GENERATION AND OPTIMIZATION OF LOCAL SHAPE DESCRIPTORS FOR POINT MATCHING IN 3-D SURFACES

by

BABAK TAATI

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
August 2009

Copyright © Babak Taati, 2009
This thesis is dedicated to my family.
Abstract

We formulate Local Shape Descriptor selection for model-based object recognition in range data as an optimization problem and offer a platform that facilitates a solution. The goal of object recognition is to identify and localize objects of interest in an image. Recognition is often performed in three phases: point matching, where correspondences are established between points on the 3-D surfaces of the models and the range image; hypothesis generation, where rough alignments are found between the image and the visible models; and pose refinement, where the accuracy of the initial alignments is improved. The overall efficiency and reliability of a recognition system is highly influenced by the effectiveness of the point matching phase. Local Shape Descriptors are used for establishing point correspondences by way of encapsulating local shape, such that similarity between two descriptors indicates geometric similarity between their respective neighbourhoods.

We present a generalized platform for constructing local shape descriptors that subsumes a large class of existing methods and allows for tuning descriptors to the geometry of specific models and to sensor characteristics. Our descriptors, termed as Variable-Dimensional Local Shape Descriptors, are constructed as multivariate observations of several local properties and are represented as histograms. The optimal set of properties, which maximizes the performance of a recognition system, depend on the geometry of the objects of interest and the noise characteristics of range image acquisition devices and
is selected through pre-processing the models and sample training images. Experimental
analysis confirms the superiority of optimized descriptors over generic ones in recognition
tasks in LIDAR and dense stereo range images.
Acknowledgments

First and foremost I wish to thank my advisor, professor Michael Greenspan. He has been a role model for me throughout my graduate studies and a few lines are too short to express my deep appreciation for having him as my supervisor. I am deeply grateful to him for his continuous guidance and steady support throughout my graduate studies. His enthusiasm kept me motivated, even when faced with challenging research problems. I benefited greatly from our regular discussions and meetings. On numerous occasions a short chat with him saved hours, days, or even weeks of my time as he pointed me to the right direction. I am also thankful to him for giving me the freedom to explore and research various aspects of my thesis and even work on research topics unrelated to my thesis. As I approach finishing my studies, I owe a great deal of my professional confidence to having worked with Dr. Greenspan. As a passionate space exploration enthusiast, I am also grateful beyond words for the opportunity to work on a project with space related applications and for arranging my internship at MDA Space Missions. I would also like to thank my thesis examiners, Professors Frank Ferrie, David Rappaport, and Geoffrey Chan for taking their time to consider my work and for their helpful comments.

I am much obliged to my labmates at the Robotics and Computer Vision laboratory who made contributions to my work through discussions, collaboration, and experimental assistance. I would particularly like to thank my good friend Limin Shang for his tremendous
help in acquiring stereo images and for letting me use his demo framework for presenting my algorithm. I am also thankful to the department staff, particularly Debra Fraser, Bernice Ison, and Steve Babcock for being supportive and helpful.

I am much indebted to Dr. Piotr Jasiobedzki, my supervisor at MDA Space Missions, for his guidance during my year of work in his R&D group. I learned a great deal during my internship and was thrilled to apply algorithms that were developed for my thesis on a space related prototype. I would also like to thank other members of the R&D group, particularly Dr. Stephen Se, Michel Bondy, Rubakumar Krishnasamy, and Jeffrey Puk who contributed to my work through their helpful suggestions.

I am sincerely grateful to professor Sven Dickinson for hosting me in his research group when I was a visitor in the Computational Vision Group at the University of Toronto. I learned a lot during my year at UofT and met many wonderful people who directly or indirectly contributed to my progress. I am especially grateful to Relu Patrascu and Stefan Mathe for helping me access Linux clusters to run some of my experiments. I am also thankful to professor Aaron Hertzmann in the Dynamic Graphics Project (GDP) lab for letting me to use their VIVID 3D scanner. I am likewise grateful to John Hancock for helping me learn how to work with the scanner and with PolyWorks.

I am thankful to all my friends who were part of my life during my graduate studies and put up with me and encouraged me during busy times. Last but not least, I am truly blessed for having a family that has given me their unconditional support throughout my life and has always pushed me to strive for the best. I am truly grateful.
Table of Contents

Abstract ii

Acknowledgments iv

Table of Contents vi

List of Tables ix

List of Figures xi

List of Symbols xiii

Chapter 1:

Introduction 1

1.1 Problem Statement and Applications 1

1.2 Point Matching 4

1.3 The Need for High Precision 5

1.4 Increasing Precision Through Tuning 7

1.5 The Prevalence of Range Data 8

1.6 Contributions 10

1.7 Organization of Thesis 11
List of Tables

3.1 LSD Generation Pseudo-Code. .................................................. 44
3.2 Several Descriptors Expressed in VD-LSD Formulation .................. 45

5.1 Selected Subsets with FFS-ACP for Angel and Big Bird Using Various Numbers of Training Images. ................................................. 64
5.2 Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 100 RANSAC Iterations (with Nene-Nayar Search). ............................. 68
5.3 Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 500 RANSAC Iterations (with Nene-Nayar Search). ............................. 69
5.4 Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 100 RANSAC Iterations (without Nene-Nayar Search). ............................. 69
5.5 Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 500 RANSAC Iterations (without Nene-Nayar Search). ............................. 69
5.6 Gnome Model Recognition Rates (Percentage) without Nene-Nayar Search in LIDAR Cluttered Scenes with 500 RANSAC Iterations, with Various Numbers of Cluster Centres for Vector Quantization, and Exact and Approximate k-d tree Bin Assignment. ............................................ 70
5.7 Average Processing Times (ms) for Recognizing Big Bird Using Vector Quantized VD-LSD in Cluttered LIDAR Range Images. ............... 71
5.8 Recognition Rates (Percentage) in Stereo Cluttered Scenes with 1,000 RANSAC Iterations (with Nene-Nayar Search). . . . . . . . . . . . . . . . . . . . . . 75

5.9 Recognition Rates (Percentage) in Stereo Cluttered Scenes with 5,000 RANSAC Iterations (with Nene-Nayar Search). . . . . . . . . . . . . . . . . . . . . . 75

5.10 RADARSAT Recognition Rates (Percentage) in LIDAR Scenes with 500 and 100 RANSAC Iterations (Without Nene-Nayar Search). . . . . . . . . . 80
## List of Figures

1.1 Example of Object Recognition in Range Data .......................... 2

1.2 Model-Based Object Recognition in Range Data, Performed in Three Phases. 3

1.3 Required Number of RANSAC Iterations for Various Assurance Levels, Plotted Against Precision. .......................... 6

2.1 Spin Image Cylindrical Coordinates $\alpha$ and $\beta$. ...................... 15

2.2 Sample Spin Images of Two Points on the Head (top) and the Tail (bottom) of a Dinosaur Model. .......................... 17

3.1 A Sample Image Point Cloud of an Object (Angel), Two Selected Points (on the Face and Lower Arm of the Statue), Neighbourhood ($N_r$) of the Sample Points with $r = 1.5cm$ (Shadowed), and the PCS Frames ($\vec{i}, \vec{j}, \vec{k}$) Associated with them. .......................... 37

3.2 Ambiguity in the PCS Frame. ................................................. 39

3.3 Median Angle Between the $\vec{k}$ Axes and the Ground Truth Surface Normal for a Sample Point Cloud as a Function of PCS Radius $r$. .......................... 50

3.4 Normalized Histograms of $DCG$ Values Corresponding to Two Subsets and the Difficulty in Comparing Them. .......................... 55

4.1 Point Matching with Optimal LSDs for Model-Based Object Recognition. ................................................. 57
5.1 3-D Models Used in the Experiments (From Left to Right, Queen’s Models: Angel, Big Bird, Gnome, Kid, and Zoe; UWA Models: Chef, Chicken, Parasaurolophus, and T-Rex; and MDA Space Mission model RADARSAT). 62

5.2 (a) Sample LIDAR Range Image that Contains Angel, Big Bird, Kid, and Zoe (b) the Models Highlighted as Found by Our Algorithm. 68

5.3 Precision Levels for Detecting Three Models in Sample Images Using Scalar Quantized and Vector Quantized VD-LSD and Spin Images, Plotted Against $s_t$. 72

5.4 (a) Sample Stereo Range Image that Contains Angle, Gnome, and Zoe (b) the Models Highlighted as Found by Our Algorithm. 76

5.5 (a) Sample Point Cloud that Contains UWA Models Chef, Chicken, Parasaurolophus, and T-Rex (b) the Models Highlighted as Found by Our Algorithm. 77

5.6 The Experimental Setup at MDA Space Missions for RADARSAT Pose Acquisition and Tracking Experiments. 78

5.7 (a) Sample LIDAR Range Image of RADARSAT (b) the Acquired Pose as Found by Our Algorithm. 79

5.8 (a) Sample Stereo Range Image of RADARSAT (b) the Acquired Pose as Found by Our Algorithm. 79

5.9 RADARSAT Grapple Fixture. 80

5.10 Two Sample ($N_r$) Neighbourhoods (shadowed) on the RADARSAT Model with $r = 15cm$, and Axes $\vec{i}$ and $\vec{j}$ of the PCS Frames for Each Neighbourhood. 82
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSD</td>
<td>Local Shape Descriptor</td>
</tr>
<tr>
<td>$p_r$</td>
<td>precision</td>
</tr>
<tr>
<td>$a$</td>
<td>RANSAC assurance level</td>
</tr>
<tr>
<td>$m$</td>
<td>number of models</td>
</tr>
<tr>
<td>$M$</td>
<td>model point cloud</td>
</tr>
<tr>
<td>$n$</td>
<td>number of points in a model point cloud</td>
</tr>
<tr>
<td>$p$</td>
<td>model point</td>
</tr>
<tr>
<td>$q$</td>
<td>scene point</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>the set of real numbers</td>
</tr>
<tr>
<td>$T$</td>
<td>homogeneous alignment transformation</td>
</tr>
<tr>
<td>$\chi(.)$</td>
<td>descriptor of a point</td>
</tr>
<tr>
<td>$d(.)$</td>
<td>distance function defined in the space of LSDs</td>
</tr>
<tr>
<td>$\xi$</td>
<td>distance threshold in the 3-D space</td>
</tr>
<tr>
<td>$I_1$</td>
<td>maximum value in a column of the confusion matrix</td>
</tr>
<tr>
<td>$I_2$</td>
<td>second largest value in a column of the confusion matrix</td>
</tr>
<tr>
<td>$s$</td>
<td>match ambiguity measure</td>
</tr>
<tr>
<td>$s_t$</td>
<td>match ambiguity threshold</td>
</tr>
<tr>
<td>TP</td>
<td>true positive</td>
</tr>
</tbody>
</table>
FP       false positive
FN       false negative
S        property map
$N_r(p)$ local neighbourhood of point $p$, with radius $r$
$A_s$    support angle
$C$      covariance matrix
$O: (\hat{i}, \hat{j}, \hat{k})$ principal component space frame associated to each point
$R = [\hat{i}, \hat{j}, \hat{k}]$ rotation matrix associated to each point
PCS      Principal Component Space
$r$      PCS neighbourhood radius
$R$      LSD neighbourhood radius
$F_P$    set of position properties
$F_D$    set of direction properties
$F_R$    set of dispersion properties
$F$      set of all properties
$\vec{n}$ surface normal
$
{|x|, |y|, z}$ basic position properties
{$X_a, Y_a, Z_a, D}$ extended position properties
{$|\cos(\theta)|, |\cos(\phi)|,$
$\cos(\psi), \hat{\theta}, \hat{\phi}, \psi}$ direction properties, axes angles
{$\hat{roll}, |pitch|, |yaw|$} dispersion properties, $ZYX$ Euler angles
{$\bar{\alpha}, \bar{\beta}, \bar{\gamma}$} dispersion properties, $ZZY$ Euler angles
{$e_1, e_2, e_3$} basic dispersion properties
{$\tilde{e}_1, \tilde{e}_2, \tilde{e}_3$} normalized dispersion properties
A subset of $F$.

Dimensionality of a subset.

Property map corresponding to subset $f$.

VD-LSD of a point.

Lower threshold for salient region selection.

Higher threshold for salient region selection.

Variable-Dimensional Local Shapd Descriptor.

Genetic algorithms.

Simulated annealing.

Forward feature selection.

Subtract operator between property subsets.

Property subset outperform symbol.

Unambiguous Correct Percentage.

All Correct Percentage.

Discounted Cumulative Gain.

Wilcoxon Signed Rank.

Scalar quantization.

Vector quantization.

Nene-Nayyar search parameter.
Chapter 1

Introduction

1.1 Problem Statement and Applications

Model-based object recognition in range data involves the detection and localization of 3-D models in range images. Given a set of 3-D models and a range image, detection is defined as identifying the visible models, and localization is defined as finding the 3-D rigid transformations that align the visible models with the image. This is a difficult problem and solving it, particularly in real-time, has been researched for several years [6, 43, 49].

Range images are similar to regular images, except that for each pixel the depth value, i.e. the distance to the range sensor, is also available. Pixel intensity information are also sometimes available in addition to depth values. Figure 1.1(a) illustrates a complete 3-D model represented as a point cloud. Figure 1.1(b) shows a point cloud, extracted from a range image, in which the model is partially visible. Less than 19% of the surface of the model is visible in the image and the rest is occluded by the other objects or by itself (i.e. self-occluded). Furthermore, less than 16% of the surface area visible in the scene belongs to the shown model. Figure 1.1(c) shows the model which has been automatically
identified and localized in the image, despite these relatively high levels of scene clutter and occlusion.

A 3-D rigid transformation has three positional and three rotational degrees of freedom and exhaustive search through this 6-D pose space is infeasible. A large class of techniques aim at efficiently solving this problem without requiring exhaustive search by following a three-phase scheme consisting of point matching, hypothesis generation, and pose refinement [7, 27, 29, 34, 55, 65, 74, 78, 114, 116, 119, 131, 132].

In the first phase, tentative matches are established between several points on the image and their corresponding points on models by comparing the local shapes of various regions of the two data sets. Since the output of the first phase often contains some incorrect matches (i.e. outliers), a statistically robust algorithm such as RANSAC [32] or the Generalized Hough Transform [9], is then utilized in the second phase to generate and verify rigid transformations that align visible models with the image. Finally, in the third phase, the accuracy of the recovered alignments are improved using a refinement algorithm. Figure 1.2 illustrates the block diagram of these three phases.
We emphasize the difference between the aforementioned general scheme for aligning range data, and pose refinement algorithms utilized in the final phase. In the general case, the relative 6-D pose between 3-D surfaces is completely unknown and needs to be determined. On the other hand, pose refinement algorithms, also known as registration refinement or registration algorithms for short, aim at improving the accuracy of the alignment between two (or more) range data sets if an initial rough estimate of the relative pose is known. The Iterative Closest Point (ICP) [15] algorithm and its many variants (e.g. [41, 56]) are among such pairwise refinement methods.

In the absence of an initial alignment, or if the initial estimate is too inaccurate, refinement methods often get trapped in local minima in the relative pose space and fail. The first two phases of the general range data localization scheme are therefore necessary in order to provide an initial pose estimate that lies within the global minimum potential well of the refinement algorithm.

We also differentiate between object recognition and pose determination. Object recognition refers to the general case where there are multiple models in the database. Pose determination refers to localizing a single known object within a range image, and defines an easier problem that object recognition.

While the primary focus of the research presented here has been on object recognition...
in range data, we note that efficient alignment of 3-D surfaces is fundamental for a variety of other applications including tracking [104, 120], 3-D model building and multi-view reconstruction [48, 79, 118], robotic operations such as bin picking and industrial inspection [16, 18, 38], 3-D modeling and localization of mobile robots [10, 101], augmented reality [39], and medical imaging and computer assisted surgery [70, 73].

1.2 Point Matching

The focus of this research is on the first phase of the aforementioned three-phase process, i.e. point matching. *Local Descriptors* have been widely used for point matching both in 2-D intensity images [11, 68] and in 3-D range data [7, 27, 29, 34, 55, 78, 116, 119, 131, 132]. In range data, *Local Shape Descriptors (LSDs)* encapsulate local geometrical variations such that similarity between two LSDs signifies geometric similarity between their respective local neighbourhoods and suggests a point match. During online recognition, LSDs are computed for several interest points on the image and are matched with model LSDs which are computed offline.

Ideally, LSDs should be highly descriptive so that their similarity is commensurate with the geometric similarity of the local neighbourhoods they represent. LSDs should also be robust with respect to reasonable levels of noise, changes in viewing angle, occlusion and self-occlusion, clutter, changes in resolution, and other non-ideal conditions that may arise. Furthermore, they should ideally be efficient to compute and compare, so that point matching can be performed fast, particularly during the online stage.
1.3 The Need for High Precision

Point correspondences established by LSD matching often contain a large percentage of incorrect matches, also known as false positives or outliers. *Precision*, defined as the percentage of detected matches that are correct (i.e. inliers), is an important performance measure of a point matching algorithm. Increasing the precision will not only improve the effectiveness of a method at recognizing a larger number of more varied objects, but it can also improve upon the efficiency of the transformation generation and verification phase. For instance, if RANSAC [32] is used in the second phase, a smaller number of iterations will be required if precision is higher.

RANSAC can handle large levels of outlier point matches (*false positives*) with a sufficiently high number of iterations. However, simple combinatorics shows that its computational cost, i.e. the required number of iterations for achieving a specified assurance level, strongly depends on precision. The assurance level [32] indicates the chance of finding the correct pose by drawing a correct group match from the candidate list of matched points. For instance, the assurance level of 99% indicates that if at least one correct group match exists in the candidate list, the chance of missing it is less than 1%. Figure 1.3 illustrates the required number of RANSAC iterations plotted versus precision levels ($p_r$), such that a certain assurance level ($a$) is achieved. Appendix A provides a brief overview of the RANSAC algorithm and the derivations for establishing the relationship between $p_r$, $a$, and the number of iterations.

As the figure illustrates, low precision levels require a high number of iterations. Equivalently, with the same number of iterations, higher assurance levels are obtained when the precision is higher. For instance, with $p_r = 15\%$, 200 iterations achieves $a = 49.1\%$, where as with $p_r = 30\%$, the same number of iterations guarantees that the correct pose will be
found with an assurance of 99.6%.

Enhancing the point matching phase to increase precision thus translates to either requiring fewer RANSAC iterations for the same level of assurance, or perhaps more importantly, higher assurance levels obtained with the same number of iterations. A higher assurance level is equivalent to higher recognition rates as it indicates lower chances of overlooking the correct alignment.

Therefore, for applications where high reliability in recognition and localization are important, e.g. in estimating and tracking the pose of a satellite for unmanned rendezvous and docking [53, 120], it is highly desirable to have a point matching method that provides very high levels of precision. Furthermore, so long as the computational cost of enhancing precision levels is not too large, overall computational efficiency is achieved with fewer
RANSAC iterations.

1.4 Increasing Precision Through Tuning

In this work, we aim at increasing the precision levels of point matching. We formulate LSD selection as an optimization problem to maximize precision and investigate solutions that tune LSDs to the geometry of specific models. Our method utilizes a set of invariant properties [118–120] to formulate a generalized LSD generation platform, from which the most appropriate subset for online recognition is selected through offline processing of target models via a feature selection routine.

The idea of tuning object recognition algorithms to specific models is in fact not new. In [17, 50] for instance, automatic programming was used to select model features that were useful for recognition tasks. Such methods, however, became less popular with the advent of Local Descriptors, as efforts were focused mostly on devising LSDs that were compact and low dimensional, typically 1-D, 2-D or 3-D. This minimalistic approach was often justified by citing the need to have compact descriptors to reduce the computational and memory cost of point matching [29, 55, 132]. In many cases however, the preference for low-dimensional descriptors was so entrenched as to remain unstated and unjustified [7, 116]. In the work presented here, rather than manually designing low dimensional LSDs, the LSDs are automatically generated and may be higher (up to 9) dimensional.

Our formulation demonstrates two important trade-offs in the identification and localization process. The first is the trade-off between offline and online processing costs. Our formulation allows pre-processing of the models in order to enhance online performance. The second trade-off is between the computational cost of the first online phase, i.e. point
CHAPTER 1. INTRODUCTION

matching, versus that of the second online phase, i.e. RANSAC, as well as the overall reliability. Our formulation allows investing more time in point matching in order to gain computational efficiency in the hypothesis generation phase, and more importantly, in order to increase object recognition rates. We also previously showed that a correct choice of LSD can increase the effectiveness of point correspondences in pair-wise range image alignment, thereby increasing the possible acquisition angle between overlapping images \[118\].

1.5 The Prevalence of Range Data

Non-contact range sensors could be classified into two main categories of active and passive. Active sensors project a beam or a pattern of light or laser and use the reflection to determine depth values. Methods of inferring depth in active sensors include triangulation, measuring the time of flight, and structured light. Passive sensors, notably stereo camera pairs, receive the ambient light from multiple sensors and use triangulation for calculating depth.

During the past decade range image acquisition devices have become more readily available. Advances in LIDAR technology (e.g. Time-of-Flight and Flash LIDAR) have translated to faster and more accurate range image acquisition at lower costs. While the trade-off among acquisition time, accuracy, and cost persists, several commercially available LIDAR scanners \[1,24,51,82,86,87,93,105,133\] offer choices on the efficient frontier of the trade-off space, ranging from real-time acquisition \[51\], to very cost efficient \[87\], to integrated scanning and modeling systems \[105\], and to highly specialized systems, e.g. for space or military applications \[86\].

Progress in dense stereo and multi-view reconstruction algorithms has also enabled
CHAPTER 1. INTRODUCTION

more accurate range data from stereo processing. Whereas early dense stereo range image acquisition systems and algorithms tended to require very long processing times and produced poor range data (due to a variety of factors, such as non-Lambertian reflection and inaccuracies in camera calibration), recent advances have enabled both faster and more accurate stereo range data. On one hand, binocular and trinocular calibrated camera heads continue to thrive [94], some providing hardware accelerated stereo processing units enabling close to video-rate range image acquisition [95]. On the other hand, multi-view stereo algorithms have become more advanced and can now reliably provide accurate 3-D data, up to a scale value, based on a set of uncalibrated images or a video sequence. A typical such systems consists of interest point matching (e.g. [11, 68, 81]), extracting and calibrating camera parameters through applying robust Structure-From-Motion algorithms on the matched points [46, 110], Bundle Adjustment [122] to refine the estimated camera positions (e.g. [36, 67]), and multi-view dense range data generation (e.g. [37]). The required processing time for accurate multi-view dense stereo from uncalibrated images remains high and, therefore, these approaches are not yet suitable for real-time or near real-time object sensing. They do however provide a powerful tool for 3-D modeling [126], which is an offline phase required for model-based object recognition. Faster processing, particularly with the aid of hardware acceleration, is also possible. Such hardware acceleration can be achieved through utilizing the parallel processing capabilities of Graphics Processing Units (GPU) (e.g. [109, 130]) or custom designed FPGA boards (e.g. [95]).

With the continuing trend in increased availability, lower prices, and higher accuracies in range data acquisition systems, range data will only become more prevalent in the coming years and the need for applications, systems, and algorithms processing range data, such as those explored in this thesis, will likely continue to gain importance.
1.6 Contributions

The most important contribution of this work is a generalized platform for LSD construction based on a set of local properties. The properties are computed using the principal component space of small neighbourhoods and are derived such that they are invariant with respect to rigid transformations and are robust with respect to common non-ideal conditions such as partial occlusion, self-occlusion, noise, and clutter.

We tune our LSDs for each model (or each group of models) by selecting their optimal (or near-optimal) set of properties. This involves offline pre-computation on the models which is aimed at easing the online computation performed on the scene data and increasing the overall reliability. This platform leads to a set of LSDs with variable dimensionality which subsume a large class of previously developed descriptors. Furthermore, it provides a framework for systematic selection of LSDs which in turn enables optimizing LSDs to the geometry of objects of interest by way of mapping the optimization problem into a more manageable feature selection problem. Several methods are experimentally investigated for solving the feature selection problem.

The platform also introduces the concept of maximalist descriptors, where descriptive-ness and robustness of LSDs are emphasized rather than their compactness. This is in contrast with many of the previously developed LSDs which take a minimalist approach by trying to construct low dimensional and compact descriptors. These potentially high dimensional (up to 9-D) descriptors are the key to more effective point correspondence without sacrificing either computational or memory efficiency, although there is a tradeoff.

For experimental analysis, recognition tests were performed on real range images of several objects, acquired with both LIDAR and stereovision sensors. The images included high levels of occlusion and scene clutter and typical levels of noise associated with range
acquisition devices. Stereo range images in particular included high levels of noise. Results confirmed that descriptors tuned to model geometry achieve higher precision levels, and often drastically outperformed recognition rates of generic descriptors such as Spin Images with similar processing times. To the best of our knowledge, the reported recognition rates in noisy stereo data are unprecedented, particularly considering that, for reasons explained later in Section 3.5, the method uses only the 3-D shape information and discards all the appearance, i.e. colour or intensity, information available in a stereo range image. Through collaboration with MDA Space Missions, a space robotics company in Brampton, Ontario, the proposed method is utilized in a fully-functioning prototype for real-time satellite pose acquisition and tracking.

Our optimization platform enables tuning the descriptors to the shape of individual models or batch training, i.e. optimizing descriptors based on the geometry of a group of models. The experiments reported in this thesis explore both possibilities and emphasize the circumstances appropriate for either choice. The reported results also include experimental overfitting analysis, empirical selection of the number of cluster centres for vector quantization, and comparison between exact and approximate k-d trees.

### 1.7 Organization of Thesis

The remainder of this thesis is organized as follows: Chapter 2 reviews the previous work on object recognition in range data, focusing on the methods that apply the three-phase scheme described earlier. Historical and alternative approaches are reviewed in Section 2.1. Local Shape Descriptors are reviewed in Section 2.2. Hypothesis generation and pose refinement techniques are reviewed in Section 2.3. For the sake of completeness, Global Shape Descriptors are briefly reviewed in Section 2.4. The chapter concludes in Section
2.5 with some observations based on the preceding literature review. These observations provide the basis for the novel method presented in this thesis.

The theoretical aspects of the work are covered in Chapter 3. Section 3.1 sets the notations and Section 3.2 discusses the match establishment process. A formal problem statement aimed at achieving high precision levels is presented in Section 3.3. Invariant properties are introduced in Section 3.5 and their invariance levels are examined. In Section 3.6, Local Shape Descriptor selection is formulated as an optimization problem which can be solved using feature selection from the list of invariant properties. Section 3.7 explains how our formulation is a generalization of a large group existing methods and discusses some of the benefits of applying this formulation. Three feature selection methods and three evaluation measures are explored in Section 3.8 for finding a solution through an offline training phase.

The details of our object recognition system based on the optimized descriptors are explained in Chapter 4. Extensive experiments with a variety of objects and under various conditions are performed to verify the effectiveness of the method and the results are presented in Chapter 5. Conclusions and future work are discussed in Chapter 6.
Chapter 2

Previous Work

2.1 Historical and Alternative Approaches

While this research concentrates on methods that follow the three-phase pattern introduced in Section 1.1, it should be noted that alternative range data alignment techniques also exist. Historically, early methods such as the Interpretation Tree [43, 44], Alignment [49], Geometric Hashing (GH) [60], and Geometric Probing [42] do not follow this three phase scheme. Some of these methods (e.g. GH) could be utilized in the second phase (hypothesis generation) in the presence of an efficient and effective point matching technique.

Among more recent approaches that do not follow the three phase pattern are Potential Well Space Embedding (PWSE) [103] and the 4-Points Congruent Sets (4PCS) [4]. PWSE exploits the distribution of the local minima in the ICP [15] relative pose space for recognizing and locating objects in range data, particularly in cases where segmentation of foreground and background is possible. The 4PCS takes an exhaustive RANSAC approach, but utilizes a special property of congruent sets to improve upon the computational complexity. We note that some of these alternative approaches, e.g. 4PCS, could also benefit
CHAPTER 2. PREVIOUS WORK

from effective point matching.

2.2 Local Shape Descriptors

Johnson and Hebert [55] construct their LSDs, known as Spin Images, as 2-D histograms. Spin Images [55] are discussed in detail in this review of Local Shape Descriptors for two reasons: first, because they are perhaps the most well known and one of the most widely used LSDs; second, because they are used as a comparison base in the experimental Section of this thesis (Chapter 5).

The term Spin Image signifies the fact that the resulting histogram is equivalent with representing the neighbouring points in a 2-D frame that is rotationally invariant around the interest point about the surface normal. Interest points, for which LSDs are computed, are selected close to edges so that areas with high geometric variation are selected and flat areas, which convey minimal geometric information, are avoided. Points lying exactly on edges are discarded since they lead to degenerate tangent and normal computations.

In order to construct the Spin Image of an interest point, the distance to the tangent plane ($\beta$) and the perpendicular distance to the normal line ($\alpha$) are computed for all the neighbouring points within a certain Support Distance ($D_s$). This is equivalent with forming a cylindrical coordinate system to each interest point, computing the cylindrical coordinates of its neighbours, keeping the distance coordinates while discarding the angle coordinate. This process associates a list of 2-D points in the ($\alpha$, $\beta$) coordinate frame to each interest point. Figure 2.1 illustrates a region in a 3-D point cloud, an interest point ($p$), the surface normal and the tangent plane going through $p$, and the $\alpha$ and $\beta$ properties of a neighbouring point ($p'$). A $15 \times 15$ histogram of this list is smoothed, to account for noise, and is
used as the LSD. Histograms are normalized so that the encapsulation is robust with respect to mesh or point cloud density. Finding the model Spin Image that is most similar to a scene Spin Image is equivalent to a nearest neighbour search in the high dimensional (i.e. $15 \times 15 = 225$-D) space of Spin Images. Efficient nearest neighbour search is performed using an algorithm presented in [85] which outperforms k-d tree structures [14] for neighbour finding in high dimensional spaces.

Spin Images propose using the correlation coefficient as a similarity metric and use the $L_2$ norm of the difference vector of normalized histograms to measure the correlation coefficient. Applying the $L_2$ norm enables direct use of the Nene-Nayar [85] nearest neighbour search for histogram matching without any alteration. It also facilitates the use of PCA compression as it guarantees the best compression method in terms of generating the minimal error in the $L_2$ norm.

Support Angles ($A_s$) are introduced for constructing Spin Images for model points in order to limit the effect of self-occlusion and increase robustness with respect to the viewing angle. Points within the neighbourhood whose surface normals form large angles (e.g. $> 60^\circ$) with the surface normal of the interest point do not contribute to the Spin Image of the interest point since they are likely to not be visible, due to self-occlusion, in range
images where the interest point is visible. To ease the computational burden of Spin Image comparison, PCA-based compression is investigated to reduce the size of descriptors. As expected, computational efficiency is gained at the expense of reduced recognition performance.

The original Spin Image publication operates on surface meshes and takes nearby vertices (i.e., within $D_s$) as contributing neighbours. Later improvements introduce uniform sampling of each mesh face to account for non-uniform tessellations. While the algorithm is designed to operate on surface meshes, it could easily be extended to operate on point clouds since the tangent plane and the surface normal are relatively easy to estimate [83].

Figure 2.2 illustrates a 3-D model, two selected points on the model, and the Spin Images of those two points (without Support Angle consideration). Large Support Distance, indicated by the darker region on the model, and large number of bins along each dimension are used in the construction of these sample Spin Images for better visualization.

*Pairwise Geometric Histograms* [7] are constructed very similarly to Spin Images, except that the 2-D histograms are based on two properties termed $d$ and $\alpha$. Term $d$ is identical to $\beta$ in the Spin Image notation. Term $\alpha$ denotes the angle between the surface normal of a neighbouring point and the interest point. The Bhattacharyya Distance [30, 33] is used as the distance metric for comparing histograms.

*Point Signatures* [29] are constructed for each interest point as a 1-D array based on the distance profile of the intersection of a sphere, centred at the interest point, with the object from the tangent plane. *Point Fingerprints* [116] are based on computing multiple Point Signatures at various radii. The geodesic distance is used instead of the Euclidean distance to account for possible ambiguities that could arise from multiple intersecting paths between a sphere and the surface. The name evokes the resemblance between the
Figure 2.2: Sample Spin Images of Two Points on the Head (top) and the Tail (bottom) of a Dinosaur Model.

intersection of multiple spheres and the surface with fingerprints.

The idea of Spin Images is extended differently in Textured Spin-Images [21], where 3-D histograms are computed instead of 2-D Spin Image histograms, the added dimension being colour (or intensity). However, since colour is not invariant with respect to lighting, the addition reduces the robustness of descriptors and is applicable only in near ideal conditions. Only experimental results with synthetic range views are reported. Furthermore, many range sensing devices do not provide colour information and solely output 3-D point clouds.

Another modification of Spin Images is presented as Spin Images with Spherical Parameterization [54], where instead of assigning a cylindrical coordinate system to each interest point, a local spherical coordinate system is formed. It is argued that in cases where surface
normals are not available or cannot be estimated with good accuracy, spherical coordinates provide a more stable descriptor. As in the original Spin Images, only two coordinates of the neighbouring points are kept while dropping the third.

Tensor-Based Correspondence [78, 79] offers another LSD computed as three dimensional tensors. The tensors are used both for model building and for object recognition in cluttered scenes. A major contribution of this algorithm is in storing the model tensors in a hash table to achieve efficient online matching between image tensors and their closest model tensor. By applying this hashing technique, recognition time becomes independent of the number of models in the database.

More recently, Local Surface Patches [27, 28] are another LSD constructed as 2-D histograms of two properties of neighbouring points: the angle between the surface normal and the surface normal of the interest point (as in [7]), and the Shape Index. The Shape Index is defined as a measure of local surface curvature. Similar to Spin Images, support angles and bilinear interpolation are used to reduce the effects of self-occlusion and noise, respectively. The Shape Index measure is also used for selecting interest points for LSD construction by discarding areas with low shape variations. Similar to the aforementioned Tensor-Based Correspondence method [78], a hashing scheme is used to store all model LSDs in order to ease the computational burden of matching image LSDs to their nearest model LSD. Hash keys are computed based on the mean and variance of Shape Index values over each surface patch. Local Surface Patches are used in [28] for building a recognition system capable of distinguishing highly similar objects. An embedding algorithm is used to map the descriptor histograms into a low-dimensional space for efficient storage and matching. This is equivalent with applying vector quantization, rather than scalar quantization, for representing the descriptor histograms.
CHAPTER 2. PREVIOUS WORK

Other LSDs include *Splash* [111], *Surface Signatures* [132], *Statistical Matrices* [131] (similar to Spin Images with Spherical Parameterization), *Regional Point Descriptors* [34], *Spherical Spin Images* [96], *Symbolic Signatures* [97], and *Harmonic Shape Images* [134]. A review of several LSD techniques is presented in [77]. An earlier review of a broader class of range data alignment techniques is offered in [23].

Since the focus of this research has been on object recognition in range data, a review of object recognition methods in 2-D images is beyond the scope of this thesis. Many such techniques utilize descriptors encoding local appearance information. For the sake of completeness, and also to illustrate a point of similarity between shape and appearance descriptors, a very brief introduction to appearance descriptors is included here.

Local Descriptors in 2-D images, for point matching across intensity (or colour) images, include SIFT [68], SURF [11], GLOH [81], CSIFT [2], and an extension of Spin Images to 2-D intensity images [61]. Similar to the 3-D case, the process of generating descriptors in 2-D images could be divided into two main steps of interest point selection and descriptor construction for the selected interest points. Earlier methods typically placed interest points, also known as keypoints, on local geometric image features such as edges and corners [45, 99, 107]. Later methods often use affine invariant [76], scale-invariant [11, 68], or both affine and scale invariant interest points [80]. A survey of local invariant interest point selection methods is presented in [123].

In SIFT descriptors, scale invariance in keypoint selection is achieved by way of detecting the extrema in the scale space. Keypoints lying on edge pixels are further discarded in SIFT descriptors since their local neighbourhoods are deemed more likely to be viewpoint dependent than those not lying on edges. Orientation invariance is achieved through assigning an invariant direction to each keypoint. For instance, the orientation is aligned...
with the dominant gradient direction in SIFT descriptors. Descriptors are then constructed by way of encapsulating the local image properties around each interest point, typically in a histogram format. Local neighbourhood size is determined by the scale of each detected keypoint. In SIFT features, the local image around each keypoint is encapsulated in an orientation histogram storing the local gradient values.

Other 2-D descriptors are not reviewed here as our focus has been on point matching in range data. A recent survey and experimental performance analysis of several local descriptors in 2-D images is presented in [81].

2.3 Hypothesis Generation and Pose Refinement

In the three-phase process of range data alignment using LSDs, the list of candidate matches provided by the first phase is used in the second phase to hypothesize alignments which are later accepted (or rejected) during a verification phase. RANSAC, Generalized Hough Transform (GHT) [9], and Pose Clustering [89, 112] are among algorithms that could be utilized in this phase.

A comparison study [102] concludes that the choice between RANSAC and GHT depends on the percentage of outliers among matched points. If the list of tentative point matches contains very high levels of outliers (e.g. precision $< 3 - 4\%$), then GHT is preferable. On the other hand, if the list contains at least a minimum level of inliers (e.g. precision $\geq 4 - 5\%$) then RANSAC is more suited. As a result, in the presence of distinctive descriptors, resulting in good or even moderately good point matching, RANSAC is often the better choice.

In RANSAC, over several iterations, groups of point matches are randomly drawn to generate candidate poses. Since the minimum number of required 3-D point matches for
computing a rigid transformation is three, group matches must contain at least three point
matches. Each hypothesized alignment is then evaluated and RANSAC terminates success-
fully as soon as an alignment is verified. The algorithm aborts without finding an alignment
if none is verified after a computational threshold is reached, e.g. after a certain number of
iterations. Further details regarding RANSAC, as well as alternatives to 3-point RANSAC,
are discussed in Appendix A.

Hypothesized alignments could be verified by various means, such as counting the num-
ber of points that overlap between the image and the aligned model. In the simplest case,
the verification step could return the candidate pose that aligns the largest overlap area,
if that area is larger than a certain pre-defined threshold. The threshold could be defined
relative to the areas of the models. For instance, estimated poses could be discarded if the
aligned area constitutes less than 10% of the area of the model. However, this threshold
automatically leads to unsuccessful detection and localization of models that are more than
90% occluded. Alternative and more complicated verification methods could check for in-
consistencies in occluding surfaces, active regions of range sensing devices, etc. Some of
these more advanced techniques are exploited in [78].

In the absence of any candidate point matches, random assignments could be used as
the input of the hypothesis generation phase. This however, severely limits the applicability
of the method since the combinatorial explosion of possible point matches renders the
process too computationally costly even for cases where the number of points in the model
or image clouds is as low as about one hundred points. In [63] for instance, finding the
alignment between point clouds containing a mere 100 points takes over 2.5 minutes and
processing clouds with 200 points takes over 18 minutes. Rigidity constraints could be
used in order to contain the combinatorial explosion and reduce the number of possible
cases by rejecting inconsistent group matches. Since the alignment between the image and the model is rigid, group matches used for estimating this rigid transformation must conform with the constraints of rigidity; that is, preservation of length and angles. For example, in 3-point RANSAC, the three points in each point cloud form a triangle and the two triangles should be equal. To account for noise and slightly inaccurate matches, small deviations are often tolerated. Since rigidity constraints are powerful in rejecting incorrect group matches, they are often used even in the presence of a relatively reliable candidate point matches list. An example of combining rigidity constraints with RANSAC without utilizing a point matching phase is the technique presented in [22].

Pose refinement, also known as registration refinement, is the final phase of the three-phase identification and localization scheme. In the context of object recognition, pose refinement aims at reducing the error in the alignment found by the hypothesis generation phase. Pose refinement is also utilized in several interactive applications where a rough alignment between 3-D surfaces can be initialized by user input, for example by providing a few manually selected point matches.

The most well-known pose refinement technique is the Iterative Closest Point (ICP) algorithm [15], which reduces the error over several iterations. At each iteration, each point on one surface is matched to its nearest neighbour on the other surface according to the current alignment. An adjusting alignment is then computed using the established matches via a least squares estimation. The relative pose between the two surfaces is then adjusted accordingly before the algorithm proceeds to the next iteration. The ICP is proven to converge to a local minimum in the relative pose space. Convergence to the global minimum however is not guaranteed and only occurs if the initial alignment is within a small region, known as the potential well, of the correct alignment.
A strength of the ICP is its flexibility, as the algorithm can be applied in any dimensionality and for a variety of transformation types. For instance, whereas in refining the relative pose of range data the algorithm works with 3-D points and rigid transformations, it could easily be applied to 2-D points and affine or perspective transformations.

A review of some ICP variants is offered in [75, 98], and [40, 84] offer alternative pose refinement algorithms. A 3-D bundle adjustment algorithm is presented in [48] that refines the overall alignment among multiple range images for 3-D modeling applications.

2.4 Global Shape Descriptors

The difference between Local Shape Descriptors and Global Shape Descriptors (GSDs) is that the latter aims at encapsulating the entire shape of an object and is suitable for applications such as similarity-based 3-D model retrieval, as in [5, 91, 125]. Since the focus of our work has been on LSDs, the literature survey has mostly reviewed LSD methods. For the sake of completeness, some major GSDs are also included here.

Osada et al. [91] devised a simple yet effective method, termed Shape Distribution, for encapsulating arbitrary shapes as a probability distribution. A Shape Function is used to compute a value based on the distances between randomly drawn pairs or triplets of points, or the angles between the lines connecting them. A 1-D histogram of these values is formed to represent the shape of the object. In order to find a good Shape Function, several functions are examined and the best one is empirically chosen. The chosen function simply computes the distance between a pair of randomly drawn points. The resulting histogram is invariant to rigid transformation and could become scale invariant by a simple normalization.
For articulated objects, an extension of the Shape Function computes the geodesic distance instead of the Euclidean distance to achieve configuration invariance so long as the change in configuration does not change the topology of the object [20]. Particular care is required to achieve some level of invariance with respect to changes in the topology [19].

Ip et al. [52] have extended the Shape Function such that when computing the distance between a pair of points, the condition of the line connecting the points is also considered. The suggested Shape Function discriminates between pairs based on whether the connecting line lies fully within the model, partially inside the model, or fully outside the model. Separate histograms are formed for each case and a weighted average of the resulting histograms is used as the shape representation. The weights are based on the percentage of occurrence of each type of these connecting lines. The method is applied in implementing a query engine for performing similarity based retrieval. However, the merits of discriminating between pairs based on their connecting lines are dubious. Apart from the added computational cost, the method loses generality since it can no longer operate on objects presented as point clouds or degenerate surface models. More importantly, as evident in the reported experimental results, the performance improvement over the original Shape Distribution method [91] is small. Intuitively, the weighted average histograms would resemble the original single histogram without consideration of the condition of the connecting line between random points.

Liu et al. [66] have developed the Directional Histogram Model, another extension of Shape Distribution which exploits hardware-accelerated computation of orthographic projections on GPUs to achieve time efficiency. Each model is viewed from different viewing angles and its thickness is sampled through rays projecting its closest and furthest points
to the viewing plane. The thickness values are used to form (normalized) thickness histograms that count the number of occurrences of different thickness ranges. The viewing direction can be represented by two angles in spherical coordinates and these two angles are used to decompose the thickness histograms into their spherical harmonics representations. Spherical harmonics are invariant to translation and scale (since thickness histograms are normalized), but are not rotation invariant. The coefficients from the spherical harmonics decomposition of various thickness levels over a limited bandwidth are combined to form a rotation invariant representation of a 3-D model named the \textit{Matrix Descriptor}. The Matrix Descriptors are normalized again and are used to compare the similarity of different 3-D models using a simple \(L_2\) metric.

Kazhdan et al. \cite{58} have devised another method, the \textit{Rotation Invariant Spherical Harmonic Representation}, that utilizes the spherical harmonics representation. It is noted that the performance of several shape representation and matching methods can be improved if a good rotation invariant representation of objects is available. The suggested method achieves orientation invariance by discarding the orientation (i.e. phase) information in the spherical harmonics representation. Frome et al. have extended the idea of spherical harmonics descriptors to construct their LSDs, known as the \textit{Regional Point Descriptors} \cite{34}.

Wahl et al. \cite{129} developed their method, the \textit{Surflet-Pair-Relation Histograms}, for encapsulating global shape information in a histogram. Their method is similar to the Shape Distribution method of Osada in that it constructs a histogram based on pairs of sampled points on the surface. However, the methods differ in that Wahl’s requires surface normals along with the 3-D coordinate for each point. Furthermore, instead of a 1-D array, a four dimensional histogram is constructed based on four parameters (three angles and one distance) that are calculated based on the relative location of the sampled points. To cope
CHAPTER 2. PREVIOUS WORK

with the large 4-D space, the parameter space is quantized coarsely into five intervals along each direction resulting in a histogram with a total of 625 bins.

The recognition method of Hetzel et al. [47] has an offline training phase, during which three separate histograms are constructed for each object from different viewing angles. The three histograms are based on pixel depth, surface normals, and surface curvature. During recognition, these three histograms are constructed for the scene data and the similarity between scene and model histograms are used for object matching.

More recently, Akgul et al. [5] construct their shape descriptors as multivariate samples of local features over the surface of an object to represent an underlying probability density function for each object. The method, termed the Density-Based Framework, then applies a nonparametric statistical technique to estimate the probability density functions and to match new multivariate samples, from query objects, to those of the objects in the database. Six local surface features are introduced as potential candidates to be computed for object surface points. Any subset of these local features defines a possible multivariate shape descriptors. The introduction of different local features resembles the platform presented in this thesis. However, the method falls short of investigating all possible feature subsets ($2^6$ in total) to determine the best combinations and opts for handpicking three subsets. The experimental comparison between competing descriptors is somewhat similar to another recent model retrieval study [35], in which the performance of three different shape descriptors, all based on spherical harmonics, is examined in order to experimentally determine the most suited in a given 3-D model retrieval task.

In the category of 2-D Global Descriptors, the Shape Context approach, by Belongie et al. [13], represents the global shape as a 2-D log-polar histogram. Several points are first selected on the contour of a given 2-D shape. For each selected edge point, the magnitude
and the angle of the vector connecting it to all the remaining points are computed and contribute to the histogram. Scale invariance is obtained by normalizing all radial distances by the mean (or median) distance between all the point pairs in the shape. Rotation invariance, if desired, is obtained by measuring the vector angles relative to the direction of the surface normal of the interest points. The Shape Context approach could also be utilized as a Local Descriptor, as in [81]. Kortgen et al. [59] have extended the idea of Shape Context for encapsulating 3-D shape and have utilized the resulting GSD in shape retrieval and matching.

Carlin [25] presents a comparative study of several GSDs in shape recognition and retrieval applications. The methods are experimentally evaluated according to various performance metrics based on visual perception, retrieval context, and shape representation. A recent review of many GSDs is available in [121].

2.5 Discussion

As can be observed from this concise review of various descriptors, there are some similarities between global and local descriptors and also between descriptors in 3-D data and in 2-D images. For instance, a large class of descriptors encapsulate the distribution of properties, whether local or global, in the form of a histogram. This characteristic is fully investigated and exploited in this research, as detailed in Chapter 3.

From reviewing the LSD methods, a marked separation between interest point selection, descriptor construction, and descriptor matching could also be observed. Since descriptor generation is often relatively expensive and seldom necessary for all points on the input surfaces, a point selection mechanism is often utilized to sub-sample both the models and the scene data into interest points, ideally such that the selected points lie on salient
and distinctive regions [108]. Descriptors are then constructed only for the selected interest points.

Descriptor matching is often equivalent with statistical comparison of sample distributions, typically represented as histograms. Any histogram similarity or distance measure, such as Histogram Intersection [117] (equivalent with the $L_1$ distance for normalized histograms), Bhattacharyya Distance [30], or Quadratic Distance\footnote{Quadratic Distance [88] is a histogram comparison metric that, unlike most other comparison metrics, is designed to take the similarity of neighbouring bins [113] into account. The Quadratic Distance could be interpreted as a form of the Mahalanobis Distance [72].} [88] could be used to this end.

Alternatively, instead of representing descriptors as histograms, statistical methods could applied for direct comparison of distribution samples. Nonparametric statistical techniques [106] are the preferred approach since they presume few assumptions on the underlying distributions. For instance, the Wilcoxon Rank Sum and the Moses Rank-Like tests are used in [33] as an alternative to histogram matching. The same study also presents an experimental analysis of various histogram comparison metrics.

Finally, by reviewing some of the most recent GSDs developed for model retrieval, e.g. [5,35], one could expect the resurgence of efforts to select descriptors based on the specifics of the problem at hand. For instance, in both cited methods multiple descriptors, three in each case, are explored in order to experimentally identify the most suited. Furthermore while it is shown that the performance of the proposed descriptor in [5] is matched by that of a competitor technique [128], it is also shown that the performance of the two methods is somewhat complementary. That is, the set of models retrieved well by the two methods do not fully overlap. This leaves room for investigating the type of descriptors most suited for the retrieval of different types of models.

In the case of LSDs, as mentioned earlier in Section 1.4, tuning of object recognition
methods to the specifics of models at hand went out of favour with the advent of powerful local descriptors. This was mostly due to the fact that local descriptors allowed handling of problems that were hitherto unsolvable in a computationally efficient manner. Nevertheless, as generic descriptors and the frontiers of object recognition and computational power are continuously challenged by the need for reliable real-time operation, handling of noisy stereo images, etc., we expect that the appetite for special purpose descriptors, e.g. those optimized to the set of target models or to range sensor characteristics, will only grow in the coming years. The framework presented in this thesis is in line with this growing requirement.
Chapter 3

Variable-Dimensional Local Shape Descriptors

In this chapter, we formally define the goal of a detection and localization algorithm and formulate the problem of selecting the best LSD for point matching as an optimization problem. We present a platform for constructing descriptors based on a set of local properties. The platform allows for solving the optimization problem through an offline pre-processing phase.

and [108]. Since we will only be dealing with local descriptors in range data, the terms descriptor and LSD are used interchangeably throughout the text.

3.1 Definitions and Notation

The set of models consists of $m$ point clouds $\{M_1, \ldots, M_m\}$, with each cloud $M_j$ containing a set of $n_j$ points $\{p^j_1, \ldots, p^j_{n_j}\}$, where $p \in \mathbb{R}^3$. The range image, e.g. as acquired with LIDAR or dense stereo, is transformed into a point cloud $\{q_1, \ldots, q_s\}$ and may contain
one or more of the target models, subject to partial occlusion, self-occlusion, scene and background clutter, and noise. Furthermore, the image point cloud might also include missing data (known as holes) in some regions, e.g. textureless areas in the stereovision case or dark matte surfaces and highly specular or reflective surfaces for LIDAR. For each model $M_j$ visible in the image, the true transformation that aligns it with the image is a homogeneous matrix $T_j$. That is, $T_j M_j = \{T_j p^j_1, ..., T_j p^j_{n_j}\}$ is the aligned model with the image. (By abuse of notation, $p^j_i$ now denotes a homogeneous coordinate.) The goal of object recognition is to detect the presence of each model $M_j$ that is visible in the image (detection), and determine $T_j$ for each visible $M_j$ (localization).

Let $\chi(\cdot)$ denote the descriptor of a point, e.g. $\chi(q_k)$ and $\chi(p^j_i)$ are the LSDs of the $k^{th}$ image point and the $i^{th}$ point of the $j^{th}$ model respectively. Further, let $d(\cdot)$ be the distance function defined in the space of LSDs. We assume that distance is a continuous function and is normalized such that it lies in the $[0, 1]$ interval, where 0 indicates complete similarity.

A good LSD should be both discriminating and robust with respect to noise, clutter, (self-)occlusion, and changes in viewing angle, i.e. the distance metric should return 0 (or a low value) for all true correspondences and 1 (or a high value) for all other point pairs. For a particular model $M_j$, we therefore want to select the LSD function $\chi(\cdot)$ and the distance metric $d(\cdot)$ such that the value of $d\left(\chi(p^j_i), \chi(q_k)\right)$ is low when $p^j_i$ is the true correspondent of $q_k$, and high otherwise. True correspondence signifies that the Euclidean distance of the image point $q_k$ and the aligned model point $T_j p^j_i$ is smaller than a threshold value $\xi$, i.e. $\|q_k - T_j p^j_i\| < \xi$. Here $\|\cdot\|$ represents the $L_2$ norm in 3-D Cartesian space, and $\xi$ is commensurate with the noise and resolution of the sensor and the model.
3.2 Match Establishment

A confusion matrix can be formed by computing the distance between each model LSD with each image LSD. Rows and columns of this matrix correspond with model and image LSDs respectively. There are various ways for selecting candidate point matches from a confusion matrix. One way is to select the top model descriptor for each image descriptor by finding the maximum value of each column. Another is to select point correspondences that result from unambiguous LSD matches. The latter method is often superior as it could account for the fact that some image LSDs do not correspond to any model LSDs, e.g. those that correspond to scene clutter.

The rigorous method for selecting such unambiguous matches is to search for columns that have a maximum that is sufficiently larger than the rest of the values in the column, such that it stands out in a statistically meaningful sense [90]. Since this is computationally costly, a simple approximation could be used that only requires comparing the maximum value \( I_1 \) with the second largest value \( I_2 \) [68, 118]. A match is then established between model and image LSDs only if \( I_1 \) is sufficiently larger than \( I_2 \), i.e., if \( s = (I_1 - I_2) / I_1 \) is greater than a predetermined threshold value \( s_t \). For computational efficiency, it is not necessary to compute the confusion matrix in its entirety and at most finding the two nearest neighbours of each LSD is sufficient. Efficient nearest neighbour finding for descriptor matching could be attained via either a hashing and voting mechanism, as in [79, 118], a k-d tree search, or the Nene-Nayar algorithm [85] for search in high-dimensional spaces, as in [55].
CHAPTER 3. VARIABLE-DIMENSIONAL LOCAL SHAPE DESCRIPTORS

3.3 Formal Problem Statement

Whichever method is used for hypothesizing point matches, the generated list of tentative matches contains some correct matches, i.e. true positives ($TP$), and some incorrect matches, i.e. false positives ($FP$). Computational efficiency of the RANSAC phase of registration depends on precision, defined as $TP/(TP + FP)$. We formulate the problem of finding the best Local Shape Descriptor function $\chi(.)$ for recognizing a particular target object as the following optimization problem:

$$\arg\max_{\chi} \left( \frac{TP}{TP + FP} \right)$$

(3.1)

Formula (3.1) aims at maximizing precision, but it imposes no requirements to obtain high values of recall, defined as $TP/(TP + FN)$, where $FN$ indicates false negatives, i.e. correct matches that are not detected. As a result, it might create a bias for selecting a very small number of correct matches while neglecting other true matches that exist in the confusion matrix.

While obtaining high precision at the expense of recall might be undesirable in many applications, it is exactly the desired situation in our particular application. For example, consider the case when the outcome of the matching phase is exactly three point matches which are all correct. This is in fact the best scenario for RANSAC since it would compute the correct pose in a single iteration. On the other hand, more iterations would be needed if recall was high but precision was lower. In practice, due to uncertainties, we often opt for accepting more than the minimum required three matches. Adjusting the $s_t$ threshold allows for setting the number of matches that are returned by the point matching phase. Typically returning 20-40 matches is adequate.
Although Formula (3.1) captures the essence of the problem, it is too general to facilitate a solution, as $\chi(\cdot)$ could be any descriptor function. As such, it is desirable to express it in a simpler form where a solution, or at least an approximate solution, could be found. In this work, we use a set of invariant properties [119] to lay out a solvable expression of this formula.

To this end, we first present a scheme for constructing LSDs which is a generalization for a large class of descriptors, such as [7, 55]. The formulation is such that it allows utilizing a feature selection routine for automatic selection of the optimal descriptor for each model. We distinguish \textit{automatic selection} from \textit{intuitive selection} in that while the former does not require any user interpretation, the latter requires a combination of user observation or insight, manual adjustment, or the use of ad hoc and heuristic measures.

### 3.4 Property Map

The LSD for each interest point aims at encapsulating the local geometry surrounding the point by way of recording some invariant properties of all the points in its vicinity. These properties define the relationship between the neighbouring points, and the interest point and should be invariant to in-plane rotations, and relatively robust with respect to the viewing angle, noise, occlusion, and self-occlusion.

Formally, $S$, defined as:

$$S : \mathbb{R}^3 \rightarrow \mathbb{R}^L$$  \hspace{1cm} (3.2)

is a mapping from the 3-D Cartesian coordinates of points to an $L$-D space of properties, and is used to compute $L$ properties for each neighbour point.
If the mapping were ideal, then the distribution of properties in the $L$-$D$ space would be unique for each interest point and would convey the local shape information. Rigorous comparison of image and model distributions can be performed by applying multi-variate distribution equality statistical tests. Since the normalcy assumption, which is assumed in most parametric statistical tests, does not hold for the underlying distributions, nonparametric tests, such as the multivariate Kolmogorov-Smirnov test [57, 92], are better suited for this task. Due to the high computational costs of such approaches, histograms provide a more attractive method for representing and comparing sample distributions. Any histogramming scheme, such as scalar quantization with uniform or non-uniform binning, or vector quantization with various numbers of cluster centres could be used to this end. Any histogram similarity measure, such as those discussed in Section 2.5, could be used as the distance function $d(.)$.

In Spin Images [55], the mapping $S$, known as the spin-map, converts the 3-D Cartesian coordinates of each neighbour point to the 2-D Spin Image coordinates of $\alpha$ and $\beta$, as defined in Section 2.2. Likewise, Pairwise Geometric Histograms [7] map the 3-D coordinates onto a different 2-D space, denoted with $\alpha$ and $d$.

Properties of all the neighbouring points, or a histogram of those properties, convey the local shape information effectively only if the mapping $S$ is selected properly. The proposed formulation is a generalization of a large class of LSDs as it allows for optimizing the descriptors by rewriting Formula (3.1) as follows:

$$\argmax_S \left( \frac{TP}{TP + FP} \right)$$

(3.3)
3.5 Invariant Properties

To generate invariant properties, first a small local neighbourhood is formed around each point. If the point cloud \( M \) is expressed in an arbitrary frame \( O \), and \( p \in M \), then the local neighbourhood around \( p \) is defined as \( N_r(p) = \{ \forall p' \in M \mid ||p - p'|| \leq r \} \), where \( r \) is the neighbourhood radius. To reduce the effect of self-occlusion, the neighbourhood formation process for model point clouds could be augmented with support angle \((A_s)\) considerations [55].

Each neighbourhood \( N_r(p) \) contains a set of 3-D points, with a \( 3 \times 3 \) covariance matrix \( C(p) = \frac{1}{|\{N_r(p)\}|} \left[ (N_r(p) - \bar{N}_r(p))(N_r(p) - \bar{N}_r(p))^T \right] \). Here \( |\{N_r(p)\}| \) denotes the cardinality of the set \( N_r(p) \), i.e. the number of points in \( N_r(p) \); and \( \bar{N}_r(p) \) is the centre of mass of \( N_r(p) \), computed by averaging the coordinates of all the points in the neighbourhood. By abuse of notation, the subtract operator in the \( N_r(p) - \bar{N}_r(p) \) expression denotes subtracting the centre of mass coordinates from each of the points in the neighbourhood.

Eigenvalue decomposition of \( C(p) \) associates an orthonormal frame \((\vec{i}, \vec{j}, \vec{k})\) to each point \( p \) and three Eigenvalue scalars \((e_1 \geq e_2 \geq e_3)\), which indicate the dispersion of \( N_r(p) \) along each axis of the frame. The eigenvalues are non-negative and the eigenvectors are real, since covariance matrices are always symmetric positive semi-definite. The frame, known as the Principal Component Space (PCS), is treated as a Cartesian coordinate frame \( O_p \) centered at \( p \) and corresponds with a \( 3 \times 3 \) rotation matrix \( R_p = [\vec{i} \mid \vec{j} \mid \vec{k}] \). Figure 3.1 illustrates a sample range image, two sample points and their neighbourhoods (shadowed), and the PCS frames of each of the selected points.

When generating a descriptor for interest point \( p \in M \) we refer to its PCS frame as the reference frame. The geometric properties of each neighbour point \( p' \in N_R(p) \) define the relationship between \( O_{p'} \) and \( O_p \). Here the \( N_R(p) \) neighbourhood is defined the same
as $N_r(p)$, albeit with a different, and typically larger, value of radius $R$. We emphasize that descriptor neighbourhood ($N_R$) and PCS neighbourhood ($N_r$) are not necessarily identical since $R$ and $r$ do not have to be equal. The properties consist of position properties ($F_P$), direction properties ($F_D$), and dispersion properties ($F_R$). We denote the entire set of properties as $F \triangleq F_P \cup F_D \cup F_R$.

The coordinates of each neighbouring point expressed in the reference frame $O_p$ form three basic position properties. The basic position properties of point $p'$ with respect to interest point $p$ are computed as $[x, y, z]^T = R_p^T (p' - p)$. Several other position properties could be computed by combining these basic properties, e.g. $Y_a = \sqrt{z^2 + x^2}$ equivalent to the perpendicular distance to the $\vec{j}$ axis.

The rotation that registers $O_{p'}$ with $O_p$, i.e. $R_{p,p'} = R_p^T R_{p'}$, can be expressed with three parameters which constitute the direction properties. The rotation can be represented in various forms and therefore it is possible to generate a variety of equivalent direction properties such as the ZYX Euler angles ($roll$, $pitch$, and $yaw$) or the ZYZ Euler angles ($\alpha$, $\beta$, and $\gamma$) [100] (see appendix B for more details). The inner products of the corresponding axes between the two frames form further direction properties, as $\cos(\theta) = \vec{i} \cdot \vec{j}$,
cos(\phi) = \vec{j} \cdot \vec{j}', and cos(\psi) = \vec{k} \cdot \vec{k}', where \( R'_{p} = [\vec{j}' \mid \vec{j} \mid \vec{k}'] \).

Eigenvalues \((e_1, e_2, \text{ and } e_3)\) associated with each neighbour point \( p' \in N_R(p) \) form the three basic dispersion properties. Combinations of these values, or their normalized versions, form alternative dispersion properties.

Some of the aforementioned properties suffer from an ambiguity in estimating the frame of each point. We alter those properties to cancel the effect of this ambiguity. When computing the eigenvalue decomposition of \( C \), the eigenvectors have an ambiguity in their signs. In other words, if \( \vec{v} \) is an eigenvector of \( C \), i.e. \( C \vec{v} = \lambda \vec{v} \), then the flipped vector \( -\vec{v} \) is also an eigenvector of \( C \). This results in 3 degrees of ambiguity in computing the rotation matrix \( R_p \) for frame \( O_p \), as either one of the \( \vec{i}, \vec{j}, \text{ or } \vec{k} \) eigenvectors could flip.

If the value of \( r \) is small, then \( \vec{k} \) approximates the surface normal at each point. We exploit this fact to resolve the ambiguity in the direction of \( \vec{k} \) when the input data is available in range image (as opposed to point cloud) format. When the viewing angle \( \vec{w} \) is known, as is often the case with modern sensor systems, then fixing the direction of \( \vec{k} \) is achieved by ensuring the inequality \( \vec{k} \cdot \vec{w} < 0 \). For model point clouds, if the cloud is generated by merging several images from different viewpoints, such as in \([79, 118]\), then the viewing angle is known for each point and the same technique could be applied. If the model cloud is generated from dense sampling of a CAD model, then surface normals are available for each point. Since \( \vec{k} \) approximates the normal vector \( \vec{n} \), ensuring \( \vec{k} \cdot \vec{n} > 0 \) guarantees uniqueness. In either case, if the inequality does not hold, we simply flip \( \vec{k} \), i.e. use \( -\vec{k} \) instead. The second ambiguity is resolved by ensuring the right-handedness of the frame, i.e. \( \vec{i} \cdot (\vec{j} \times \vec{k}) > 0 \). To this end, the direction of \( \vec{j} \) is flipped if the inequality does not hold.

There still remains one further ambiguity that cannot be resolved. If \( O_p \) is the frame of point \( p \) with rotation matrix \( R_p = [\vec{i} \mid \vec{j} \mid \vec{k}] \), then rotation matrix \( \bar{R}_p = [-\vec{i} \mid -\vec{j} \mid \vec{k}] \)
also defines a valid frame $\bar{O}_p$. In other words, there is an ambiguity in the sign of vectors $\vec{i}$ and $\vec{j}$. Figure 3.2 illustrates this ambiguity in a ridge local neighbourhood of a sample point cloud. As a result, properties that depend on the direction of these vectors cannot be reliably computed. For instance, property $x$ is the projection of $(p' - p)$ on $\vec{i}$ and as such it is impossible to specify the sign of $x$.

Since there is no theoretical method of resolving this last ambiguity, we simply remove the need for doing so by altering the properties so that they do not depend on the sign of vectors $\vec{i}$ and $\vec{j}$. For instance, instead of properties $x$ and $y$, we simply use $|x|$ and $|y|$. Direction properties $\cos(\theta)$, $\cos(\phi)$, roll, pitch, yaw, $\alpha$, and $\gamma$ are also affected by the ambiguity and are therefore altered as explained below. Appendix B provides a more in-depth explanation of this issue as well as the proof that the altered properties are invariant with respect to a possible flip in the directions of $\vec{i}$ and $\vec{j}$.

The complete list of properties used in this work contains seven position properties, twelve direction properties, and six dispersion properties as follows:
\[
F_P = \{ |x|, |y|, z, X_a, Y_a, Z_a, D \}
\]
\[
F_D = \{ |\cos(\theta)|, |\cos(\phi)|, \cos(\psi), \tilde{\theta}, \tilde{\phi}, \psi, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{roll}, |\text{pitch}|, |\text{yaw}| \}
\]
\[
F_R = \{ e_1, e_2, e_3, \tilde{e}_1, \tilde{e}_2, \tilde{e}_3 \}
\]

(3.4)

where
\[
X_a = \sqrt{y^2 + z^2}, \quad Y_a = \sqrt{x^2 + z^2}, \quad Z_a = \sqrt{x^2 + y^2}, \quad D = \sqrt{x^2 + y^2 + z^2},
\]
\[
\tilde{\theta} = \cos^{-1}(|\cos(\theta)|), \quad \tilde{\phi} = \cos^{-1}(|\cos(\phi)|),
\]
\[
\tilde{\text{roll}} = \begin{cases} 
\text{roll} & \text{roll} \geq 0 \\
\text{roll} + \pi & \text{roll} < 0
\end{cases}, \quad \tilde{\alpha} = \begin{cases} 
\alpha & \alpha \geq 0 \\
\alpha + \pi & \alpha < 0
\end{cases}, \quad \tilde{\gamma} = \begin{cases} 
\gamma & \gamma \geq 0 \\
\gamma + \pi & \gamma < 0
\end{cases}, \quad \text{and}
\]
\[
\tilde{e}_k = \frac{e_k}{e_1 + e_2 + e_3}, \quad k = 1, 2, 3.
\]

The advantage of including normalized dispersion properties \((\tilde{e}_1, \tilde{e}_2, \tilde{e}_3)\) in \(F_R\) is two-fold. First, they are intuitively appealing, corresponding with percentage dispersion along each principal axis. In particular, \(\tilde{e}_3\) approximates the change in geometric curvature at each point [8]. Second, during the online computation of image LSDs, a simple thresholding on the value of \(\tilde{e}_3\) can filter out points on low curvature areas at no extra computational cost. Consequently, LSDs are generated only for salient points while planar or nearly planar surfaces are discarded since they contain little geometric significance. Very high curvature areas might also not be suited for LSD placement since accurate PCS frames or
surface normals cannot be computed for them. This approach for interest point selection is somewhat similar to shape index filtering used in [27] since shape index is also a curvature measure.

In this work, we have achieved good results with simple thresholding based on the $e_3$ value. However, it is possible to use more advanced techniques. For models, the optimization formulation could be extended to include model interest point selection for LSD placement. Model pre-processing could also be applied for selecting highly distinctive regions in the models as in [108]. Enhanced scene interest point selection is also possible by algorithms such as the 3-D Cueing [26]. In 3-D Cueing, based on pre-processing the model database, each scene image is passed through a fast pre-processing step that filters out points that are deemed likely to be from scene clutter. LSDs could then be placed on areas which are thought to have a higher probability of belonging to one of the models.

It is possible to contemplate a limitless number of properties, for instance by combining those listed in Equation (3.4) with each other (e.g. $\text{atan}(x/y)$ or $e_3/e_2$). It is also possible to use higher moments for generating new properties as ours are solely based on the second moments calculated in covariance matrices. Shape Index, as in [27], colour or texture, and other curvature measures offer other possibilities. We restricted our properties to those in $F$ because, while this list is not too large in size, it is a good representative of a large class of possible properties. Furthermore, although many range acquisition devices can collect radiometric information registered with range data, we have avoided the temptation to use colour or intensity properties due to low invariance levels of colour and intensity information with respect to lighting and viewpoint changes. As a result, in this work, our descriptors are purely shape-based.
Our formulation lends itself well to extensions when other invariant properties are available in particular circumstances. The property list could also be adjusted for specific applications based on computational considerations. In Geographical Information System (GIS) applications for instance, the number of points in each cloud is typically very large (e.g. several millions). So PCS computation for each point might be inefficient. In these situations, the property list might be altered so that only position properties \( (F_p) \) are used since they only require PCS frame computation for interest points. In many such applications, range sensing devices also collect surface normal information. In such cases, direction properties that depend on the surface normal (e.g. \( \psi \)) could be added back to the list.

Finally, we note that the PCS frame is not unique for several types of points and another form of ambiguity could arise when (at least) two eigenvalues are approximately equal. Our feature selection routine addresses this issue by automatically selecting for properties that are not ambiguous, depending on the geometry of the models.

### 3.6 Local Shape Descriptor Optimization

Each subset \( f = \{f_1, \ldots, f_L\} \subset F \) is one possible manifestation of \( S \), denoted by \( S_f \), which defines a mapping from \( \mathbb{R}^3 \) to \( \mathbb{R}^L \). Properties \( f_1, \ldots, f_L \) could be computed for each point \( p' \in N_R(p) \) to form \( f_1(p'), \ldots, f_L(p') \), equivalent with coordinates of a point in an \( L \)-D space. That is, the 3-D Cartesian coordinate of \( p' \) is converted to the \( L \)-D coordinate of \( S_f(p') \). The ensemble of \( L \)-D points corresponding to all the neighbours of \( p \), i.e. \( \{S_f(p') \mid \forall p' \in N_R(p)\} \), is represented as an \( L \)-D histogram to form the LSD of point \( p \) as \( \chi_v(p) \).

When \( L \) is sufficiently large, e.g. \( L \geq 4 \), scalar quantization could lead to very large histograms, even when each axis is partitioned into a small number of bins. For instance, when
$L = 6$, if each axis is coarsely divided into 10 bins, the total number of bins will equal $10^6$. The majority of these bins will be empty since the neighbourhood size will typically be far smaller than the total number of bins. In such cases, alternative histogramming techniques such as vector quantization more compactly represent the underlying distribution of the set of high-dimensional points, and avoid the computational and memory costs of large histograms. Vector quantization is a common method for mapping high-dimensional data into a low-dimensional space such that the distance relationships are maximally preserved [28]. Another possibility is to store non-empty bins of (either vector or scalar quantized) histograms in a hash table [78, 118].

Descriptor $\chi_v(p)$ is termed a **Variable-Dimensional LSD**, or **VD-LSD**, since the dimensionality $L$ can vary depending on the chosen property subset. The VD-LSD of a point, $\chi_v(p)$, is a function of $R$, $r$, and $f$. Table 3.1 presents the pseudo-code that summarizes the entire process for VD-LSD construction. Low and high thresholds on $\bar{e}$ for salient point selection are denoted with $\varepsilon_1$ and $\varepsilon_2$ respectively. Interest points are selected by random subsampling of salient points.

We can now re-write the optimization problem expressed in Formula (3.1) and (3.3) into a more tangible form as:

$$\arg\max_{R,r,f}(TP + TP)$$

where the goal is now to set PCS radius $r$, neighbourhood radius $R$, and select the optimal subset $f \subset F$. Here $TP$ and $FP$ are functions of $R$, $r$, and $f$. 

∀ p ∈ pointCloud :

\[ N = N_r(p) = \{ \forall q \in \text{pointCloud} \mid ||p - q|| \leq r \} \]

\[ C = \text{covariance}(N) \]

\[ \text{eigenDecom}(C) \rightarrow \text{PCSFrame}(p) \]

\[ \{(i, j, k) \} \]

\[ \{ (e_1, e_2, e_3) \} \]

\[ F = F_P \cup F_D \cup F_R \] , \[ f = \{ f_1, ..., f_L \} \subset F \] , \[ S_f : \mathbb{R}^3 \rightarrow \mathbb{R}^L \]

\[ \text{salientPointCloud} = \{ \forall q \in \text{pointCloud} \mid \varepsilon_1 \leq \bar{e}_3(q) \leq \varepsilon_2 \} \]

\[ \text{interestPoints} \subset \text{salientPointCloud} \]

∀ p ∈ interestPoints :

\[ \text{propertyList}(p) = \{ S_f(q) \mid \forall q \in N_R(p) \} \]

\[ \chi_v(p) = \text{histogram}(\text{propertyList}(p)) \]

Table 3.1: LSD Generation Pseudo-Code.

### 3.7 Generalization and Optimal Characteristics

The elegance of the VD-LSD formulation is that it offers a generalized platform that subsumes a large class of descriptors. For instance, Spin Images [55] are equivalent to 2-D VD-LSD based on the \{z, Z_a\} properties, and Pairwise Geometric Histograms [7] are equivalent to a 2-D VD-LSD based on the \{z, \psi\} properties. Spin Images with Spherical Parameterization [54] (and Statistical Matrices [131]) are similar to Spin Images, except they parameterize neighbouring points in a spherical coordinate system rather than a cylindrical one. They are equivalent to 2-D VD-LSD based on \{D, \nu\} properties, where \( \tan(\nu) = \frac{Z_a}{Z_u} \). Likewise, Textured Spin-Images [21] are 3-D descriptors based on properties \{z, Z_a, I\}, where \( I \) represents the intensity (or colour) of each neighbouring point. Surface Signatures [132] are 2-D descriptors based on point distance and the angle between surface normals, equivalent to
CHAPTER 3. VARIABLE-DIMENSIONAL LOCAL SHAPE DESCRIPTORS

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Dimensionality $(L)$</th>
<th>Properties $(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spin Images [55]</td>
<td>2</td>
<td>${z, Z_\alpha}$</td>
</tr>
<tr>
<td>Pairwise Geometric Histograms [7]</td>
<td>2</td>
<td>${z, \psi}$</td>
</tr>
<tr>
<td>Spin Images with Spherical Parameterization [54]</td>
<td>2</td>
<td>${D, \nu}$</td>
</tr>
<tr>
<td>Textured Spin-Images [21]</td>
<td>3</td>
<td>${z, Z_\alpha, I}$</td>
</tr>
<tr>
<td>Surface Signatures [132]</td>
<td>2</td>
<td>${D, \psi}$</td>
</tr>
<tr>
<td>Local Surface Patches [27]</td>
<td>2</td>
<td>${\psi, S_I}$</td>
</tr>
<tr>
<td>Tensor-Based Correspondence [78]</td>
<td>3</td>
<td>${x', y', z}$</td>
</tr>
<tr>
<td>Point Signatures [29]</td>
<td>1</td>
<td>${z}$</td>
</tr>
<tr>
<td>Point Fingerprint [116]</td>
<td>2</td>
<td>${z, D'}$</td>
</tr>
</tbody>
</table>

Table 3.2: Several Descriptors Expressed in VD-LSD Formulation

VD-LSD with $\{D, \psi\}$ properties. Local Surface Patches [27] are 2-D descriptors based on $\{\psi, S_I\}$ properties, where $S_I$ denotes the Shape Index. Tensor-Based Correspondence [78] are similarly constructed based on $\{x', y', z\}$ properties, where $x'$ and $y'$ are position properties which are not used in our experiments, but could be computed by rotating the $x$ and $y$ properties about the $\vec{k}$ axis. Finally, Point Signatures [29] and Point Fingerprints [116] are constructed as arrays of point distances from the tangent plane around the intersection of a sphere, or a set of spheres, with the image surface. They therefore are based on properties $\{z\}$ and $\{z, D'\}$ respectively, albeit not represented in the form of a histogram. Here $D'$ denotes the geodesic point distance (as opposed to the Euclidean point distance $D$) since geodesic distance is used in Point Fingerprints to avoid the possible ambiguities in intersecting a sphere with a concave surface. Table 3.2 presents a summary of this information and expresses the dimensionality and the properties of several existing LSDs in the VD-LSD formulation. As it can be seen, all of these intuitively designed LSDs are of low dimensionality, i.e. $L \leq 3$.

Properties in $F$ carry various levels of information and offer different levels of robustness and discriminative power. Researchers have traditionally constructed their LSDs with
ad hoc and intuitive selection of properties. Spin Images for instance, based on the \{z, Z_a\} properties, were inspired by the correct observation that often assigning a 2-D coordinate system to a point in a cloud can be done with sufficient accuracy, whereas assigning a unique and accurate 3-D frame to each point is not always possible. This is due to an inherent ambiguity that arises in the PCS frames when at least two of the eigenvalues are approximately equal. This occurs most commonly in flat areas, but could also arise in other cases, for example around the vertex of a right cone when viewed from the top.

In a typical model or scene range image, many surfaces have low or moderate levels of curvature and their PCS neighbourhoods \(N_r\) will approximate a set of points on a circle or a slightly warped circle. The PCS frame associated with a perfect circle aligns the \(\vec{k}\) axis accurately with the normal vector, but the second and the third eigenvalues are equal and the \(\vec{i}\) and \(\vec{j}\) axes could rotate freely about the normal in the circle plane. Therefore, for such nearly circular neighbourhoods, even if the surface were slightly curved and the circle was warped, the \(\vec{i}\) and \(\vec{j}\) axes could not be reliably extracted with the same level of accuracy as the \(\vec{k}\).

In [55], it is noted that 2-D Spin Image coordinates \(\{\alpha, \beta\}\) (equivalent with \(\{Z_a, z\}\) in our notation) under-define the 3-D location of neighbour points, but carry enough descriptive power that their histogram is successful in matching at least a portion of points correctly. Such intuition is based on correct insight, and generic descriptors such as Spin Images or Tensor-based descriptors [78] indeed provide a good choice for cases where pre-processing is not possible. However, for tasks where the pre-processing of models is possible, systematic subset selection offers a superior alternative, as one could optimize the descriptor based on the geometry of the target objects and significantly improve on the point matching performance.
Our systematic approach capitalizes on the crucial observation that the robustness and descriptive power of various properties depends on the geometry of target objects. To illustrate this fact, let us consider the sample case of assigning a 2-D or a 3-D frame to each point, as was discussed for Spin Images. For some neighbourhoods, for example those resembling long cylindrical shapes, it is indeed possible to associate a 3-D Cartesian frame to each point with sufficient accuracy and without ambiguity, as the $\vec{i}$ axis will align along the length of the cylinder, $\vec{k}$ will remain parallel to the surface normal, and $\vec{j}$ will be perpendicular to both $\vec{i}$ and $\vec{k}$, crossing the cylinder. The selected point on the lower arm of the Angel model in Figure 3.1 illustrates this case as the arm resembles a cylinder. Considering this fact, for objects that are constructed from many cylindrical (or similarly shaped) parts, e.g. robotic manipulators, it is possible to assign a 3-D frame to many of their points. Similarly, it could be argued that some other properties that at first might appear less robust or discriminating, might in fact be well suited for detecting some objects with special shapes.

As opposed to intuitive selection, we generate a large set of properties ($F$) and then systematically explore it to automatically find the best subset ($f \in F$). The chosen list in Equation (3.4) is devised such that it contains a large pool of possibilities. For instance, the position properties are selected such that they assign various 3-D, 2-D, and even 1-D frames to each interest point. Whereas selecting the subset $\{x, y, z\}$ is equivalent with assigning a full 3-D Cartesian frame, selecting the subset $\{z, Za\}$ is equivalent with assigning a 2-D cylindrical frame identical to the one used in Spin Images. However, the choice in our formulation is not limited to these two subsets and one could assign a plethora of alternative frames, such as the cylindrical frames equivalent with subsets $\{x, Xa\}$ or $\{y, Ya\}$. The systematic selection method proposed here chooses the best subset based on the specifics
of the target model shape.

3.8 Descriptor Selection and Evaluation

In general, any solution to the formulated optimization problem should consider finding the \( r, R, \) and \( f \) values simultaneously. While not explored in this work, it is also possible to further extend the optimization problem to incorporate other aspects such as the distance metric for LSD comparison, interest point selection, number of LSDs per point cloud, histogram representation, ambiguity threshold \( \delta_t \), etc. It is very difficult to devise a system to simultaneously solve such a complicated problem. Here, we present a partial solution that tackles one of the most challenging and least explored aspects of this problem, i.e. automatic selection of the best subset \( f \). Whereas previous research has examined automatic selection of other parameters such as \( r \) [83] and \( R \) [64, 65] and interest point placement [26, 64, 108, 127], automatic selection of the \( \chi(\cdot) \) function has been significantly less vigorously investigated [5, 35].

3.8.1 Intuitive Selection of Radii \( r \) and \( R \)

The main simplification in our solution is that we assume that we can find the values of \( R, r, \) and subset \( f \) independent of each other. A major contribution of this work is on exploring various feature selection algorithms and various evaluation measures for selecting subset \( f \) based on the shape of target objects. Radius values \( r \) and \( R \) are selected empirically.

Radius \( R \) is equivalent with the support distance as defined in [55] and \( N_R(p) \) defines the set of neighbour points that contribute to the descriptor of point \( p \). Neighbourhood \( N_R \) is therefore formed only for a small subset of points for which LSDs are generated. PCS
neighbourhood $N_r$ on the other hand is formed for every point in the image. From the perspective of computational efficiency, it is therefore desirable to keep $r$ small.

Our experiments showed that in the absence of scene clutter, larger values of $R$ enhanced point matching performance. With increasing $r$, performance initially improved and then deteriorated, forming a peak at an optimal value. This is explained by the fact that a good value for $r$ is one where the $\vec{k}$ vector associated with each point roughly approximates the surface normal at that point. Since we fix the direction of $\vec{k}$ by the direction of the surface normal (or the viewing angle), it is imperative that the angle between $\vec{k}$ and the surface normal ($\vec{n}$) is never beyond $90^\circ$. Therefore, it is desirable to select $r$ such that the angle $\text{acos}(\vec{k} \cdot \vec{n})$ is small. Very small values of $r$ lead to frames that are susceptible to degradation by noise and large values make $N_r$ prone to include scene and background clutter or be affected by (self-)occlusion. We empirically select the value of $r$ such that the surface normal approximation is good for most of the points, which depends on the point cloud density and on noise levels and shape. Figure 3.3 illustrates the median angle between the $\vec{k}$ vectors and the ground truth surface normals at several points in a LIDAR range image of the Angel model shown in Figure 3.1. Increasing $r$ initially corresponds to a sharp improvement in the error angles followed by a gradual increase, forming an optimal value around $r = 2cm$ in this example. More accurate surface normal estimation and neighbourhood size selection are studied in detail in [83].

Since, in the absence of scene clutter, larger values of $R$ typically lead to better point matching, the upper value of $R$ is mostly bounded by computational considerations, so long as it remains smaller than the dimensions of the model. Therefore, $R$ is set such that neighbourhood formation (i.e. finding $N_R$) and LSD generation could be performed efficiently for several image descriptors. In our experiments, in the presence of scene clutter, we found
that setting $R$ to about $20 - 30\%$ of the average dimension of target objects provided satisfactory results by presenting a balance between computational cost and point matching performance. Average model dimension was estimated by averaging the height, width, and depth values of the bounding boxes of the objects.

### 3.8.2 Automatic Subset Selection

Selecting the best subset $f \subset F$ is challenging and lies at the heart of our work. With 25 properties in $F$, the total number of all possible subsets equals $\sum_{L=1}^{25} \binom{25}{L} \approx 33.5$ million. A large number of these subsets, however, contain redundant information. There are at most three independent properties of each type, i.e. three position, three rotation, and three dispersion properties. For instance, $f = \{x, y, Z_a\}$ is a redundant subset, since the properties included are not independent as $Z_a = \sqrt{x^2 + y^2}$. As such, an LSD computed as a 3-D histogram of these three properties does not carry any more information than a 2-D histogram based on $f = \{x, y\}$. 
After removing such redundant subsets, the total number of all valid subsets decreases by nearly two orders of magnitude, to 443,042. This is still too large a set to search exhaustively in a reasonable time frame.

To address this problem, we use feature selection schemes that explore a portion of this space and aim at finding the best subset through a supervised phase. Formula (3.5) in effect transformed the optimization problem in Formula (3.3) into a feature selection problem. Therefore, we can use any feature selection algorithms for this purpose. In this work, we investigate three different algorithms, namely Genetic Algorithms (GA), Simulated Annealing (SA), and Forward Feature Selection (FFS).

GA models the feature selection problem as an evolutionary survival scheme. Individual properties are likened to single genes and an evolutionary process tries to select the fittest sequence of genes, i.e. the best performing subset. The process is initiated with a pool of randomly chosen subsets as the first generation. At each generation, first, all the subsets in the pool are evaluated and are ranked based on their performance on the training data. The top performing subsets from each generation are then designated as survivors according to a pre-determined survival rate. Next generation subsets are then prepared by random mixing, i.e. mating, between the surviving subsets. Finally, a portion of the subsets produced for the next generation are randomly mutated to counter possible entrapments in local minima. After a series of generations, GA typically converges to a solution corresponding to a high performing subset on the training data.

SA models the problem differently by simulating the process undergone by misplaced atoms in a heated metal during a slow cooling regime. The search starts with a randomly chosen subset and involves a global parameter termed temperature, initially set at a high value. At each step, the process considers taking a random walk, i.e. slightly modifying
the property subset at hand, and probabilistically decides between accepting the random walk or keeping the current subset. Each random walk is evaluated as *downhill* or *uphill* based on whether it improves or degrades the performance on training data. The step size, i.e. a quantified measure of outperforming or underperforming the current subset, is also computed. Downhill walks are always accepted. To escape local minima, uphill walks are also accepted by an *acceptance probability*, computed based on the current temperature and the step size. The probability function is such that accepting larger uphill walks is more likely at higher temperatures. The temperature is reduced gradually through a cooling schedule, thus reducing the chance of accepting uphill walks as the search proceeds. In the limit, as the temperature approaches zero, the algorithm gradually transforms to a greedy search accepting only downhill steps. By setting the search parameters (e.g. the initial temperature and the cooling rate) properly, the algorithm typically converges to an optimum point and finds a high performing subset. Further details regarding GA and SA can be found in [31].

In FFS, we start with examining all the 1-D descriptors and sorting them based on their point matching performance. In the second round, we select the top $k$ from this sorted list, and combine them with each of the 25 properties in $F$ to form a list of 2-D subsets. Thus in every round, a large pool of subsets are tested and sorted based on their performance, and the top $k$ are combined with all the remaining properties to form subsets of a higher dimension. For $k = 1$, this is equivalent with a *greedy search* algorithm. In our experiments however, we use much larger values of $k$ (e.g. $k = 50$), similar to a *beam search*, to avoid the pitfalls of a simple greedy search.
CHAPTER 3. VARIABLE-DIMENSIONAL LOCAL SHAPE DESCRIPTORS

3.8.3 Evaluation and Comparison of LSDs

We use a set of ground truth data for the training phase and use them to evaluate subsets and compare the performance of subset pairs. All the abovementioned feature selection methods require pair-wise comparison between subsets. In GA and FFS, this comparison is used to sort pools of subsets based on their performance on the ground truth data. In SA, pair-wise comparison is used to probabilistically accept or reject a random walk based on the amount of improvement or degradation. Therefore, for SA, a comparison method between subsets is required that not only can determine which subset is superior, but also can quantify "by how much" the superior subset outperforms the inferior.

Formally, if $f_1$ and $f_2$ are two subsets of $F$, we need a subtract operator, $\ominus$, such that $s = f_1 \ominus f_2$ returns a scalar value ($s \in \mathbb{R}$) and $s > 0$ indicates that LSDs constructed based on $f_1$ properties outperform LSDs constructed based on $f_2$ in point matching. We denote $s > 0$ with the $\triangleright$ symbol as $f_1 \triangleright f_2$.

For SA, the magnitude of $f_1 \ominus f_2$ is important, whereas for GA and FFS the sign of the value, i.e. $\text{sgn}(f_1 \ominus f_2)$, is sufficient. Here $\text{sgn}$ denotes the signum function. We also note that since GA and FFS require sorting of a pool of subsets in order to select the top ones at each round, the selected subtract operator should be such that the resulting $\triangleright$ operator is transitive. This is, if $f_1, f_2, f_3 \subset F$, then $(f_1 \triangleright f_2) \& (f_2 \triangleright f_3) \Rightarrow (f_1 \triangleright f_3)$.

For evaluating a subset $f$, we generate the VD-LSD for several model points and their ground truth matches on several training images, showing the model from different viewpoints, and compare them to form a confusion matrix. This matrix is then evaluated to measure the effectiveness of subset $f$ in point matching. Three different measures are used for evaluating a confusion matrix and each have a comparison method associated with it.
These three measures are: Unambiguous Correct Percentage ($UCP$), All Correct Percentage ($ACP$), and Discounted Cumulative Gain ($DCG$).

$UCP$ and $ACP$ are the percentage of correct matches returned, i.e. precision, with and without rejecting ambiguous matches, that is, with $s_t > 0$ and $s_t = 0$ respectively. Since they associate a real number with each subset as a measure of its performance, comparison between subsets is performed by simple subtraction of these values. For instance, if the $UCP_1$ and $UCP_2$ are unambiguous correct percentages of subsets $f_1$ and $f_2$, then $f_1 \ominus f_2 \triangleq UCP_1 - UCP_2$.

The third evaluation measure, the DCG, is an information retrieval measure assigned to each scene LSD. DCG is reported to be a superior measure in some applications [5,62,108]. To compute the DCG of each scene LSD, the corresponding column in the confusion matrix is first sorted in descending order. Ideally, the correct match appears on top of the sorted list by having the largest similarity value in the column, in which case the measure is incremented by the maximum value of 1. But the measure also assigns a discounted increment ($\frac{1}{\log_2(i_x)}$) for correct locations that appear further down the list, where $i_x$ is the rank of the correct match in the sorted list. Finally, the measure is normalized by a maximum possible value such that it lies in the $[0, 1]$ interval. In general, prior to normalization, values higher than one can occur only if multiple ground truth matches exist for the scene LSD. In the special case of evaluating point matches, since there is only one ground truth match for each scene LSD, the normalization factor is 1, and the DCG is easily computed as a single term rather than a summation.

The DCG measure associates an array, whose length is equal to the number of image descriptors, to each subset. Higher values are desired throughout the array, but simple subtraction can no longer be used as a comparison method. To illustrate the difficulty in
CHAPTER 3. VARIABLE-DIMENSIONAL LOCAL SHAPE DESCRIPTORS

Figure 3.4: Normalized Histograms of DCG Values Corresponding to Two Subsets and the Difficulty in Comparing Them.

comparing DCG arrays, Figure 3.4 shows normalized histograms of two randomly generated arrays, representing possible DCG arrays resulting from applying subsets $f_1$ and $f_2$. As the figure illustrates, it is not trivially obvious which DCG array is superior to the other. We apply a nonparametric statistical test to perform this comparison.

When comparing two arrays $DCG_1$ and $DCG_2$, resulting from subsets $f_1$ and $f_2$, the arrays can be interpreted as paired samples from two underlying distributions and we are interested in selecting the distribution that lies further to the right, i.e. closer to 1, in a statistically meaningful way. The underlying distribution cannot be assumed to be normally distributed. In fact, our experiments shows that histograms of DCG arrays typically did not resemble a normal distribution at all. Therefore, we have chosen the Wilcoxon Signed Rank (WSR) test [69, 106] as a comparison method. This nonparametric approach does not assume any normalcy in the distribution of data and is suitable for comparing paired samples. Appendix C provides an introductory explanation of the Wilcoxon Signed Rank test.
Chapter 4

Object Recognition with VD-LSD

In this chapter, we discuss the details of our model-based object recognition system based on the VD-LSD formulation. Implementation details are also presented in this chapter.

4.1 System Design

Several applications could benefit from the enhanced point matching resulting from optimal descriptor selection as described in Chapter 3. These applications include 3-D modeling [118], tracking [120], and model-based object recognition [119]. In this thesis, we investigate the enhancement achieved in object recognition applications through using optimal LSDs. Figure 1.2 in Chapter 1 illustrated the three phase block diagram of a generic detection and localization system in range data. This chapter explains the details of each phase in the model-based object recognition context, as well as the details of our particular implementation.

Figure 4.1 magnifies the point matching block for a model-based object recognition system. As the figure illustrates, optimal LSD selection and LSD generation for the models
is performed offline, while scene LSD generation and matching between model and scene LSDs is performed online. The output of the point matching block is set of candidate point matches between the scene and the models, which are in turn fed to the second phase of object recognition, i.e. hypothesis generation.

In our implementation, the LSD selection block is based on the VD-LSD formulation and is performed as described in Chapter 3. Interest point selection, as explained in Section 3.5, is performed via subsampling of salient regions. Salient regions, as detailed in Section 3.6 are selected via simple thresholding of the $\tilde{e}_3$ values.
4.2 Implementation Details

LSDs were represented as histograms and were normalized so the sum of all the bins in each histogram added to one. This normalization increased robustness with respect to changes of sampling resolution. Both scalar quantized and vector quantized representations were tested. Hashing schemes for efficient storage and LSD matching were also explored earlier [118] and, as expected, were found to be as effective as scalar quantized representations in matching quality. Coarse binning was used to accommodate high-dimensional scalar quantized histograms.

For vector quantized histograms, several cluster centres first need to be identified such that they captured the underlying distribution in the $L$-D space of properties. For each object, an $L$-D space was populated with local properties of points in salient regions of the model and the k-means algorithm [71] was used for cluster finding. Centroids of the found clusters were used as the embedding for the vector quantized representation. During online histogram formation, each $L$-D entry, resulting from the properties of a neighbour point, contributes to increment the bin associated with its nearest cluster centre. The k-d tree structure was used for this nearest neighbour search. Since $L < 9$, k-d trees provide very efficient nearest neighbour finding. To avoid the cluster finding process for each subset during the training phase, only scalar quantized histograms were used during the training phase. Histogram Intersection [117] was used as a robust distance metric for LSD matching.

The RANSAC algorithm [32] was used for hypothesis generation, the second block in Figure 1.2. Rigidity constraints (Section 2.3) were used to contain the combinatorial cost of RANSAC by discarding group matches that were obviously incorrect. For efficient computation, only length ratios were imposed and no constraint was imposed on the angles.
As explained in Section 2.3, a simple count of the number of aligned points was used for the verification step inside each RANSAC iteration. A small distance threshold ($\xi$) was used for distinguishing between aligned and non-aligned points. The count indicates the area of model, relative to its total surface area, that is aligned with the scene point cloud.

ICP was used for pose refinement, the final phase of localization. Voxel ICP [3] was used as an efficient ICP implementation and also for efficient verification inside each RANSAC iteration. The same voxelized space was also used for efficient neighbourhood formation. In our experiments, this provided a more efficient implementation for neighbourhood formation than other methods such as utilizing k-d tree structures or other nearest neighbour search algorithms.

Our descriptors, as formulated in Chapter 3, assume that both the models and the image data are represented as 3-D point clouds in our setup. This provides an advantage over methods that operate on surface meshes since most range sensing devices provide range images or point clouds as their primary output and require further processing to convert range images to other formats such as a surface mesh. It is also beneficial to represent models and images in the same format to simplify the LSD computation phase as the same computation could be performed on both the models and the images. Furthermore, the algorithm can be directly applied to pair-wise range image registration, for 3-D modeling or tracking applications, without any changes.

For descriptor matching we used a slightly modified version of the Nene-Nayar search since the algorithm is more efficient than k-d tree structures for neighbour finding in dimensionalities higher than 10 [85]. The algorithm performs an exhaustive search for the nearest neighbour but achieves computational efficiency by limiting the search within a small region around the query point. In the original algorithm, the region around each point is in
the shape of a hyper-cube, but to conform with the properties of the Euclidean distance, only points within a hyper-sphere inscribed within the hyper-cube are considered valid and the rest are discarded. We have preferred Histogram Intersection [9] for LSD matching since it provides more robust matching than applying the Euclidean distance. Histogram Intersection and Euclidean distance are equivalent with the $L_1$ and $L_2$ norms respectively. Consequently, we altered the original Nene-Nayar algorithm by accepting all the points within the hyper-cube region. Unambiguous model-scene LSD matches were detected and established by comparing the nearest and the second closest matches as described in Section 3.2.
Chapter 5

Experiments

We present our experimental results in this chapter. Observations are made on the automatically selected subsets and the effect of object geometry and sensor noise level are discussed. Recognition rates and precision plots are used to show the superiority of optimal VD-LSD over Spin Images. Timing results, specifying a breakdown of processing times at each computing block, are also included.

5.1 Experimental Setup

Object recognition experiments were performed with various models and with both LIDAR and dense stereo images. The 3-D models included the four models from the University of Western Australia (UWA) [78], the RADARSAT satellite model from MDA Space Missions [53, 104], and five new models (Queen’s models) that we generated by merging several range images acquired with a Konica-Minolta VIVID 3-D scanner [82] (Figure 5.1). Scene data included LIDAR images of the UWA set and both LIDAR and dense stereo images of Queen’s models and RADARSAT, 686 images in total. By comparison, experimental
results on 100 images are reported for Spin Images [55]. Reported results for the Tensor-Based Correspondence method [78] include experiments with 50 LIDAR range images and 560 simulated range images.

Values of $R = 10\ cm$ and $r = 1\ cm$ were selected for the Queen’s models based on experiments on the training images. A support angle value of $A_s = 90^\circ$ was used to counter the self-occlusion effect when generating model LSDs [55].

We limited the total number of subsets that were explicitly explored by feature selection methods to less than 4,000, roughly equivalent to 1% of the search space. The parameters of the feature selection algorithms, such as the initial and freezing temperatures for SA, were set empirically. We noticed that the outcome was typically not too sensitive to these values, so long as they remained within a reasonable range. The computational cost of training with the $DCG$ measure was higher than with the other two measures, as it required computing entire confusion matrices (and sorting their columns) as opposed to finding only the top or the top two matches when using the $ACP$ and $UCP$ measures. In our implementation, training took a few (e.g. 4-6) hours per model on a 2.8 GHz single processor PC.

LSDs were distributed roughly uniformly on salient areas of models and images. Salient areas were selected by filtering $\bar{e}_3$ values as described in Section 3.5. Subsampling rates were selected such that the typical numbers of LSDs per model and per image were about...
2,000 and 1,000 respectively. Subsampling of image salient areas was performed randomly. For model clouds, the subsampled points were selected such that the distance between each pair of selected points (in Cartesian space) was not less than $\xi$. The distance threshold $\xi$ was selected as a fraction of the model size, e.g. $1 - 2\%$ of the average dimensions. This ensured that unnecessary computation in matching a scene LSD to two nearly identical model LSDs was avoided. The average dimension, as mentioned in Section 3.8.1, was estimated by averaging the height, width, and depth values. The same distance threshold $\xi$ was also used in establishing true correspondences and forming ground truth confusion matrices required for the training phase.

Ground truth alignments were confirmed visually both for the training images and the test images. Test image ground truth alignments were required for the purpose of evaluating estimated poses. Since the ground truth alignments could not be 100\% accurate, a small deviance (e.g. $< 3$ cm and $< 5^{\circ}$) between the recovered and the ground truth alignments was tolerated during evaluation \footnote{Our entire data set containing our five new models, LIDAR, dense stereo, and Flash LIDAR scenes, and ground truth alignments is available online through the Queen’s University Robotics and Computer Vision Laboratory (RCVLab) website.}.

## 5.2 Avoiding Overfitting

Training algorithms are sometimes susceptible to overfitting and rely on the proper selection of training data. Simply put, overfitting could occur when training data is not sufficient with respect to the size of the parameter space. In such cases, a false solution might fit the idiosyncrasies in the training data very well without correctly representing the underlying structure of the data. In 2-D, fitting a high degree polynomial to a set of points that roughly form a line is an example of overfitting. In our algorithm, overfitting might occur
if too few training images are used or if sparse subsampling results in too few LSDs placed on each training image. We experimentally determined the appropriate numbers to avoid overfitting.

The number of training images were varied from 1 to 10 and the selected subsets were observed. Table 5.1 shows the selected subsets for Angel and Big Bird models using the FFS method with ACP measure. Multiple LSDs are reported when more than one subset (e.g. 2) shared the best results on the training data. As the table illustrates, if the number of training images is large enough (e.g. > 6), adding more training images does not change the selected subset. This could perhaps be explained by the fact that six images taken from different viewpoints could nearly cover the entire area of a mostly convex object and could provide a good overall view of the object.

Similar experiments were performed to measure the sensitivity of the training methods
to the selection of training images and to the number and placement of LSDs on each training image. Too few training images (e.g. < 6) or too few LSDs per image (e.g. < 100) resulted in some sensitivity to the choice of the training images or the location of LSDs placed on each image. With enough training images and enough LSDs per image, these sensitivities subsided to virtual non-existence. Based on these results, in all the subsequent experiments, nine training images were used for each model and ∼ 200 LSDs were placed on each of the training images. Training image LSDs were placed both on the visible areas of the model, and on scene clutter so that the feature selection process favoured discriminating subsets that could perform well in the presence of clutter. With nine training images, the selected subset for each model did not change with the choice of training images or the placement of LSDs on the training images except in one case: for Big Bird, the selected subset changed from \{z, D, \psi\} to \{z, D, \cos(\psi)\} with a particular placement of LSDs on a particular selection of training images. This is a minor difference and had little effect on the overall point matching outcome.

5.3 Queen’s Models, LIDAR Range Images

Comprehensive LSD selection was performed on the five Queen’s models and LIDAR training images, using all three methods and all three measures. Using the ACP and UCP measures, as expected, the optimal subsets found for any given model by the three different methods were identical, confirming the effectiveness of all three methods (i.e. FFS, SA, and GA). DCG outcomes were less consistent and in general underperformed those of UCP and ACP. For brevity, we report only test results with subsets chosen by FFS – ACP, as follows for the five models: Angel \{z, Z_a, \cos(\psi)\}, Big Bird \{z, D, \psi\}, Gnome \{z, Z_a, \bar{e}_3\}, Kid \{Z_a, D, \psi, e_3\}, and Zoe \{z, D, \bar{e}_3\}. 
The average $ACP$ rates (over all five models, recorded on the training data) for the selected subsets was 10.53%. The average $ACP$ rates on the same training data for Spin Images [55] (i.e. $\{z, Za\}$) and Pairwise Geometric Histograms [7] (i.e. $\{z, \psi\}$) were 8.3% and 3.68% respectively. Among 1-D descriptors, those based on $\{z\}$ lead to the highest $ACP$ rates (average 2.76%). Other properties that typically ranked among the best 1-D properties include $Z_a$, $D$, $e_3$, $\bar{e}_3$, $\cos(\psi)$, $\psi$, and to a lesser extent $X_a$ and $Y_a$. Among 2-D descriptors, those based on $\{z, D\}$ resulted in the highest $ACP$ levels for all five models, averaging 8.62%. That is, based on the performance on the training data, Spin Images or Pairwise Geometric Histograms did not provide the best point matching results even among 2-D descriptors.

We first tested the tuned VD-LSDs, i.e. those with the chosen subsets, for pose determination of Queen’s models in isolated images with minimal scene clutter. For this test, four images of each object from different views were acquired and our algorithm was able to find the correct alignment between these images and the corresponding models 100% of the time with only 15 RANSAC iterations, indicating very high precision levels.

We then tested detection and localization in cluttered scenes and in the presence of partial occlusion. We acquired 39 images of each object in highly cluttered scenes, where the clutter included other objects and background. Large sections of objects were occluded in several of these scenes. Figure 5.2(a) illustrates a sample of such an image. Figure 5.2(b) shows the same range image in which the models as found by our algorithm are highlighted. Tables 5.2 and 5.3 show recognition rates, i.e. the percentage of images in which each model was correctly detected and localized, using tuned VD-LSD with scalar quantization ($SQ$) and vector quantization ($VQ$), and Spin Images, with 100 and 500 RANSAC iterations. To take into account the random selection of group matches in RANSAC, each test was run...
multiple times for all the experiments reported throughout this thesis and median rates are reported. No significant difference was observed in the detection rate among various runs of RANSAC.

Ambiguous match rejection, i.e. \( s_t > 0 \), had a significant effect on improving recognition rates. Since the same technique could be applied to enhance the point matching results for Spin Images, Spin Images results with \( s_t > 0 \) are also reported. Original (orig.) and improved (impr.) rows in the table correspond to Spin Images with \( s_t = 0 \) and \( s_t > 0 \) respectively. We used our implementation of Spin Images that performed in accordance with results reported in [55, 79]. No PCA-compression was utilized for Spin Images so the algorithm operated at its highest performance level. To provide a fair comparison, 3-point RANSAC (see Appendix A) was used for both VD-LSD and Spin Images in all the experiments reported here. This is in contrast with, and a fairer comparison than, results reported in [78] where 1-point RANSAC is used for Tensor matching for comparison against Spin Images with 3-point RANSAC.

The results show a clear lead over Spin Image recognition rates in nearly all cases. The only exception was localizing the Kid model in which Spin Images outperformed vector quantized (although not scalar quantized) VD-LSD. Our average recognition rates with 100 RANSAC iterations using scalar quantization are about 35 percentage points better than that of improved Spin Images with the same number of iterations and nearly 15 percentage points better than that of Spin Images with 500 RANSAC iterations. Vector quantized VD-LSD with 100 RANSAC iterations also performed over 16 percentage points better than Spin Images with the same number of iterations.

Recognition rates reported in Tables 5.2 and 5.3 reflect experimental results where the
(a) LIDAR Range Image  
(b) Objects Recognized

Figure 5.2: (a) Sample LIDAR Range Image that Contains Angel, Big Bird, Kid, and Zoe (b) the Models Highlighted as Found by Our Algorithm.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Angel</th>
<th>Big Bird</th>
<th>Gnome</th>
<th>Kid</th>
<th>Zoe</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD-LSD (SQ)</td>
<td>76.9</td>
<td>87.2</td>
<td>56.4</td>
<td>64.1</td>
<td>48.7</td>
<td><strong>66.7</strong></td>
</tr>
<tr>
<td>VD-LSD (VQ)</td>
<td>41.0</td>
<td>84.6</td>
<td>43.6</td>
<td>30.8</td>
<td>38.5</td>
<td><strong>47.7</strong></td>
</tr>
<tr>
<td>Spin Image (impr.)</td>
<td>17.9</td>
<td>38.5</td>
<td>28.2</td>
<td>43.6</td>
<td>28.2</td>
<td><strong>31.3</strong></td>
</tr>
<tr>
<td>Spin Image (orig.)</td>
<td>2.6</td>
<td>17.9</td>
<td>10.3</td>
<td>15.4</td>
<td>2.6</td>
<td><strong>9.7</strong></td>
</tr>
</tbody>
</table>

Table 5.2: Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 100 RANSAC Iterations (with Nene-Nayar Search).

Nene-Nayar nearest neighbour search [85] was used for computational efficiency in matching scene LSDs to their model LSD match. The search parameter $\varepsilon$, defining the hyper-cube size, was set based on experiments with the training images. Finding a good value for $\varepsilon$ requires delicate tuning to balance computational efficiency against nearest neighbour search performance. Since the overall point matching performance is affected with bad choices of $\varepsilon$, it might be beneficial to also compare the recognition rates when this algorithm is not utilized. Tables 5.4 and 5.5 presents the recognition rates without the Nene-Nayar search, again with 100 and 500 RANSAC iterations respectively. These results echo those of the previous tables and confirm the significant lead of tuned VD-LSD over Spin Images.
CHAPTER 5. EXPERIMENTS

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Angel</th>
<th>Big Bird</th>
<th>Gnome</th>
<th>Kid</th>
<th>Zoe</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD-LSD (SQ)</td>
<td>87.2</td>
<td>97.4</td>
<td>74.4</td>
<td>79.5</td>
<td>61.5</td>
<td>80.0</td>
</tr>
<tr>
<td>VD-LSD (VQ)</td>
<td>53.8</td>
<td>92.3</td>
<td>59.0</td>
<td>51.3</td>
<td>66.7</td>
<td>64.6</td>
</tr>
<tr>
<td>Spin Image (impr.)</td>
<td>43.6</td>
<td>59.0</td>
<td>46.2</td>
<td>66.7</td>
<td>46.2</td>
<td>52.3</td>
</tr>
<tr>
<td>Spin Image (orig.)</td>
<td>23.1</td>
<td>64.1</td>
<td>25.6</td>
<td>43.6</td>
<td>23.1</td>
<td>35.9</td>
</tr>
</tbody>
</table>

Table 5.3: Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 500 RANSAC Iterations (with Nene-Nayar Search).

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Angel</th>
<th>Big Bird</th>
<th>Gnome</th>
<th>Kid</th>
<th>Zoe</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD-LSD (SQ)</td>
<td>79.5</td>
<td>94.9</td>
<td>56.4</td>
<td>61.5</td>
<td>48.7</td>
<td>83.8</td>
</tr>
<tr>
<td>VD-LSD (VQ)</td>
<td>35.9</td>
<td>92.3</td>
<td>51.3</td>
<td>38.5</td>
<td>43.6</td>
<td>67.7</td>
</tr>
<tr>
<td>Spin Image (impr.)</td>
<td>33.3</td>
<td>64.1</td>
<td>25.6</td>
<td>38.5</td>
<td>20.5</td>
<td>53.8</td>
</tr>
<tr>
<td>Spin Image (orig.)</td>
<td>0.0</td>
<td>17.9</td>
<td>10.3</td>
<td>17.9</td>
<td>10.3</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Table 5.4: Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 100 RANSAC Iterations (without Nene-Nayar Search).

For assigning the L-D contribution of each neighbour point to vector quantized histograms, approximate k-d trees [12, 41] were found to be virtually as effective as, and in some cases even more effective than, exact k-d trees, and were therefore preferred due to their computational advantage. The number of cluster centres was set to 256 based on experiments on the training data with various numbers ranging from 16 to 512. Table 5.6 shows median recognition rates without Nene-Nayar search for the Gnome model after 500 RANSAC iterations on the test images, with various number of cluster centres and using

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Angel</th>
<th>Big Bird</th>
<th>Gnome</th>
<th>Kid</th>
<th>Zoe</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD-LSD (SQ)</td>
<td>89.7</td>
<td>100</td>
<td>70.5</td>
<td>84.6</td>
<td>71.8</td>
<td>83.8</td>
</tr>
<tr>
<td>VD-LSD (VQ)</td>
<td>56.4</td>
<td>97.4</td>
<td>69.2</td>
<td>51.3</td>
<td>64.1</td>
<td>67.7</td>
</tr>
<tr>
<td>Spin Image (impr.)</td>
<td>53.8</td>
<td>84.6</td>
<td>38.5</td>
<td>51.3</td>
<td>41.0</td>
<td>53.8</td>
</tr>
<tr>
<td>Spin Image (orig.)</td>
<td>15.4</td>
<td>64.1</td>
<td>25.6</td>
<td>43.6</td>
<td>28.2</td>
<td>35.4</td>
</tr>
</tbody>
</table>

Table 5.5: Recognition Rates (Percentage) in LIDAR Cluttered Scenes with 500 RANSAC Iterations (without Nene-Nayar Search).
Table 5.6: Gnome Model Recognition Rates (Percentage) without Nene-Nayar Search in LIDAR Cluttered Scenes with 500 RANSAC Iterations, with Various Numbers of Cluster Centres for Vector Quantization, and Exact and Approximate k-d tree Bin Assignment.

<table>
<thead>
<tr>
<th># Cluster Centres</th>
<th>Exact</th>
<th>Approximate</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>5.1</td>
<td>2.6</td>
</tr>
<tr>
<td>32</td>
<td>23.1</td>
<td>20.5</td>
</tr>
<tr>
<td>64</td>
<td>20.5</td>
<td>20.5</td>
</tr>
<tr>
<td>128</td>
<td>61.5</td>
<td>61.5</td>
</tr>
<tr>
<td>256</td>
<td>66.7</td>
<td>69.2</td>
</tr>
<tr>
<td>512</td>
<td>64.1</td>
<td>61.5</td>
</tr>
</tbody>
</table>

Both exact and approximate k-d trees. Choosing 256 cluster centres enabled a fair comparison against Spin Images in terms of descriptor size. We used $16 \times 16$ Spin Images so that the total length of Spin Image and vector quantized VD-LSD arrays were identical. $16 \times 16$ Spin Images were found to slightly outperform the $15 \times 15$ histograms used in the original Spin Image publication.

Table 5.7 presents the average processing times for recognizing Big Bird in the test image using vector quantized VD-LSD with 256 cluster centres with 500 RANSAC iterations. The timing experiments were performed on two computers, one with an Intel Pentium 4 2.8GHz processor and another with an Intel Core 2 Quad Q6600 CPU at 2.4GHz. The table presents run times on both machines. Note that our implementation does not explicitly take advantage of the parallelism capabilities of multi-core CPUs. The table also presents a break down of the processing time into various stages of the algorithm, including voxelization, PCS formation, scene LSD generation, comparison with model LSDs, RANSAC, and ICP. The model (Big Bird) consisted of 30,183 points and 1,959 descriptors were generated for it offline. The range images contained 34,471 points on average and 965 descriptors were generated per image on average. Computing the PCS frames of all the points is the major extra online computational step of VD-LSD over Spin Images which took less than 400ms on average on the Q6600 processor. By contrast, Spin Images require surface normal (and tangent plane) estimation for all interest points, which took 35ms per image on
average. The table also illustrates that RANSAC is a major computational block of the entire recognition algorithm and any efforts to reduce the required number of iterations, by way of enhancing precision levels, could also potentially improve computational efficiency.

We also note that several of the properties selected for Queen’s models’ VD-LSDs could be computed without forming the PCS frames if the surface normals were available at each point, for example through the sensing device. As a result, detection and localization of Big Bird for instance, would not require PCS formation for all the points since properties \( z, D, \) and \( \psi \) only require the surface normal direction at each point.

Higher than Spin Image recognition rates reported in Tables 5.2 through 5.5 are the result of higher precision levels in point matching. To further illustrate this case, we present sample figures that explicitly compare the obtained precision levels. Figure 5.3 illustrates precision levels for detecting the Angel, Big Bird, and Gnome models in three sample images using scalar quantized and vector quantized VD-LSD and Spin Images, plotted against \( s_t \), the threshold for rejecting ambiguous matches. As \( s_t \) rises, precision levels for both Spin Images and VD-LSD improve. Since RANSAC requires 3 point matches to generate a transformation matrix, increasing \( s_t \) beyond a critical value, indicated with vertical
lines in the plots, saturates the improvement and causes failure. As observed in the sample plots, Spin Images reached this saturation level at lower precision levels than VD-LSD. $s_t \approx 0.06 - 0.08$ typically provides good results for VD-LSD without risking saturation. To avoid the risk of saturation, $s_t$ was selected adaptively (with negligible computational cost) in all the experiments. That is, instead of setting a hard threshold, the value was computed online such that the top $N$ (e.g. $N=25$ to $50$) unambiguous matches were returned in each case.

## 5.4 Queen’s Models, Stereo Range Images

In the next experiment, we performed object recognition in dense stereo data. While model-based object recognition in stereovision data has been researched for many years [115,124], to the best of our knowledge, this is the first time high recognition rates (> 70%) are reported in real low-quality stereo range images of complex free-form objects.

In order to enhance the coverage of dense stereo point clouds, we projected random textures onto our scenes while collecting stereo images. The Bumblebee camera by Point Grey Research, along with its stereo processing software library, was used for range image acquisition. Depth filtering discarded a large portion of the background points so clutter
was mostly due to the presence of other objects in the scene. Stereo images contained much greater noise levels than LIDAR data. Higher accuracy stereo images could be generated, at the expense of longer image capture times, by projecting multiple different random patterns on the scenes and merging the results via a median filter to increase coverage and enhance accuracy. For brevity, we only report results without the use of such an averaging technique in this paper.

It is reasonable to assume that the optimal subsets not only depend on model shapes, but also on the data acquisition device as the noise characteristics in LIDAR and stereo data vary vastly. Therefore, new training with stereo range images was required. *Batch training* was utilized to select the best subset for recognizing the ensemble of our five models in stereo data.

In some cases, instead of optimizing VD-LSDs to the specifics of each individual model, it might be desirable to perform training for groups of models. Such batch training could bring two advantages: first, in that training time becomes less dependent of the number of models in the database. Second, in that it avoids a potential computational bottleneck that could arise in online LSD generation if the number of unique VD-LSDs selected for models in the database were too large. The negative aspect of batch training is that, as the number of models in the database grows, the selected LSD might become too generic and sub-optimal for detecting individual models. This is a trade-off that needs to be addressed at the design level based on the requirements of the object recognition system at hand.

For instance, in space applications, such as satellite tracking or detecting various parts of a spacecraft, the number of models is small and selecting the optimal VD-LSD for each model is highly beneficial in order to obtain high recognition rates. On the other hand, in other applications where the number of models in the database is very large, selecting the
best LSD for the entire set of models is preferable. Is it also possible to group objects into shape categories and perform batch training for each category. An appealing aspect of our formulation is that it allows for both individual training or batch training, depending on the requirements of the system. Batch training is easily accomplished by combining training images of all the models in the training data and generating ground truth confusion matrices accordingly.

To illustrate batch training, in our experiments with dense stereo images, we aimed at selecting one VD-LSD subset for the entire group of Queen’s models. Batch training selected two subsets, with identical performance on the training data, for the group of our five models. These two subsets were \( \{z, D\} \) and \( \{z, Z_a\} \). The second subset is identical to the two properties used in constructing Spin Images. Tables 5.8 and 5.9 report the recognition rates with the \( \{z, Z_a\} \) subset and with use of Nene-Nayar search. The number of test images are 69 for Gnome, 68 for Big Bird and Kid, 66 for Angel, and 65 for Zoe. Spin Image results are not reported since the selected subset is identical to that of Spin Images. However, we note that, similar to the LIDAR experiments, the use of \( s_t > 0 \) improved the recognition rate over the original Spin Image method. A higher number of RANSAC iterations were required compared to the LIDAR experiments to compensate for the higher noise levels in stereo data. Figure 5.4(a) illustrates a sample stereo image in which models Angel, Gnome, and Zoe are present. Figure 5.4(b) illustrates the same range image in which the models as found by our algorithm are highlighted.

Apart from the fact that batch training favours more generic descriptors, higher noise levels in stereo data could also partially account for the automatic selection of lower dimensional and more generic descriptors in the stereo data. As a result of higher noise levels, surface normal estimation is not as accurate in stereo data as it is in LIDAR data. Therefore,
whereas subsets selected for recognition in LIDAR data took advantage of surface normal directions (e.g. by utilizing property $\psi$), automatic feature selection reverted to more basic properties that could be computed more robustly in stereo data. This resulted in lower precision levels in the point matching results, which in turn, had to be countered by a higher number of RANSAC iterations.

### 5.5 UWA Models

The UWA models were processed similarly, using the FFS method and $ACP$ measure for training. Since the model and scene data was available in surface mesh format, we first sampled the data into point clouds, averaging $\sim 12,500$ 3-D points per image. Values of $R = 7$ cm and $r = 1$ cm were selected based on experiments on the training images. Selected subsets were $\{z, Z_a, \psi, \bar{e}_3\}$, $\{Z_a, \bar{e}_3\}$, $\{z, D\}$, and $\{z, D\}$ for Chef, Chicken, Parasaurolophus, and T-Rex respectively.

Figure 5.5 illustrates a sample scene image containing all four models and the detected and localized models as recognized by VD-LSD. As the figure illustrates, model Chicken, partially present on the left side of the range image, was not detected in this image due to being highly occluded.
CHAPTER 5. EXPERIMENTS

Figure 5.4: (a) Sample Stereo Range Image that Contains Angle, Gnome, and Zoe (b) the Models Highlighted as Found by Our Algorithm.

Results were similar to those of previous models, repeating the advantage over Spin Images. The selected subsets for T-Rex and Parasaurolophus (i.e. \( \{z, D\} \)) is similar to that of Spin Images. As a result, VD-LSD recognition rates only marginally outperform Spin Images for those two models. For Chef and Chicken on the other hand, the recognition rates for vector-quantized VD-LSD were nearly 17 percentage points better than that of Spin Images with the same number of RANSAC iterations.

5.6 RADARSAT

We performed object recognition experiments on both LIDAR and stereo range images of a one-fifth scale mock-up model of the RADARSAT satellite. Stereo range images
were acquired as described earlier. LIDAR images were acquired with an Optech ILRIS 3-D LIDAR sensor that was mounted on a manipulator robot. The satellite mock-up was also mounted on another manipulator robot. As the two robotic arms were commanded at different configurations, the sensors could capture images of the satellite from various distances and viewpoints. Figure 5.6 illustrates the schematic diagram of this experimental setup at MDA Space Missions.

For evaluation, the robotic arms were moved to simulate the motion of a spacecraft approaching the satellite. As in the case of a real spacecraft approaching a satellite, movement of the arms was slow relative to the data acquisition rate so the motion skew was negligible. Since the configuration of both manipulators was known during the scanning period, a ground truth rigid transformation existed for each scene. A CAD model of the RADARSAT satellite is shown in Figure 5.1.

Here we report the recognition rates for each frame using tuned VD-LSDs. But we note
that in reality, the VD-LSDs are used only for initial pose acquisition and a (faster) pose tracking technique, the Bounded Hough Transform [104] is used for inter-frame tracking. VD-LSD pose acquisition needs only to be invoked on the initial frame, and potentially on a limited number of subsequent frames only if the tracking is lost. Since both stereo and LIDAR range images of the satellite were denser than needed, a 50\% downsampling was used to reduce the processing times. After this downsampling, the average number of points per image was \(\sim 22,000\) and \(\sim 25,500\) for LIDAR and stereo images respectively. With \(r = 5cm\) and \(R = 30cm\), the selected subsets for the RADARSAT based on LIDAR and stereo training images are \(\{X_a, D, \bar{e}_1, \bar{e}_2, |x|, \bar{\theta}\}\) and \(\{X_a, D, e_1, e_2, e_3, |x|, \cos(\phi), \bar{\theta}\}\) respectively.

Figures 5.7 and 5.8 illustrate sample LIDAR and stereo images along with the estimated pose of the satellite in each image. The satellite model is symmetric since its front and back views are identical except for a small grapple fixture located on the front side. The grapple fixture, shown in Figure 5.9, is not visible in several of our test images. Furthermore, its small size relative to the size of the model, \(\sim 13cm\) in diameter compared to \(\sim 210cm\) span of the satellite, means that in many range images, particularly in stereo images, the range
of noise is comparable to the size of the grapple fixture. For instance, in the range image shown in Figure 5.8(a), the grapple fixture is not recognizable even though the image is taken from a front view.

As a result, based on the range data alone, mirror alignments, which align the back side of satellite with the front of the images, are indistinguishable from correct alignments. This ambiguity could be resolved by other means, such as the use of intensity images. Table 5.10 presents the recognition rates for the satellite in 101 LIDAR range images, where mirror alignments are accepted as correct and Nene-Nayar search is not utilized for LSD matching.
CHAPTER 5. EXPERIMENTS

Figure 5.9: RADARSAT Grapple Fixture.

<table>
<thead>
<tr>
<th>RANSAC</th>
<th>Descriptor</th>
<th>Recog. Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Iterations</td>
<td>VD-LSD (SQ)</td>
<td>74.3</td>
</tr>
<tr>
<td></td>
<td>VD-LSD (VQ)</td>
<td>67.3</td>
</tr>
<tr>
<td></td>
<td>Spin Image (impr.)</td>
<td>40.6</td>
</tr>
<tr>
<td>500 Iterations</td>
<td>VD-LSD (SQ)</td>
<td>84.2</td>
</tr>
<tr>
<td></td>
<td>VD-LSD (VQ)</td>
<td>84.2</td>
</tr>
<tr>
<td></td>
<td>Spin Image (impr.)</td>
<td>72.3</td>
</tr>
</tbody>
</table>

Table 5.10: RADARSAT Recognition Rates (Percentage) in LIDAR Scenes with 500 and 100 RANSAC Iterations (Without Nene-Nayar Search).

Improved Spin Images (i.e. with $s_t > 0$) recognition rates are also reported for comparison. Using the Nene-Nayar search, vector quantized VD-LSDs outperformed Spin Images by 66.3% versus 35.6% recognition rates with 100 RANSAC iterations and by 83.2% versus 70.3% with 500 RANSAC iterations. In stereo test images, 100% recognition was obtained when scalar quantized VD-LSD were used. Vector quantized VD-LSD provided 66.7% recognition rates, 16.7 percentage point higher than 50.0% provided by Spin Images. Our stereo images of the satellite did not include any difficult side views and, as such, posed easier pose acquisition problems than those of the LIDAR images. This is reflected in the higher recognition rates obtained on the stereo images. To summarize, the results from the pose acquisition experiments with the RADARSAT model repeat the advantage of VD-LSD over Spin Images observed in previous experiments.

We observe that the subsets selected for the RADARSAT model with LIDAR and stereo training data are very similar to each other but differ from those selected for Queen’s and UWA models in two aspects: first, in that the dimensionality of the RADARSAT subsets are
larger (e.g. 6-D with LIDAR training images); and second, in the choice of properties that are selected. This can be explained by the fact that the shape of the satellite model differs greatly from that of all the other models in our experiments. For instance, the satellite model contains large planar areas on the solar panels, the base, and the main body, and also contains an abundance of rectangular edges. By contrast, the other models are free-form and contain no rectangular edges and only small planar or nearly planar areas.

It could be argued that since planar areas are less distinctive and unique, feature selection had to rely on more properties to distinguish between different neighbourhoods. Presence of large planar areas could also explain why the feature selection routine automatically opted for a different set of properties for the RADARSAT than those selected for the other models. For instance, the $z$ property, equivalent with the distance from the tangent plane, is used in the selected subsets of four out of the five Queen’s models and three out of the four UWA models but it is not used in the subsets chosen for the satellite model. This is because in large planar areas, the distance from the tangent plane is constant (and equal to zero) across descriptor neighbourhoods ($N_R$). By contrast, properties such as $e_1$ and $e_2$ (or $\overline{e}_1$ and $\overline{e}_2$) are effective in distinguishing different areas of planar rectangular shapes depending on the distance to the horizontal and vertical edges. This is illustrated in Figure 5.10, where two points and their respective $N_r$ neighbourhoods on the left solar panel of the satellite are highlighted. Axes $\vec{i}$ and $\vec{j}$ of each neighbourhood are also illustrated such that the length of each axis is proportional to the magnitude of its corresponding eigenvalue. In both cases, the $\vec{k}$ axis is perpendicular to the solar panel, correctly approximating the surface normal, and its length is zero since there is no dispersion along the surface normal in a completely planar neighbourhood. To better visualize the example, the model shown in the figure is downsampled by a factor of four and the PCS radius is larger than the one
used in the reported experiments.

The distance from the tangent plane (and the direction of the surface normal) are identical for both points. As such, the $z$, $\psi$, and $\cos(\psi)$ properties for the two points would be identical as well. However, the ratio of the lengths of the major and semi-major axes are not identical for the two points. For the selected point to the right, the length of the $\vec{i}$ axis is considerably longer than that of the $\vec{j}$ axis and the $e_1$ and $e_2$ properties are 0.78 and 0.22 respectively. For the other selected point, closer to the left corner of the solar panel, the length ratio is not as high and the $e_1$ and $e_2$ properties are 0.61 and 0.39 respectively. This illustrates that the two selected points could be distinguished based on their $e_1$ and $e_2$ properties, even though their $z$ and $\psi$ properties are identical. We emphasize that such analysis is purely for intuitive explanation of the selected properties and feature selection is performed entirely automatically in all cases.
Chapter 6

Conclusions and Future Work

In this work, we expressed the selection of good Local Shape Descriptors for point matching across 3-D surfaces as an optimization problem. Variable-Dimensional Local Shape Descriptors were then used to build a generative platform, upon which the optimization problem could be mapped into a solvable feature selection problem. On the theoretical level, our descriptors are a generalization that subsumes a large class of point matching algorithms. On the practical side, as confirmed experimentally, higher precision levels provided by descriptors tuned to the geometry of target objects lead to higher recognition rates than generic descriptors such as Spin Images with similar processing times.

Optimizing descriptors to the geometry of models has significant implications in designing any model-based object recognition system. This is because all model-based methods assume that the model set is known in advance and as a result could benefit from pre-processing models offline in order to enhance the online performance. Batch training was shown to facilitate descriptor selection and could be used even in cases where the number of models is large. Applications in pose acquisition, tracking, and 3-D modeling also benefit from optimized descriptors since they too require efficient alignment of range data.
In particular, extensive experimental results in satellite pose acquisition were presented in this thesis. As the algorithm is easily parallelizable, a hardware implementation is being developed aimed at multiple frames per seconds processing time in the particular case of satellite pose acquisition.

Our formulation also allows for selecting descriptors based on the range sensing device. Since range image characteristics, such as noise distribution and coverage, are sensor dependent, invariance and descriptiveness of properties depend not only on the geometry of the models, but also on the sensing device. As a result, it is natural that descriptors that are optimal for one sensor (e.g. LIDAR) might not be optimal for another (e.g. stereovision). The best properties for each sensor or sensor class could be determined by using a large number of images of varied scenes for training. This would be particularly beneficial for pair-wise image registration for 3-D modeling. In such applications, training to the geometry of the model is not possible since the model does not exist yet. Therefore, optimal descriptor selection for modeling applications could be based on tuning to the range sensing device.

We observe that, from a list of 25 properties, the chosen subsets for all the models except the satellite are all based on seven properties: \( z \), \( \psi \), \( \cos(\psi) \), \( e_3 \), \( \bar{e}_3 \), \( Z_a \), and \( D \). We speculate that this re-occurrence, at least partially, is due to an underlying high-level similarity between the models. The five Queen’s models for instance, were all lawn ornaments that shared (or lacked) some common elements, such as lack of rectangular edges or large planar areas. This leaves room for investigating the possibility of grouping objects and finding optimal subsets for sets of objects, based on high-level shape characteristics, rather than processing individual models. We aim at investigating a grouping scheme that could ease the computational burden of the training phase by placing objects into categories based on
a fast pre-processing step. Batch training could then be performed for each category rather than training for individual models.

To this end, the optimal subsets for several models of different shapes will be computed in order to derive high level characteristics, as quantifiable measures, as to what properties are suitable for what types of objects (e.g. objects with many sharp or 90° angles, objects with many cylindrical parts, objects with smooth surfaces etc). Unsupervised machine learning algorithms will be used for the automatic classification task. Based on these results, a fast test will be devised which would process each new object and characterize its shape to index into a group database with pre-selected subsets. Another direction of future work involves analyzing the modes of failure of the algorithm (i.e. images were detection fails), in particular for subsets selected using the \textit{DCG} measure.

We are also investigating the possibility of extending this work to local descriptor selection for point matching in intensity or colour images. Combining range and intensity information is another goal of our future work. In the particular case of detecting the pose of the \textsc{RADARSAT} model, combining shape and appearance information is crucial in resolving the symmetric ambiguity in the shape of the object.

A learning-while-tracking strategy will be devised for the satellite pose acquisition and tracking application to improve upon the the probability of target pose re-acquisition. In the current version of our prototype, the model point cloud is generated by sampling a generic and idealized CAD model of the target object and, as such does not fully represent all the details in the shape and appearance of the actual object. For instance, the base of the satellite is covered with heat insulating sheets that are often wrinkled and provide an abundance of appearance-based information suitable for tracking. However, the idealized model cannot include the randomly formed wrinkles and approximates the base with smooth surfaces.
While this approximation is sufficient for the purpose of constructing purely shape-based local descriptors, it discards some appearance information that could be useful.

We intend to advance the system such that pose acquisition is initiated by matching the idealized model to input range images, but as tracking continues, the system would collect data on the appearance of the object. With each tracking frame, the 3-D model would be augmented with new appearance information to assist subsequent recovery of the target in the event it is lost, due to momentary occlusion or fast maneuvers for instance. Should the appearance or the shape of the target object change with time, e.g. if the insulating sheets shift and wrinkle in alternate ways or if the lighting condition changes, the algorithm should recognize this and update the appearance information. Means of determining such changes and dealing with them require further research.
Bibliography


[87] NextEngine. 3D scanners. [https://www.nextengine.com](https://www.nextengine.com), [accessed on Dec 7, 2008].


Appendix A

RANSAC

RANSAC, abbreviated from RANdom SAmple Consensus, is an iterative method for robustly estimating the parameters of a model from a set of data containing possible outliers. The algorithm was first published by Fischler and Bolles [32] in 1981 and has since become one of the most widely used algorithms in computer vision among other fields. A strong point of RANSAC is that it allows for probabilistic analysis on the required number of iterations through simple combinatorics.

In cases where observed data is polluted with outliers, standard model fitting methods such as least squares estimation are not applicable since their outcome is typically highly influenced by the outliers. RANSAC counters the outlier effect by taking an iterative approach and using the minimum number of required data points for model fitting at each iteration. At each step, a group of data points are randomly drawn from the complete set and a model is hypothesized by a standard fitting technique such as least squares using the selected points. The model is then verified by evaluating how well it fits the remaining data points, typically by way of classifying each point as inlier or outlier and counting the number of inliers. Through several iterations, the model which is confirmed with the largest
number of inliers is selected as the best model. A final least squares fit through all the inlier point is often used to reduce the residual errors.

The required number of iterations depends on the prior estimate on the percentage of inliers. If $n$ is the minimum number of points needed for standard model fitting, and $p_r$ is the initial estimate on the percentage of inliers, the probability of randomly drawing $n$ inliers, as needed for generating the correct model, equals $p_r^n$ and the probability of drawing at least one outlier equals $1 - p_r^n$. After $k$ iterations, the probability of not having drawn a group consisting only of inliers is $(1 - p_r^n)^k$. The assurance level $a$ is defined as the probability of obtaining the correct model within $k$ iterations and is computed as $a = 1 - (1 - p_r^n)^k$. Consequently, the required number of iterations given a desired assurance level is computed as $k = \frac{\log(1-a)}{\log(1-p_r^n)}$. If at any point during the process the percentage of inliers exceeds $p_r$, $p_r$ is updated and the required number of iterations is reduced accordingly.

In the specific case of estimating a rigid transformation from a list of tentative 3-D point matches, each point match constitutes a single RANSAC data point. The minimum number of point matches for least squares estimation of a rigid transformation is 3, i.e. $n = 3$. Throughout this thesis, we refer to this case as 3-point RANSAC. Figure 1.3 in Chapter 1 illustrated the required number of 3-point RANSAC iterations for achieving three different assurance levels plotted versus precision levels.

If extra information is available for each point, then fewer than three points might be required. For instance, if the direction of surface normals is known for the matched points, as in the case of Spin Images [55], then two point matches are sufficient for extracting a rigid transformation. Similarly, if a 3-D frame is associated with each point, as in the case of Tensor-Based Correspondence [78] or the VD-LSD method proposed in this thesis, then just a single point match is sufficient. We refer to these alternatives as 2-point and 1-point
RANSAC, respectively.

Given a hypothesized rigid transformation, a simple threshold on the Euclidean distance could be used for classifying remaining surface points as inliers or outliers. Since the rigid transformation is being verified as the one aligning two surfaces, the entire set of surface points, and not just the ones listed in the tentative point match list, could be used during the verification process.
Appendix B

Removing the Ambiguity in Direction Properties

Local frames of points $p$ and $p'$, i.e. $O_p$ with $O_{p'}$, are defined with rotation matrices $R_p$ and $R_{p'}$, centred at $p$ and $p'$ respectively, and defined as follows:

$$R_p = [\hat{i} \mid \hat{j} \mid \hat{k}] \text{ , and }$$
$$R_{p'} = [\hat{i}' \mid \hat{j}' \mid \hat{k}].$$

The rotation that registers $O_{p'}$ with $O_p$ is computed as:

$$R_{p,p'} = R_{p}^{T} \cdot R_{p'} = \begin{bmatrix}
\hat{i} \cdot \hat{i}' & \hat{i} \cdot \hat{j}' & \hat{i} \cdot \hat{k}' \\
\hat{j} \cdot \hat{i}' & \hat{j} \cdot \hat{j}' & \hat{j} \cdot \hat{k}' \\
\hat{k} \cdot \hat{i}' & \hat{k} \cdot \hat{j}' & \hat{k} \cdot \hat{k}'
\end{bmatrix}$$

That is, if a point $\vec{X} = [x, y, z]^T$ is expressed in the $O_{p'}$ frame, multiplying it by $R_{p,p'}$ (i.e. $R_{p,p'} \cdot \vec{X}$) is the expression of the same point in the $O_p$ frame.

ZYX Euler angles [100] roll, pitch, and yaw are computed as follows:
APPENDIX B. REMOVING THE AMBIGUITY IN DIRECTION PROPERTIES

\[ \text{roll} = \arctan2(R_{21}, R_{11}), \]
\[ \text{pitch} = \arctan2(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}), \text{ and } \]
\[ \text{yaw} = \arctan2(R_{32}, R_{33}) \]

where \( R_{ij} \) is the element on the \( i^{th} \) row and \( j^{th} \) column of the \( 3 \times 3 \) \( R_{p,p'} \) matrix, with \( i, j = 1, 2, 3 \). The range of feasible values for the angles is as follows: \( \text{roll} \in [-\pi, \pi] \), \( \text{pitch} \in [-\pi/2, \pi/2] \), and \( \text{yaw} \in [-\pi, \pi] \).

Similarly, the ZYZ Euler angles [100] \( \alpha, \beta, \) and \( \gamma \) are computed as follows:

\[ \alpha = \arctan2(R_{23}, R_{13}), \]
\[ \beta = \arctan2(\sqrt{R_{13}^2 + R_{23}^2}, R_{33}), \text{ and } \]
\[ \gamma = \arctan2(R_{32}, R_{31}) \]

The range of feasible values for the ZYZ angles is as follows: \( \alpha \in [-\pi, \pi], \beta \in [0, \pi], \) and \( \gamma \in [-\pi, \pi] \).

Due to ambiguities in the sign of eigenvectors, the direction of the \( \vec{i} \) and \( \vec{j} \) vectors (or the \( \vec{i}' \) and \( \vec{j}' \) vectors) might flip, affecting either the rotation matrix \( R_p \), the rotation matrix \( R_{p'} \), or both. For instance, if the direction of the \( \vec{i}' \) and \( \vec{j}' \) vectors is flipped, the \( R_{p,p'} \) matrix is affected as follows:

\[
R_{p,p'} = \begin{bmatrix}
-\vec{i} \cdot \vec{i}' & -\vec{i} \cdot \vec{j}' & \vec{i} \cdot \vec{k}' \\
-\vec{j} \cdot \vec{i}' & -\vec{j} \cdot \vec{j}' & \vec{j} \cdot \vec{k}' \\
-\vec{k} \cdot \vec{i}' & -\vec{k} \cdot \vec{j}' & \vec{k} \cdot \vec{k}'
\end{bmatrix}
\]

Similarly, the \( R_{p,p'} \) matrix could be re-written if the \( R_{p'} \) is affected by a flip or if both \( R_p \) and \( R_{p'} \) are affected. We emphasize that the \( \vec{k}' \) and \( \vec{k}'' \) vectors could not flip since the disambiguity in their direction is resolved using the surface normal direction or the viewing direction, as explained in Section 3.5. Enforcing right-handedness of the frames resolves another degree of ambiguity and therefore the \( \vec{i} \) and \( \vec{j} \) vectors could only flip simultaneously.

**Lemma:** The following direction properties are not affected by a flip in either \( R_p, R_{p'} \),
or both \( R_p \) and \( R_{p'} \). (The range for each property is also specified).

\[
\tilde{\text{roll}} = \begin{cases} 
\text{roll} & \text{roll} \geq 0 \\
\text{roll} + \pi & \text{roll} < 0
\end{cases} \quad \tilde{\text{roll}} \in [0, \pi]
\]

\[
|\text{pitch}| = |\text{pitch}| \in [0, \frac{\pi}{2}]
\]

\[
|\text{yaw}| = |\text{yaw}| \in [0, \pi]
\]

\[
\tilde{\alpha} = \begin{cases} 
\alpha & \alpha \geq 0 \\
\alpha + \pi & \alpha < 0
\end{cases} \quad \tilde{\alpha} \in [0, \pi]
\]

\[
\beta \in [0, \frac{\pi}{2}]
\]

\[
\tilde{\gamma} = \begin{cases} 
\gamma & \gamma \geq 0 \\
\gamma + \pi & \gamma < 0
\end{cases} \quad \tilde{\gamma} \in [0, \pi]
\]

\[
|\cos(\theta)| = |\vec{i} \cdot \vec{i}'| \quad |\cos(\theta)| \in [0, 1]
\]

\[
|\cos(\phi)| = |\vec{j} \cdot \vec{j}'| \quad |\cos(\phi)| \in [0, 1]
\]

\[
\cos(\psi) = \vec{k} \cdot \vec{k}' \quad \cos(\psi) \in [-1, 1]
\]

\[
\tilde{\theta} = \cos^{-1}(|\cos(\theta)|) \quad \tilde{\theta} \in [0, \frac{\pi}{2}]
\]

\[
\tilde{\phi} = \cos^{-1}(|\cos(\phi)|) \quad \tilde{\phi} \in [0, \frac{\pi}{2}]
\]

\[
\psi \in [-\pi, \pi]
\]

Here \( \theta \) is the angle between vectors \( \vec{i} \) and \( \vec{i}' \), \( \phi \) is the angle between vectors \( \vec{j} \) and \( \vec{j}' \), and \( \psi \) is the angle between vectors \( \vec{k} \) and \( \vec{k}' \).

**Proof:** The proof for properties derived from the vector angle properties (\( \theta \), \( \psi \), and \( \psi \)) is trivial. For Euler angle properties, for brevity, we only consider the case for a flip in \( R_p \) and the ZYX Euler angles. The proof for the other flip cases and for the ZYZ Euler angles is similar and equally simple.

\(^1\)While the theoretical range for \( \psi \) is \([ -\pi, \pi ]\), in most cases the value lies within the \([ -\frac{\pi}{2}, -\frac{\pi}{2} ]\) range. This is because the angle between the surface normals of points in a neighbourhood does not typically exceed 90°. This condition is sometimes enforced by applying the Support Angle criteria [55] as explained in Section 2.2.
Without any flips, the ZYZ Euler angles are computed as follows:

\[
\alpha = \text{atan2}(\vec{j} \cdot \vec{k}', \vec{i} \cdot \vec{k}'), \\
\beta = \text{atan2}(\sqrt{(\vec{i} \cdot \vec{k}')^2 + (\vec{j} \cdot \vec{k}')^2}, \vec{k} \cdot \vec{k}'), \text{ and} \\
\gamma = \text{atan2}(\vec{k} \cdot \vec{j}', \vec{k} \cdot \vec{i}')
\]

With flipped \( R_p \), the ZYZ Euler angles are computed as follows:

\[
\alpha_f = \text{atan2}(\vec{j} \cdot \vec{k}', \vec{i} \cdot \vec{k}'), \\
\beta_f = \text{atan2}(\sqrt{(\vec{i} \cdot \vec{k}')^2 + (\vec{j} \cdot \vec{k}')^2}, \vec{k} \cdot \vec{k}'), \text{ and} \\
\gamma_f = \text{atan2}(\vec{k} \cdot \vec{j}', \vec{k} \cdot \vec{i}')
\]

Therefore,

\[
\begin{align*}
\alpha_f &= \begin{cases} 
\alpha - \pi & \alpha \geq 0 \\
\alpha + \pi & \alpha < 0 
\end{cases} \\
\beta_f &= \beta \\
\gamma_f &= \gamma
\end{align*}
\]

and therefore, \( \alpha_f = \tilde{\alpha} \), \( \beta_f = \tilde{\beta} \), and \( \gamma_f = \tilde{\gamma} \) in the case of a flip in \( R_p \). \( \square \)
Appendix C

Wilcoxon Signed Rank Test

The Wilcoxon Signed Rank test (WSR) [69, 106] is a nonparametric statistical hypothesis test, which provides an alternative to the paired Student’s t-test [106] when the distribution of the data cannot be assumed to be normal. Given two sets of paired observations, the null hypothesis states that the median difference between the paired values is zero.

The test calculates the differences between paired samples and ranks the absolute differences ascendingly. The algebraic sum of all the ranks is a measure of the similarity between the two underlying distributions, where smaller sums indicate higher similarities.

To set the notation, let \( \{x_1, x_2, \ldots, x_n\} \) and \( \{y_1, y_2, \ldots, y_n\} \) be paired samples such that \( x_i \) is paired with \( y_i \) for \( i = 1, 2, \ldots, n \). In order to perform the test, the paired differences are first computed as \( \{d_1, d_2, \ldots, d_n\} \), where \( d_i = x_i - y_i \). The set of differences are then ranked from the smallest to the largest by their absolute values. That is, the sorting index \( \{r_1, r_2, \ldots, r_n\} \) is found such that \( |d_{r_i}| \leq |d_{r_j}| \) if \( i < j \). The signed rank for each \( d_i \) is computed as \( s_i = r_i \cdot \text{sgn}(d_i) \). The \( W \) statistic, equivalent with the sum of all signed ranks, is computed as \( W = \sum_{i=1}^{n} s_i \).

The null hypothesis implies that there is no tendency in either direction. In such cases,
the numbers of positive and negative signs will be approximately equal to cancel out each other and the resulting $W$ statistic will be zero or very small.

For cases where the number of paired samples is small (e.g. $n < 10$), critical values of $W$ for desired significance levels can be explicitly computed or looked up from appropriate tables, such as those presented in [69]. For large values of $n$, distribution of the $W$ statistic approaches a normal distribution with mean and variance values of $\mu_W = 0$ and $\sigma_W = \sqrt{n(n+1)(2n+1)/6}$ respectively. Statistic $z$, as a zero-mean unit variance normal distribution, is computed as $z = \frac{W - \mu_W}{\sigma_W}$. For desired significance levels, critical values of $z$ statistic can be found by referring to the normal distribution tables.

The WSR test assumes that there are no ties in paired differences. Tied scores are assigned a mean rank but special care is needed to adjust the $z$ statistic [69] in cases where ties exist, i.e. when $\exists i, j \mid d_i = d_j$. The test also assumes that the difference values $(d_i)$ are distributed symmetrically around the median. Furthermore, like many statistical tests, the WSR test assumes that the samples $(x_i$ and $y_i)$ are randomly and independently drawn. Deviations from either the symmetric or the independence assumptions reduce or compromise the validity of the WSR test.

We point out that the $\triangleright$ operator defined in Chapter 3 using the WSR test in the space of subsets is only transitive in a statistical sense. In other words, the WSR test does not define a metric in the space of DCG arrays since the triangle inequality does not hold. Such generalized metric space is sometimes referred to as a semi-metric space. To illustrate, if $f_1 \ominus f_2 = z_{12}$ and $f_2 \ominus f_3 = z_{23}$ are both large positive number (e.g. 2.0 and 1.8), indicating high certainly in $f_1 \triangleright f_3$ and $f_1 \triangleright f_3$, then $f_1 \ominus f_3 = z_{13}$ will almost certainly be positive. On the other hand, if $z_{12}$ and $z_{23}$ are both small positive numbers (e.g. 0.01 and 0.02), indicating nearly no confidence in $f_1 \triangleright f_3$ and $f_1 \triangleright f_3$, then $z_{13}$ will not necessarily be
positive. Since the goal of sorting a pool of subsets is to find the best ones, i.e. those that are significantly better than the others, this issue has minimal effect on the training results. We expected that this issue would have minimal effect on the training results.