotics and Autonomous Systems*, 56(12): 1069–1081, 2008.

[94] Y. L. Ip, A. B. Rad, and Y. K. Wong. Autonomous exploration and mapping in an unknown environment. In *Proc. International Conference on Machine Learning and Cybernetics*, volume 7, pages 4194–4199, Shanghai, 2004.

[95] R. Martinez-Cantin, N. Freitas, A. Doucet, and J.A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Proc. Robotics Science and Systems, RSS*, Atlanta, 2007.

[96] T. Kollar and N. Roy. Trajectory optimization using reinforcement learning for map exploration. *International Journal of Robotic Research*, 27(2):175–196, 2008.

[97] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[98] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions.* Probability and Statistics. Wiley and Sons, New York, 1997.

[99] R. N. Neal and G. E. Hinton. A view of the em algorithm that justies incremental, sparse and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, Dordrecht, 1998.

[100] F. Dellaert. Addressing the correspondence problem: A Markov chain Monte Carlo approach. Technical report, Robotics Institute, Carnegie Mellon University, 2000.

[101] R. Benenson, S. Petti, T. Fraichard, and M. Parent. Integrating perception and planning for autonomous navigation of urban vehicles. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 98–104, Inria, Paris, 2006.

[102] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and System, IROS*, volume 1, pages 540–545, Switzerland, 2002.

[103] V. A. Sujan and S. Dubowsky. Efficient information-based visual robotic mapping in unstructured environments. *The International Journal of Robotics Research*, 24 (4):275–293, 2005.

[104] P. Renton, M. Greenspan, H.A. Elmaraghy, and H. Zghal. Plan-n-scan: A robotic system for collision-free autonomous exploration and workspace mapping. *Journal Intelligent. Robotics Systems*, 24(3):207–234, 1999.

[105] J. M. Sanchiz and R. B. Fisher. A next-best-view algorithm for 3D scene recovery with 5 degrees of freedom. In *Proc. British machine vision conference*, pages 163–172, Nottingham, UK, 1999.

[106] Y. Li, B. He, and P. Bao. Automatic view planning with self-termination in 3D object reconstructions. *Sensors and Actuators A: Physical*, 122(2):335–344, 2005.

[107] J. E. Banta, L. R. Wong, C. Dumont, and M. A. Abidi. A next-best-view system for autonomous 3-d object reconstruction. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 30(5):589–598, 2000.

[108] G. Impoco, P. Cignoni, and R. Scopigno. A six-degrees-of-freedom planning algorithm for the acquisition of complex surfaces. *International Journal of Shape Modeling*, 1(11):1–23, 2005.

[109] M. K. Reed, P. K. Allen, and I. Stamos. Automated model acquisition from range images with view planning. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 72–77, Canada, 1997.

[110] W. Stuerzlinger. Imaging all visible surfaces. In *Proc. Conference on Graphics interface*, pages 115–122, Kingston, Ontario, Canada, 1999.

[111] S. Son, S. Kim, and K.H. Lee. Path planning of multi-patched freeform surfaces for laser scanning. *The International Journal of Advanced Manufacturing Technology*, 22(5-6):424–435, 2003.

[112] R. Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999.

[113] R. Kummerle, P. Pfaff, R. Triebel, and W. Burgard. Active Monte Carlo localization in outdoor terrains using multi-level surface maps. In *Autonome Mobile Systeme 2007*, Informatik aktuell, pages 29–35. Springer, Heidelberg, 2008.

[114] S. Larsson and J. A. P. Kjellander. Path planning for laser scanning with an industrial robot. *Robotics and Autonomous Systems*, 56(7):615–624, 2008.

[115] S. Se and P. Jasiobedzki. Photo-realistic 3D model reconstruction. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 3076–3082, Orlando, 2006.

[116] Y. Yu and K. Gupta. An information theoretical approach to view planning with kinematic and geometric constraints. In *Proc, IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1948–1953, Seoul, 2001.

[117] E. Kruse, R. Gutsche, and F. M. Wahl. Efficient, iterative, sensor based 3-d map building using rating functions in configuration space. In *Proc, IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1067–1072, Minneapolis, 1996.

[118] M. Mehrandezh and K. Gupta. Simultaneous path planning and free space exploration with skin sensor. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 4, pages 3838–3843, Las Vegas, 2002.

[119] P. Wang and K. Gupta. Computing c-space entropy for view planning based on beam sensor model. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2389–2394, Lausanne, Switzerland, 2002.

[120] M. Fernandez, K. Gupta, and J. C. Fraile. Simultaneous path planning and exploration for manipulators with eye and skin sensors. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 1, pages 914–919, Las Vegas, 2003.

[121] H. H. Gonzalez-Banos, D. Hsu, and J. C. Latombe. Motion planning: Recent developments. In S.S. Ge and F.L. Lewis, editors, *Autonomous Mobile Robots: Sensing, Control, Decision-Making.* CRC Press, 2006.

[122] S. Lu and J. H. Chung. Weighted path planning based on collision detection. *Industrial Robot: An International Journal*, 32(6):477–484, 2005.

[123] B. Nemec and L. Zlajpah. Force control of redundant robots in unstructured environment. *IEEE Transactions on Industrial Electronics*, 49(1):233–240, 2002.

[124] Y. Guo and M. Balakrishnan. Complete coverage control for nonholonomic mobile robots in dynamic environments. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 1704–1709, Orlando, 2006.

[125] J. G. Juang. Application of repulsive force and genetic algorithm to multi-manipulator collision avoidance. In *Proc. 5th Asian Control Conference.*, volume 2, pages 971–976, Melbourne, Australia, 2004.

[126] J. Xu, D. K. Liu, and G. Fang. An efficient method for collision detection and distance queries in a robotic bridge maintenance system. In T. Tarn, S. Chen, and C. Zhou, editors, *Robot Welding Intelligence and Automation*, pages 71–82. Springer-Verlag, Berlin, 2007.

[127] O. Brock and O. Khatib. Real-time obstacle avoidance and motion coordination in a multi-robot workcell. In *Proc. IEEE International Symposium on Assembly and Task Planning, ISATP*, pages 274–279, Porto, Portugal, 1999.

[128] P. Chotiprayanakul, D. K. Liu, D. Wang, and G. Dissanayake. A 3-dimensional force field method for robot collision avoidance in bridge maintenance. In *Proc. 24th Intl. Symp. on Automation and Robotics in Construction*, pages 139–145, Kochin, India, 2007.

[129] M. Greenspan and N. Burtnyk. Obstacle count independent real-time collision avoidance. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 1073–1080, Minneapolis, 1996.

[130] C. C. Lin and J. H. Chuang. Potential-based path planning for robot manipulators in 3-d workspace. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 3, pages 3353–3358, Taipei, 2003.

[131] Z. Yao and K. Gupta. Self-motion graph in path planning for redundant robots along specified end-effector paths. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 2004 – 2009, Orlando, 2006.

[132] M. Stilman, J. U. Schamburek, J. Kuffner, and T. Asfour. Manipulation planning among movable obstacles. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 3327–3332, Roma, Italy, 2007.

[133] M. Garber and M. C. Lin. Constraint-based motion planning for virtual prototyping. In *Solid Modeling and Applications*, pages 257–264, Saarbrucken, Germany, 2002.

[134] D. Wang, N. M. Kwok, D. K. Liu, and Q. P. Ha. Ranked pareto particle swarm optimization for mobile robot motion planning. In *Design and Control of Intelligent Robotic Systems*, volume 177, pages 97–118. Springer-Verlag, Berlin Heidelberg, 2009.

[135] K. Kondo. Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free-space enumeration. *IEEE Journal of Robotics and Automation*, 7(3):267–277, 1991.

[136] J. Zhang. An integrated method for planning smooth collision-free trajectories for robot arms. In *Proc. of the Third Annual Conference of AI, Simulation and Planning in High Autonomy Systems, Integrating Perception, Planning and Action*, pages 46–53, Menlo Park, CA, 1992.

[137] K. Gupta and G. Zhenping. Motion planning for many degrees of freedom: sequential search with backtracking. *IEEE Transactions on Robotics and Automation*, 11(6): 897–906, 1995.

[138] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University, Oct. 1998.

[139] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd, and L. E. Kavraki. Sampling-based roadmap of trees for parallel motion planning. *Robotics, IEEE Transactions on*, 21 (4):597–608, 2005.

[140] J. J. Kuffner and S. M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 2, pages 995–1001, San Francisco, 2000.

[141] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Proc. 4th int. Workshop on the Algorithmic Foundations of Robotics*, pages 293–308, Hanover, 2000.

[142] K. Miyazawa, Y. Maeda, and T. Arai. Planning of graspless manipulation based on rapidly-exploring random trees. In *Proc. 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing*, pages 7–12, Montreal, Canada, 2005.

[143] F. F. Bernardon, J. L. D. Comba, C. A. Pagot, and C. T. Silva. Gpu-based tiled ray casting using depth peeling. *Journal of Graphics Tools*, 11(3):23–29, 2006.

[144] D. Ferguson and A. Stentz. Anytime, dynamic planning in high-dimensional search spaces. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 1310–1315, Roma, Italy, 2007.

[145] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 3856–3861, Barcelona, 2005.

[146] A. Yershova and S. M. LaValle. Improving motion-planning algorithms by efficient nearest-neighbor searching. *IEEE Transactions on Robotics*, 23(1):151–157, 2007.

[147] D. Ferguson and A. Stentz. Anytime RRTs. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5369–5375, Beijing, 2006.

[148] D. Hsu, J. Tingting, J. Reif, and S. Zheng. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *Proc. IEEE International Conference on Robotics and Automation*, volume 3, pages 4420–4426, Taipei, Taiwan, 2003.

[149] D. Hsu, J. C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *International Journal of Robotics Research*, 25(7): 627–643, 2006.

[150] H. L. Cheng, D. Hsu, J. C. Latombe, and G. Sanchez-Ante. Multi-level free-space dilation for sampling narrow passages in prm planning. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1255–1260, Orlando, 2006.

[151] M. Saha, J. C. Latombe, Y. C. Chang, and F. Prinz. Finding narrow passages with probabilistic roadmaps: The small-step retraction method. *Autonomous Robots*, 19 (3):301–319, 2005.

[152] D. Hsu, G. Sanchez-Ante, H. L. Cheng, and J. C. Latombe. Multi-level free-space dilation for sampling narrow passages in prm planning. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, pages 1255–1260, Orlando, 2006.

[153] C. Lanzoni, A. Sanchez, and R. Zapata. Sensor-based motion planning for car-like mobile robots in unknown environments. In *Proc. IEEE International Conference on Robotics and Automation, ICRA*, volume 3, pages 4258–4263, Taipei, Taiwan, 2003.

[154] L. Zhiye and C. Xiong. Path planning approach based on probabilistic roadmap for sensor based car-like robot in unknown environments. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2907–2912, The Netherlands, 2004.

[155] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[156] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[157] S. Ihara. *Information Theory for Continuous Systems.* World Scientific, Singapore, 1993.

[158] C. E. Shannon. A mathematical theory of communication. *Reprinted with corrections from The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[159] C. A. Coello Coello. An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, 2000.

[160] E. Nikolaidis, D. M. Ghiocel, and S. Singhal. *Engineering Design Reliability Handbook.* CRC Press, 2004.

[161] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer Verlag, 1999.

[162] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[163] C. T. Kelley. *Iterative Methods for Optimization.* SIAM Frontiers in Applied Mathematics. SIAM Press, Philadelphia, 1999.

[164] R. Grabowski, P. Khosla, and H. Choset. Autonomous exploration via regions of interest. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, volume 2, pages 1691–1696, Las Vegas, 2003.

[165] G. Paul and D. K. Liu. Replanning of multiple autonomous vehicles in material handling. In *Proc. IEEE International Conference on Robotics, Automation and Mechatronics, RAM*, pages 231–236, Bangkok, 2006.

[166] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–643, 2005.

[167] T. Yanghai, R. T. Collins, V. Ramesh, and T. Kanade. Bayesian color constancy for outdoor object recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, volume 1, pages 1132–1139, Kauai, HI, 2001.

[168] C. A. Coello Coello. A comprehensive survey of evolutionary-based multi-objective optimization techniques. *Knowledge and Information Systems*, 1:269–308, 1999.