OBJECT RECOGNITION BY REGISTRATION OF REPEATABLE 3D INTEREST SEGMENTS

by

JOSEPH LAM

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
February 2015

Copyright © Joseph Lam, 2015
Abstract

3D object recognition using depth data remains a difficult problem in computer vision. In this thesis, an object recognition system based on registering repeatable 3D surface segments, termed recognition by registration (RbR) is proposed. The goal is to eliminate the dependency on local geometry for establishing point-to-point correspondences, while maintaining the robust trait of a global technique by applying a pairwise registration process on these individual segments.

The extraction of repeatable surface segments is achieved by inheriting the high repeatability of interest points, most often utilized to increase the performance of matching local shape descriptors. Precisely, dense sets of interest points are connected to form the surface segment boundaries by greedily optimizing a smoothness constraint. The reconstructed boundaries provide an effective means to facilitate fast 3D region growing on the object and scene surfaces, forming the 3D interest segments.

Pairwise registration of the model and scene interest segments must then consider the imperfectly extracted segments due to data noise, occlusion, and error from the segmentation itself. An adaptation of the robust 4 points congruent sets (4PCS) registration algorithm was shown to register interest segments efficiently with great success. This is achieved by utilizing the prior knowledge of the 3D model interest segments that can be preprocessed, coupled with pose clustering of the retrieved
transformation candidates.

Experimentally, the interest segment repeatability, registration rate, and the object recognition rate were evaluated using a variety of free-form objects in 3D model data corrupted with synthetic noise and real 2.5D cluttered scenes. It was found that the interest segments are highly repeatable (> 80% per top segment per scene), and that they can also be registered successfully within a reasonable number of RANSAC cycle of the 4PCS algorithm. Compared to other state-of-the art local approaches, RbR enjoyed superior object recognition rates in both accurate LiDAR data and noisy Kinect data (on average > 90% for all objects tested in both sets of data), demonstrating that the approach provides a very attractive alternate solution to those in the current literature.
Acknowledgments

My supervisor Prof. Michael Greenspan has provided nothing but support and guidance throughout my pursuit of my PhD. He has never stopped believing in me and his endless encouragement helped made this thesis a success. It is truly a privilege to have known Michael, a mentor, and now a friend.

I would also like to express sincere gratitude to my beloved girlfriend, Elaine, who has always been there for me through ups and downs. I am so grateful I can always count on the love and support from my girlfriend.

Words cannot describe how much love and patience I have received from my parents. My parents raised me to become the man I am today, and their relentless support and encouragement means the world to me. I also want to thank my brother John, a fellow graduate student at Queens and now a professor, whom I shared much laughter with throughout our years in school together.

Last but not least, I would also like to express appreciation to my colleagues at Epson Canada, whom I have been working with throughout the last years of my PhD. They have created a fantastic work environment that not only inspires me, but also gives me the energy to flourish in an environment outside of academia.
# Contents

Abstract

Acknowledgments

Contents

List of Tables

List of Figures

Glossary

Chapter 1 Introduction

1.1 3D Object Recognition ................................................. 1
1.2 Current Paradigms for Object Recognition ............................ 4
1.3 A New Approach: Recognition by Registration (RbR) ............... 8
1.4 Summary of Contributions ............................................... 10
1.5 Thesis Organization ..................................................... 12

Chapter 2 Literature Review

2.1 Object Recognition with 3D Data ..................................... 13
2.1.1 Local Shape Descriptors for Establishing Correspondences .... 13
2.1.2 Using Local Features without Correspondences: A Statistical Approach ......................................................... 18
2.1.3 Global Shape Descriptors ............................................. 21
2.1.4 Discussion ............................................................. 23
2.2 Segmentation of 3D Surfaces ................................................. 23
  2.2.1 Segmenting 3D Surfaces from Range Images ................. 25
  2.2.2 Segmenting 3D Surfaces from 3D Meshes ..................... 27
  2.2.3 Segmenting 3D Surfaces from 3D Point Cloud ............... 29
  2.2.4 Discussion .......................................................... 31

Chapter 3 The Detection of 3D Interest Points 32
  3.1 Interest Point Detector Variants ...................................... 33
    3.1.1 Fixed-scale Detector ............................................ 34
    3.1.2 Multi-scale Detector .......................................... 36
    3.1.3 Difference-scale Detector .................................... 37
    3.1.4 Selecting the Appropriate Detector .......................... 38
  3.2 Evaluation: Repeatability of Interest Points ..................... 39
    3.2.1 Data Set and Parameter Selection ............................. 40
    3.2.2 Measuring Repeatability ...................................... 42
    3.2.3 Results and Discussion ...................................... 42

Chapter 4 Interest Segment Extraction 50
  4.1 Reconstruction of Interest Segment Boundaries ................. 51
    4.1.1 Estimate an orientation at the current interest point .... 53
    4.1.2 Find the next interest point to connect .................... 54
    4.1.3 Enforce a smoothness constraint ............................. 55
    4.1.4 Removal of Redundant Representation of Interest Curve ... 57
    4.1.5 Interest Segment Boundaries from Connecting Interest Curves 58
  4.2 Interest Segments Extraction from Fast 3D Region Growing ...... 59
    4.2.1 Edge and Non-Edge Classification ............................ 60
    4.2.2 Region Growing of Non-Edge Points .......................... 60
    4.2.3 Assigning Edge Points to Surface Segments ............... 61
    4.2.4 Merging Segments .............................................. 61
  4.3 Evaluation: Interest Segments Repeatability ..................... 62
    4.3.1 Measuring Interest Segment Repeatability .................. 63
    4.3.2 Experiments ..................................................... 64
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Interest segment repeatability of the final 8 segments for object angel with different surface sampling</td>
<td>70</td>
</tr>
<tr>
<td>4.2</td>
<td>Interest segment repeatability of the final 8 segments for object bigbird with different surface sampling</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>Interest segment repeatability of the final 8 segments for object bigbird with different surface sampling</td>
<td>72</td>
</tr>
<tr>
<td>4.4</td>
<td>Interest segment repeatability of the final 8 segments for watermelonKid with different surface sampling setting</td>
<td>73</td>
</tr>
<tr>
<td>4.5</td>
<td>3D model interest segment repeatability $\Psi$ of the angel object in 9 LiDAR scenes</td>
<td>76</td>
</tr>
<tr>
<td>4.6</td>
<td>3D model interest segment repeatability $\Psi$ of the bigBird object in 9 LiDAR scenes</td>
<td>77</td>
</tr>
<tr>
<td>4.7</td>
<td>3D model interest segment repeatability $\Psi$ of the gnome object in 9 LiDAR scenes</td>
<td>78</td>
</tr>
<tr>
<td>4.8</td>
<td>3D model interest segment repeatability $\Psi$ of the watermelonKid object in 9 LiDAR scenes</td>
<td>79</td>
</tr>
<tr>
<td>6.1</td>
<td>Object recognition success rates ($\Upsilon$) of different approaches in LiDAR cluttered scenes</td>
<td>115</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 Object recognition using 3D point cloud data .................................. 5  
1.2 An overview of the Registration by Recognition (RbR) framework ........ 10  

3.1 Salient feature selections using different $\frac{\lambda_3}{\lambda_1}$ threshold .......... 34  
3.2 Different depth quantizations of the same 3D model generated synthetically . 40  
3.3 Different data density of the same 3D model generated synthetically .... 40  
3.4 Repeatability vs. Sampling Density plots for DoN interest points detected on various objects. .................................. 43  
3.5 Repeatability vs. Depth Resolution plots for DoN interest points detected on various objects. .................................. 44  
3.6 DoN interest points detected on various objects. ............................ 45  
3.7 Fixed-scale PCA interest points detected on various objects. ............... 47  
3.8 Detecting Dense set of Interest Points in 2.5D scenes from different sensor modalities .................................. 49  

4.1 Intermediate steps of extracting interest segments from the 3D model of the gnome object and from a 2.5D scene containing the same object. 52  
4.2 Estimating the orientation of interest points using PCA ..................... 54  
4.3 Connecting interest points via direction estimated by PCA ................. 55  
4.4 Enforce a local smoothness constraint ........................................ 56  
4.5 Remove redundant interest points along a newly constructed interest curve .................................. 57  
4.6 Reconstructed interest segment boundaries for various 3D models ....... 58  
4.7 An illustration of region and boundary points classification for efficient 3D region growing .................................. 60  
4.8 Merging interest segments ........................................ 63  
4.9 Interest segment repeatability ($\Psi$) of different objects from different surface sampling vs. 100 pts/cm$^2$ .................................. 66  
4.10 Interest segment repeatability of different objects from different depth quantization vs. original data .................................. 67
4.11 Interest segment extracted for the angel object from data of different surface sampling. ......................................................... 70
4.12 Interest segment extracted for the bigBird object from data of different surface sampling. ......................................................... 71
4.13 Interest segment extracted for the gnome object from data of different surface sampling. ......................................................... 72
4.14 Interest segment extracted for the watermelonKid object from data of different surface sampling. ........................................ 73
4.15 Segmented 2.5D LiDAR scene data used in this experiment. ................. 74
4.16 Interest segment repeatability of different objects in different 2.5D scenes 75
4.17 Interest segments extracted for the angel object from 3D models and 2.5D scenes. ................................................................. 76
4.18 Interest segments extracted for the bigbird object from 3D models and 2.5D scenes. ................................................................. 77
4.19 Interest segments extracted for the gnome object from 3D models and 2.5D scenes. ................................................................. 78
4.20 Interest segments extracted for the watermelonkid object from 3D models and 2.5D scenes. ......................................................... 79

5.1 Quantizing point pair from interest segments for efficient 4PCS registration ................................................................. 86
5.2 Distribution of point pairs depending on the shape of the surfaces .......... 88
5.3 Average segment registration success rate $\xi$ for all interest segments vs the number of RANSAC cycles $L$ and the number of retrieved transformation $N_T$ ......................................................................................... 94
5.4 Average segment registration success rate $\xi$ of only the best aligned interest segments per scene vs the number of RANSAC cycles $L$ and the number of retrieved transformation $N_T$ ................................................................. 95
5.5 Selected examples of successfully registered model interest segments from each object in each tested scene ........................................ 96
5.6 Low registration success rate due to shape ambiguity ......................... 97
5.7 Average segment registration success rate $\xi$ of interest segments at different repeatability score vs the number of RANSAC cycles $L$ .......... 98
5.8 Average segment registration success rate $\xi$ of interest segments at different completeness percentage vs the number of RANSAC cycles $L$ 99

6.1 The Recognition by Registration framework ........................................... 102
6.2 Nine different 3D objects used in our experiments ............................ 104
6.3 Interest segments extracted for the nine 3D objects used in the experiment ................................................................. 107
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4</td>
<td>Interest segments extracted from selected 2.5D Kinect scenes used in this experiment</td>
<td>108</td>
</tr>
<tr>
<td>6.5</td>
<td>Time breakdown of the RbR system.</td>
<td>109</td>
</tr>
<tr>
<td>6.6</td>
<td>Results from recognizing various objects in 50 LiDAR scene using different number of RANSAC iterations ( (L) )</td>
<td>110</td>
</tr>
<tr>
<td>6.7</td>
<td>Results from recognizing various objects in 50 Kinect scene using different number of RANSAC iterations ( (L) )</td>
<td>111</td>
</tr>
<tr>
<td>6.8</td>
<td>Difficulty in handling nearly symmetric objects in the absence of distinctive object part.</td>
<td>112</td>
</tr>
<tr>
<td>6.9</td>
<td>Recognizing various objects in selected cluttered and occluded LiDAR scenes</td>
<td>113</td>
</tr>
<tr>
<td>6.10</td>
<td>Recognizing various objects in selected cluttered and occluded Kinect scenes</td>
<td>114</td>
</tr>
<tr>
<td>6.11</td>
<td>Recognition rate of Spin Images, VD-LSDs, and RbR due to varying degree of scene complexity in Kinect scenes</td>
<td>116</td>
</tr>
</tbody>
</table>
Glossary

**DOF**  Degree of Freedom  Page 2.

**DoN**  Difference of Normal  Page 11.

**GSD**  Global Shape Descriptor  Page 7.

**ICP**  Iterative Closest Point  Page 3.

**LSD**  Local Shape Descriptor  Page 5.

**PCA**  Principal Component Analysis  Page 16.

**RANSAC**  RANdom SAmple Consensus  Page 12.

**RbR**  Recognition by Registration  Page 9.

$\vec{n}_1$  The normal associates with $r_1$ in DoN estimation  Page 37.

$\vec{n}_2$  The normal associates with $r_2$ in DoN estimation  Page 37.

$r_1$  The smaller scale of radius in DoN estimation  Page 37.

$r_2$  The larger scale of radius in DoN estimation  Page 37.

$\Theta$  Difference of Normal angle threshold  Page 38.

$M_{org}$  3D model data of a 3D object with originally acquired sampling density and depth resolution  Page 40.

$M_{syn}$  3D model data of a 3D object corrupted by synthetic noise  Page 40.

$P_{M_{org}}$  The set of interest points detected in $M_{org}$  Page 42.

$P_{M_{syn}}$  The set of interest points detected in $M_{syn}$  Page 42.

$p_i$  Current interest point, potentially connected to $p_h$ and $p_j$  Page 53.
The set of interest points within the local neighbourhood of \( p_i \). The set of all interest points to be tested within \( P \). A neighbour interest point of \( p_i \), potentially connected to \( p_i \) and \( p_k \). A neighbour interest point that is potentially connected to \( p_i \). Smoothness constraint of constructing an interest curve. The final number of merged interest segment. The set of 3D points that are classified as close to any interest segment boundary. The set of 3D points that are classified as far away from any interest segment boundary. Interest Segment Repeatability. 3D model data of a 3D object at the resolution used in various experiments in this thesis. 2.5D scene data consists of various 3D objects used in various experiments in this thesis. A 3D point in the 2.5D scene data. The set of 3D point correspondences between \( M \) and \( S \). A 3D point that belongs to the set \( P_{MS} \). Number of points in \( R_S \). Interest segment extracted from a 3D model. Number of points in \( R_M \). Interest segment extracted from a 2.5D scene. A 3D point in the 3D model data. The reference and fixed 3D surface in pairwise surface registration.
$Q_{target}$ The target and moving 3D surface in pairwise surface registration Page 80.

$\varphi$ Interest Segment Completeness Page 80.

$T$ The 6 DOFs rigid transformation that align $Q_{ref}$ with $Q_{target}$ Page 82.

$\delta$ Euclidean distance error tolerance for the length of $\Pi_{pair1}$ and $\Pi_{pair2}$ Page 85.

$\{p_a, p_b, p_c, p_d\}$ The four points that consitute the coplanar base $\Pi$ Page 84.

$p_e$ The interesection point of $\Pi_{pair1}$ and $\Pi_{pair2}$ Page 85.

$\Pi$ A coplanar base extracted from $Q_{ref}$ Page 84.

$\Pi_{pair1}$ The first point pair $p_a, p_b$ of the coplanar base $\Pi$ Page 85.

$\Pi_{pair2}$ The second point pair $p_c, p_d$ of the coplanar base $\Pi$ Page 85.

$U$ The set of all bases extracted from $Q_{target}$ that is approximately congruent to $\Pi$ Page 85.

$ratio_1$ The ratio calculated from $||p_a - p_e||/||p_a - p_b||$ Page 85.

$ratio_2$ The ratio calculated from $||p_c - p_e||/||p_c - p_d||$ Page 85.

$\chi$ The radius size of searching for closet intermediate point $p_e$ in 4PCS Page 85.

$\Delta$ The neighbouring bins to search for point pairs during distance quantiza-

$Dist_{xy}$ Euclidean distance between any point pair $(p_x, p_y)$ in any model segment $R_M$ Page 86.

$(p_x, p_y)$ Any point pair sampled from a model interest segment $R_M$ Page 86.

$N_T$ Number of $T$’s to be returned for each interest segment registration event Page 93.

$\xi$ Interest Segment Registration Success Rate Page 90.

$T_{RP}$ The 6 DOFs transformation of retrieved pose from an object recognition system Page 103.

$T_{TP}$ The 6 DOFs transformation of a true positive pose of a 3D model in a 2.5D scene Page 105.

$\Upsilon$ The object recognition success rate in percentage Page 105.
Chapter 1

Introduction

1.1 3D Object Recognition

Object recognition in 3-dimensional (3D) space is a long standing problem in computer vision and machine intelligence, the goal of which is to detect an object of interest, while simultaneously localizing its position and orientation in 3D. Specifically, this work focuses on recognizing a solid 3D object for which a surface model exists, and retrieving its rigid transformation in possibly cluttered and partially occluded scenes.

Recognizing and retrieving the pose of a 3D rigid object is of particular interest for a variety of potential robotic applications. These include improving daily tasks such as recognizing and picking up a hot coffee mug in the kitchen [21] for a preoccupied person or a person with a disability. A prime example of object recognition is industrial assembly line [31], where the routine process of recognizing parts and assembling them can be replaced with robots. While the primary benefit is to increase the productivity of the assembly line, it also allows humans to focus on more
cognitive-orientated tasks such as problem solving and decision making. Lastly, high profile missions such as satellite retrieval [90] where the operational space is hazardous for humans, can be benefited by deploying an un-manned space vehicle with the capabilities of recognizing and retrieving the satellite within the castoff space debris.

A common methodology to this difficult problem is model-based object recognition, where the target object instance is known and represented by a model in the database. Formally, given this known model of the object and its frame of reference, 3D object recognition retrieves the $4 \times 4$ transformation matrix that describes the 6 degree-of-freedom (DOF) pose, both translation $(X, Y, Z)$ and rotation $(R_x, R_y, R_z)$, of the model with respect to the 3D sensor frame.

Different data types exist to represent both the model and scene data. For instance, the model can be multiple photograph-like images of an object from different viewing angles, or it can be represented by densely sampled points of the object’s 3D surfaces. Effectively recognizing these various representations of model and scene further rely on the matching of descriptors [16]. Precisely, a descriptor can either locally encodes geometric relationship between points within a small neighbourhood into compact vectors that can be used to facilitate efficient point-to-point matching, or globally encodes the entire object using shape contours or point distribution into some statistical representation to match the entire object. In the current literature, model-based object recognition is therefore commonly cast as solving the problem of extracting and matching model and scene descriptors. The retrieved object pose can be further refined by registering corresponding points belonging to the objects in both the model and the scene using pairwise algorithms such as the Iterative Closest
Intensity (2D) vs. Range (3D) for Object Recognition

Two primary sources for computer vision system are intensity images and range data. Intensity images are photograph-like images, also referred as 2D images because they consist of a 2D array function \( f(u, v) \) where each pixel indexed by the \((u,v)\) coordinate stores the intensity value indicating the amount of light reflected off any surface in the scene. Descriptors for object recognition based on 2D images can rely on attributes derived from the light intensity data (e.g. image gradient magnitude and orientation), to encode information such as surface texture and surface discontinuity. Numerous intensity-based methods have been developed and have achieved great success [9, 37, 60, 84]. However, a few limitations remain in using 2D images, such as changes in illumination that causes fluctuation in the perceived intensity values and also adds unwanted shadows. Detection also can fail when object scale is changed substantially due to the movement of the object along the sensor optical axis.

Range data, on the other hand captures the distance of any arbitrary point in the scene with respect to the camera origin. Usually only its 3D point coordinates \((x, y, z)\) with respect to the camera origin are stored (hence the name 3D data), although some sensors can also acquire intensity data. Range data can be acquired using various technologies, including stereo vision, structure-from-motion (SFM) [81], time-of-flight [79] and structured light sensors [26]. Range data can either be stored as a range image by storing the measured depth at each pixel of \( f(u,v) \) similar as a 2D image (detailed definition in [38]), or it can also be stored as a list of unorganized 3D point coordinates in the form of 3D point cloud. With the recent advances in a
variety of more affordable and accurate consumer range sensors, there is a renewed interest in the community [72] to explore techniques that utilize 3D shape and local surface geometry for object recognition [3, 22, 27, 35, 44, 59, 68, 71, 82, 91, 92, 93, 98].

Since 2D and 3D data provide two independent domains of information, recent research such as that in LINEMOD [36] utilizes both domains, by applying 3D surface normals to improve the shape matching cost previously based only on intensity gradients [37]. Their results showed promising improvement when 2D and 3D information are combined effectively. In this work, we continue to untap the potential in 3D data, by exploiting the abilities to segment 3D surfaces in a repeatable fashion and to match individual 3D surface segment to drive a robust object recognition system. This document will focus mainly on comparing the technology proposed here with those that are based on pure 3D data.

1.2 Current Paradigms for Object Recognition

In general, the design and matching of descriptors for object recognition can be categorized into: (1) the local approach that is based on local distinctive feature matching, and (2) the global approach that relies on global shape matching. Designing a robust descriptor to handle object recognition based only on 3D data faces a number of challenges. As demonstrated in Figure 1.1, these include dealing with the differences in depth resolution and data sampling frequency between the model and the scene. The scene data is often an incomplete 3D reconstruction, also known as 2.5D data since it is collected from a single viewpoint. Therefore, the object is self-occluded while the other surfaces visible to the camera can also be occluded by cluttered background, which may lead to confusion in detecting the correct object.
The local approach to resolve many of these challenges is to rely on establishing one-to-one correspondences between points in the model and points in the scene. This is done by matching descriptors constructed from a small local neighbourhood at each point of both data sets. In general, local descriptors facilitate matching by encoding local geometry information into multi-dimensional vectors [20, 27, 44, 91, 92, 93]. Since local descriptors using 3D data rely on extracting shape properties, they are commonly referred as local shape descriptors (LSDs). However, LSDs constructed at geometrically non-informative regions such as points on a flat 3D plane are ambiguous.
and cannot be used to facilitate one-to-one matching.

To avoid this ambiguity, an effective strategy is to localize relevant regions and eliminate non-informative points from the data sets. The detection of these relevant regions often depends on detecting a specific response by applying a local kernel/filter/detector on the data. For example in the 2D domain, relevant regions include significant change of pixel intensity, such as corners, edges [34], and intersections [39]. Similarly, in the 3D domain, detectors can extract regions or points responding to significant changes along the 3D surfaces or manifolds such as a peak or a valley, etc. [40, 63, 67, 74, 86, 97]. Although the definition is still evolving as the literature continues to mature, these detectors are commonly referred as the *interest point operators* or *keypoints detectors*, wherein the points extracted are simply called interest points or keypoints. In addition, the evaluation of interest points were also studied in the literature [63, 77], where two important traits were identified as important quality measures of an interest point operator. The first trait is *distinctiveness*, measuring how distinguishable a point is within a region from a point from another region. The second trait is *repeatability*, measuring how robust the detector is in localizing the same point despite corruption of the data, such as noise, signal deformation, or data occlusion.

Given LSDs are only extracted at the most distinctive local region and that they can also be repeatedly extracted from the same location across different data, this can significantly increase the matching accuracy of LSDs. A 6 DOF's rigid transformation that represents the object pose can then be estimated from these correspondences (minimally three correspondences are needed). Since the pool of distinctive point
correspondence candidates contains some incorrect matches (i.e. outliers), a statistically robust algorithm such as RANSAC [24] or a Hough variant [8] can be utilized to generate and verify rigid transformations that align 3D models with the 3D scene image.

A second approach is the global approach, in which global shape descriptors (GSDs) uniquely encode the shape of a 3D object using one single descriptor. A GSD represents an object as one single descriptor, but it is also ideal for tracking objects due to lower processing requirements. For example, Shang [82] proposes Potential Well Space Embedding (PWSE) that is able to recognize pose and track an object in real time. Local shape descriptors are computationally expensive because recognition is based on finding a one-to-one correspondence for the entire point cloud. LSD performance also depends on the size of the set and the number of models registered in the database. On the other hand, GSDs solve the correspondence problem by registering the shape information on a common platform. Recognizing a 3D object by solving the point-to-point correspondence problem is reduced to the problem of matching a model descriptor in a database with one single scene descriptor.

Different representations have been developed in the literature to uniquely define a GSD. Unlike LSD matching that follows a common design scheme, GSD-based technique in contrast do not follow any specific scheme in the current literature. Some of the techniques depend partially on local features in the process of forming a global descriptor [22, 30, 62, 71, 68, 98]. Moreover, some depend completely on matching views of the scene with views of the object registered and stored offline in a database [3, 14, 15, 59, 41, 82, 66]. Since GSD is invariant to small perturbation at a local shape level, it is also ideal for solving the problem of object class retrieval,
which is the problem of classifying and recognizing a class of object (e.g. car, plane, or cup) instead of an object instance. In this thesis, our focus is the former problem of retrieving an object instance learnt from a specific 3D model.

**Current Limitations**

A limitation of the local approach is that a single set of LSDs cannot possibly capture the local geometry of a large set of objects of varying geometries. One option is to treat LSD selection as an optimization problem [91], wherein the descriptor properties are carefully designed and tuned for a specific object. Nevertheless, LSDs eventually suffer if objects lack distinctive geometric features, making them unreliable to solve one-to-one correspondence matching. LSD-based recognition is also sensor dependent, since the effectiveness of matching LSDs are sensitive to range image characteristics such as noise distribution and resolution. The global approach in contrast is immune to the problem of resolving correspondence ambiguity, and thus is more resilient to noise and some local deformations of the 3D data. However, the downside is that the global approach requires uncluttered or correctly segmented scenes, as outliers drastically reduce the effectiveness of this approach.

### 1.3 A New Approach: Recognition by Registration (RbR)

Rather than relying on matching descriptors to achieve object recognition, this thesis proposes a novel approach by drawing inspiration from two points of inquiry.
1. Is it possible to inherit the repeatability of interest points, that are more commonly used in selecting the locations of LSDs, in a more direct manner rather than using it as a points filtering process?

2. Since the performance of local features is affected by lack of local distinctiveness while global descriptors are sensitive to object occlusion, can these limitations be reduced by utilizing a registration by parts approach?

Precisely, the problem of registration usually applies to finding the transformation between two different views of the same object. However, if a 3D surface segment can be extracted repeatably from both the model and the scene, then object recognition can also be formulated as a registration problem by registering 3D surface segments. The result of combining these two observations is the Recognition by Registration (RbR) framework [55], where a segmentation algorithm that generates 3D interest segments by directly inheriting the repeatability of 3D interest points is first developed [53]. Subsequently, these repeatable interest segments, each representing a local region of the object surface, can now be used to facilitate 6DOFs pose retrieval through data registration [54]. An overview of major components of the RbR framework is illustrated in Figure 1.2, where the objective is to identify the 3D point cloud model of the angel object in the 2.5D scene acquired by a LiDAR (a time-of-flight sensor). As this report will use the 3D point cloud from a list of objects to illustrate various concepts, please refer to Figure 6.2 in chapter 6 for their corresponding intensity images.

Compared to both local and global descriptors, RbR offers a number of advantages. By expanding the restricted neighbourhood of a local descriptor around a
single interest point to an entire surface segment, local matching becomes more robust to data deformation. In addition, unlike global techniques, these segments are still local relative to both the scene and model data. Registration of these segments therefore remains effective against clutter and partial occlusion. Except for cases where the surface of the objects are mostly planar and ambiguous, in which case a more primitive-shape oriented method is required, minimally just one of the distinctive interest segments needs to be repeatably extracted and registered for a given object viewpoint, in order to yield a successful recognition event.

1.4 Summary of Contributions

Object recognition by registering interest segment can be broken down into three major components. The contributions of this thesis related to each of these components are summarized as follow:
1. The concept of a RbR framework is introduced, implemented, and evaluated. The framework consists of an offline stage in pre-processing each 3D model into its respective interest segments. These segments are matched exhaustively during the online stage with those interest segments extracted from the scene data. A re-projection verification based on evaluating retrieved poses after pose clustering, is utilized to retrieve the best 6 DOF pose. Extensive experiments were carried out in evaluating the system performance against different variables that include segment repeatability, number of iterations required for registration, and scene complexity. Two different LSDs, the Spin Images [44] and the VD-LSDs [91] were also implemented and compared with our framework.

2. A novel technique to repeatably segment surfaces of 3D point cloud data repeatably is introduced. The key here is maintaining the repeatability of segments across different data sets. The algorithm starts by extracting dense set of repeatable interest points via the Difference of Normal (DoN) interest operator [40]. Interest segment boundary reconstruction, by first joining these DoN interest points followed by an effective method for region growing based on 3D point cloud data, is then used to generate repeatable interest segments. An evaluation of selecting the appropriate the interest point detectors, and measuring segments repeatability are performed using different 3D models in both synthetic 3D model data and real 2.5D scene data.

3. A fast and accurate 3D shape registration for aligning model and scene interest segments. The motivation here is optimizing the performance of a registration algorithm to better suit the requirement of registering interest
segments for the purpose of object recognition. Here the prior model is ex-
ployed, by pre-computing the Euclidean distance between any point pairs from
model interest segments. This distance information is stored and results in a
more efficient registration process during run-time with little expense of regis-
tration accuracy. The strategy is implemented using the 4PCS [2], a RANSAC
[24] (also see Appendix B) style pairwise registration method. An evaluation of
the accuracy in registering interest segments using different 3D models in both
synthetic 3D model data and real 2.5D scene data is provided.

1.5 Thesis Organization

This document continues by first reviewing various model-based object recognition
techniques in Chapter 2, followed by discussions of different segmentation techniques
based on 3D data. The detection of interest points is crucial to the extraction of
repeatable interest points for the RbR framework. Various 3D interest operators are
therefore reviewed and their performance are analyzed in Chapter 3. In Chapter
4, a detailed explanation of the algorithm in extracting interest segments based on
interest points is introduced, and its segmentation repeatability is also evaluated. The
registration of these repeatable interest segments is then studied in Chapter 5, while
the accuracy in interest segment registration is again evaluated based on the results
obtained in the previous chapter. Chapter 6 then presents the implementation detail
of the RbR framework that combines the proposed segmentation and registration
algorithms, with detailed experiments using data collected from both accurate LiDAR
and noisy Kinect sensors for a variety of rigid objects with free-form surfaces. This
thesis ends in Chapter 7 with a conclusion and discussion of possible future work.
Chapter 2

Literature Review

In this section, different state-of-art solutions for model-based object recognition will be reviewed. The proposed RbR object recognition framework is based on registering repeatable interest segments, and therefore different segmentation algorithms for 3D data will also be covered.

2.1 Object Recognition with 3D Data

2.1.1 Local Shape Descriptors for Establishing Correspondences

The paradigm of model-based object recognition that relies on LSD matching follows a 3-phases framework:

1. the establishment of point-to-point correspondences via LSDs;

2. the hypothesis and verification generation of a rigid transformation using the correspondences;
3. and finally pose refinement via ICP.

Many efforts were devoted to the first phase of this framework that requires a compact but effective LSDs [20, 44, 70, 69, 71, 87, 91, 92, 93, 101]. The reasons behind this are two-fold. First, the matching efficiency of the recognition system increases with the complexity of LSDs. Second, research in LSD driven by 3D shape drew inspiration from recognizing objects from 2D images using 2D feature descriptors, which have reached a high level of maturity and success in recent years. As there were countless approaches proposed in the literature, selected LSDs that represent various key design concepts in the literature will be discussed. For clarity in the following discussions, a point at which each LSD is generated will be denoted as \( p \), and any point that is within the supporting local neighbourhood of \( p \) is denoted as \( p_i \).

A first approach to constructing a LSD is to encode geometric measurements into a compact signal. For example, the **point signature** developed by Chau et al. [20] is a distance profiler that measures the perpendicular distance between each \( p_i \) that lies on the intersection of a sphere surrounding \( p \) to an estimated tangent plane at \( p \). A starting position of the 1-D distance profiler is defined so that the signature is orientation invariant, determined using the vector that has the longest distance from a 3D surface to the tangent plane. When matching point signatures, robustness can be increased by consider a phase-shifted tolerance \( \epsilon_{tol} \) to the signature \( d_s(\theta) \) from the scene, when comparing with a candidate signature \( d_m(\theta) \) from the model.

The concept of point signature was later extended in the work by Sun, known as **point fingerprint** [87]. Multiple contours are sampled at various radii at each \( p \), instead of at a single intersection in the point signature. The projection of these
multiple 3D contours onto a 2D tangent plane forms a map resembling a human fingerprint. For $p$ that belongs to a flat surface, its point signature is no more than concentric circles with regular spacing between them. This provides little information for distinctive point correspondence matching. Thus, non-discriminating point fingerprints can be filtered by selecting only those that exhibit a greater irregularity measure.

A second type of LSD is to count the number of $p_i$ with respect to $p$ that satisfy some geometric or topological measurements, and then store them as an accumulator such as histograms for comparisons. One of the most widely recognized LSD for 3D vision is the spin image [44] developed by Johnson and Herbert. Each spin image is a 2D local histogram representation of a 3D object generated with respect to a specific $p$ on an object. A 2D cylindrical coordinate frame ($\alpha$ and $\beta$) “spins” around the normal at $p$, and the number of points within the defined local point neighbourhood that fall into its respective coordinate is recorded. Each entry of the 2D histogram can be treated as a pixel value in a 2D image, hence the name spin image. A drawback of LSDs based on histograms is that their performance can be affected by bin quantization. A histogram whose bins are coarsely quantized can negatively impact the LSD’s matching accuracy. On the other hand, if it the bin is finely quantized, the LSD’s performance can be easily affected by data noise and object parts occlusion. To further reduce the dimensionality of spin images, Eigen-analysis can also be applied to compress spin images into lower resolution 2D images, removing redundant pixel information without greatly compromising the descriptiveness of the LSD.
A variant of spin images is the surface signatures proposed by Yamany [101], where the cylindrical coordinates are replaced with a coordinate defined by \(d_i\), which is the distance between a \(p\) to \(p_i\) on the surface, and \(\bar{\alpha}\) which is the angle between the normal at \(p\) and the vector from \(p\) to \(p_i\). Hence, instead of having an image where each pixel is an indication of the point density of a neighbour, the curvature at \(p\) is recorded. Non-descriptive points are simply represented by the surface signatures that have low curvature on sharp jumps in curvature (i.e. noise).

Neither spin images nor surface signatures precisely answer the question: “which geometric measurement is a better property for histogram-based LSD?” This issue was addressed by the Variable Dimension LSDs (VD-LSDs) [91] proposed by Taati and Greenspan. Instead of choosing a specific geometry, VD-LSDs argued that using an enumeration of geometric properties, an optimization process can be used to formulate a subset of properties tailored for the best performance for a specific object such that LSDs take on different dimensionality depending on the shape of the object. VD-LSDs are constructed by assigning a local and orientation invariant coordinate frame for \(p\) using Principal Component Analysis (PCA, also see Appendix A for a more detail description), then a family of geometric properties that include the 3D position \(x, y, z\) of any \(p_i\) relative to the PCA frame of \(p\); the inner angles \(C_\theta, C_\phi, C_\varphi\) between the two PCA frame of \(p\) to \(p_i\); and the dispersion Eigenvector lengths \((\lambda_1, \lambda_2, \lambda_3)\) of each axis of the PCA frame at \(p\) are calculated. The tuning process compares the performance of matching the up to 9D histograms under various matching scenario, and only the dimension(s) that have the highest matching score are selected to be the VD-LSD vectors designed for a specific object. The argument behind VD-LSDs is that even though the high dimensionality of LSD is generally not
preferred, it nevertheless provides more reliable and robust point matching, especially in nosier and sparse data, which in turns reduces the RANSAC cycles required in candidates generation.

Another take on comparing features that spanned across different dimensions of histograms (e.g. Spin image = 2D, VD-LSDs = max of 9D) is to effectively combine various features into a single 1-D histogram, such as that proposed by Rusu called **Point Features Histogram (PFH)** [70]. In the original design, the PFH encapsulates 4 different features \( f \) extracted at any point neighbourhood, such as the angle between the viewpoint vector to the surface normal at \( p \) or the Euclidean distance between \( p \) to \( p_i \), and so on. Using a pre-set threshold for each of these features, each feature is in turn transformed into a binary feature and then combines into a single 1-D histogram (therefore 16 bins). Let \( f_i \) and \( s_i \) be the features and the corresponding predefined threshold, then the PFH is computed as:

\[
idx = \sum_{i=1}^{i \leq 4} \text{step}(s_i, f_i) \cdot 2^{i-1}
\]  

where \( \text{step}(s, f) \) is defined as 0 if \( f < s \) and 1 otherwise.

Rusu later improved the efficiency of PFH, called **Fast Point Feature Histogram (FPFH)** [69] by employing a more intelligent scheme in selecting its \( k \) nearest neighbours of \( p \), followed by a re-weighting of the resultant histogram of a point with the neighbouring histograms, thus reducing the computational complexity from \( O(n^2) \) of PFH to \( O(kn) \) of FPFH. A more efficient processing of the PFH opens the door to a more heavy-weight feature, such as the **Viewpoint Feature Histogram (VFH)** [71] that stores a histogram of the angles that the viewpoint direction (to the centroid of the 3D data) makes with each normal on the 3D surface. In total this requires 263 bins. A carefully designed engineering solution that combines...
pre-segmented scene data by removing table planes and by first clustering remaining point cloud allows VFH to achieve real-time applications on object recognition, using sets of IKEA kitchen-ware demonstrated on a PR2 robot.

### 2.1.2 Using Local Features without Correspondences: A Statistical Approach

The use of local features for establishing correspondences is robust to occlusions, but can be time consuming since it requires finding the correct one-to-one matching and the subsequent search for the best rigid transformation that excludes outliers. An alternate approach to utilize local features that avoids establishing correspondences is to employ a representation that statistically describes the object using local features [22, 30, 62, 68, 98]. Such a statistic representation can increase matching efficiency, and it also maintains robustness in dealing with local shape. Local features used in this way usually lack the descriptiveness to facilitate point-to-point matching, but on the other hand are much light weight descriptors that can be efficiently computed and compared. For instance, a heavy weight LSD like spin image in fact is a 2D image containing a set number of pixels, which depends on how much information the user decide to keep in the shape descriptors. The matching of a 2D image is computational expensive. On the other hand, light weight local descriptors only contain a few numbers of variables at a local location, such as its surface normal and depth value (to be explained in the rest of this section), making it much lighter and computationally efficient for matching but at the same time less descriptive.

An early work by Hetzel [30] is the multi-view approach based on matching histograms composed of various local features. Each view is expressed as three separate
histograms, each describing a distribution of an independent local feature: 1) pixel depth, 2) surface normal, and 3) surface curvature. Let $Q$ and $V$ be the histograms describing any of these local features generated from the model and the scene respectively. Matching the distribution between $Q$ and $V$ can achieved by either measuring a histogram intersection:

$$ \cap(Q, V) = \sum_i \min(q_i, v_i) \quad (2.2) $$

or using a $\chi^2$-divergence test [76]:

$$ \chi^2(Q, V) = \sum_i \frac{(q_i - v_i)^2}{(q_i + v_i)} \quad (2.3) $$

where $q_i$ and $v_i$ are the values contained at the $i^{th}$ bin of $Q$ and $V$. Alternatively, a more robust method such as a posteriori probability estimation can be used, to better handle imprecision due to view sampling or partial occlusions.

A commonly used light local feature is the surflet-pair, comprising a pair of 3D points consisting the point of interest $p$ with any point $p_i$ within a local neighbourhood. Each point is associate with its respective orientation $\vec{n}_p$ and $\vec{n}_{p_i}$. The surflet-pair is also both translation and rotation invariant. Wahl et al. [98] proposed the use of a surflet-pair-relation histogram, measuring the occurrence of surflet-pairs based on the Euclidean distance between $p$ and $p_i$, and the angle difference between $\vec{n}_p, \vec{n}_{p_i}$, or other derived attributes. Contrary to the multi-view histogram approach by Hetzel [30], only a single histogram is generated to encapsulate the global shape information of an object. This assumes that recognition is based on comparing a full 3D model with full 3D scene data. A 4D histogram is then constructed based on four parameters (three angles and one distance) derived from the surflets. In order to cope with the large 4D space, the parameter space is further quantized coarsely.
A voting strategy based on the point pair surflet was proposed by Drost [22]. During the off-line stage, a global description $H$ of the object in the form of a hash table is created. A feature vector $F = (||\vec{d}||, \angle(\vec{n}_p, \vec{d}), \angle(\vec{n}_{p_i}, \vec{d}), \angle(\vec{n}_p, \vec{n}_{p_i})))$, where $\vec{d} = p - p_i$ is calculated from each surflet, this is the hash key to a table that stores every surflets that share the same feature vector $F$. During on-line stage, a set of interest points in the scene is selected (randomly in the paper). For each interest point, all other points in the scene are paired with the interest points to create point pair features. The feature vector to these scene features are matched to the model features of $H$ and a set of potential matches is retrieved. Every potential match votes for an object pose, and the pose with the highest matched features is returned as the optimal object pose.

In Papazov’s work [68], a more straight forward approach using the point pair surflet with RANSAC is utilized. First during off-line training, any surflet where both points are within a distance $d$ are selected.Similar in concept to that of Drost’s approach [22], the surflet feature vector $F = (\alpha, \beta, \angle(\vec{n}_p, \vec{n}_{p_i}))$, where $\vec{n}$ is the normal, and $\alpha$ and $\beta$ are the angles of the surface normals to the difference vectors between $p$ and $p_i$ respectively. $F$ is served as a hash key to a table storing the two corresponding model surflet points. During on-line recognition, a surflet pair that satisfies the $d$ constraint is extracted, and $F$ is computed from the surflet and used as the key to find all potential surflet model points. A RANSAC approach iterates this process $N$ times and the best transformation $T$ with the most aligned points is recovered as the best object pose.
In the density-based nonparametric framework proposed by Akgul et al. [3], multivariate samples of local features over the object surface are represented as an underlying probability density function, which is similar in concept to VD-LSDs previously discussed [91]. 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 descriptor. Unfortunately, the method falls short of investigating all possible feature subsets (26 in total) to determine the best combinations, and opts for handpicking three subsets.

2.1.3 Global Shape Descriptors

The difference between Local Shape Descriptors and Global Shape Descriptors (GSDs) [14, 82, 66] is that the latter aims at encapsulating the entire shape of an object, which is suitable for fast comparison of possibly a sparse set of 3D data that is ideal for real-time applications. For GSDs, initial segmentation of the object from the scene data is necessary, in order to facilitate a global comparison between the model and scene data.

Osada et al. [66] proposed a simple but effective method in globally encapsulating a model into a 1D shape distribution. Each value on the shape distribution is computed by sampling model surface points fed into a shape function. A variety of shape functions were tested in the work, and the $D_2$ function that measures the distance between two random points on the surface was empirically chosen as the most effective shape function. Each shape distribution is represented as a 1D histogram that is invariant to rigid transformation and could become scale invariant by a simple normalization. Shape distribution captures the general shape of a model; therefore it
is an effective object class recognizer. This was demonstrated in their experiments, where only 10 distinct shape distributions can be observed from 70 models, in which there is only 10 different object types (i.e. classes), and each having 7 variants. Using 133 models (some belonging to the same class), shape distribution achieved about 66% classification accuracy.

To be able to handle the matching of mechanical parts where they tend to share similar global features, Ip et al. [41] extended the shape function by also considering the geometric properties of the line when connecting a pair of points. Specifically, 3 separate shape histograms are constructed for lines that lie fully within the model, partially inside the model, or fully outside the model, respectively. The motivation here is that the matching of a specific model can be more discriminative by relying on the matching of 4 different shape distribution histograms. However, only minimal improvement over the original shape descriptors was reported in the same work.

A novel multi-view global approach based on ICP minima was introduced by Shang and Greenspan [82]. The concept is based on the well known fact that when aligning two different surfaces of specific position and orientation, ICP will always drive the registration error to the same local minima. Hence, during training, the position of each 2.5D view of the 3D model can be perturbed to different locations, and then aligned with a generic surface through ICP where all local minima are recorded and form a unique error surface function, named **potential well space embedding (PWSE)**. At run time, a centroid is estimated at the scene data and the same operation is performed to generate the run-time PWSE error surface. The matching of a PWSE error surface between the model and the scene is efficient, achieving an accuracy of 97% for a database of 60 object instance while executing at
122 frames per second on standard hardware. Similar to most GSDs, PWSE is also an effective object class recognizer, achieved successful results against the Princeton shape benchmark with a 96% recognition rate.

2.1.4 Discussion

Numerous LSDs and GSDs were covered in this section, and while LSDs have achieved great success in various 3D data including cluttered scenes, or partially occluded object parts, the well known disadvantage remains its dependency on local geometry that requires a relatively accurate 3D data. On the other hand, GSDs are robust to data deformation, but requires prior segmentation of the data, and sometimes also lack the distinctiveness distinguishing two similarly shaped objects. In general, GSDs are also more suitable for real-time application since it does not require the exhaustive establishment of point-to-point correspondences, which can be computationally expensive especially in the case of lack of distinctive feature in the model.

These observations are the motivation that inspire the recognition by registering segmented object parts proposed in this thesis. We will now move on to discuss various segmentation methods in the literature that aim to segment 3D surfaces in a reliable and repeatable fashion.

2.2 Segmentation of 3D Surfaces

Segmentation is the process of partitioning image into different elements by measuring the similarity within the data [88]. The literature of image segmentation is vast and there is no one specific algorithm that can solve all segmentation problems. To
name a few categories these include: thresholding [57, 40], contour-following [100], classification and region growing [10, 17, 43, 38, 23, 65], shape fitting [6, 25, 64, 102], graph-cut and energy minimization [13, 49], and many more [56, 85].

In general, segmentation is treated as an early vision task where the output segments are the input data for more advanced reasoning such as object detection, recognition, or target tracking. For example, context-based segmentation can utilize constraints imposed by known man-made structures such as road connectivity, to partition data from massive 3D urban data [56]. The segmented output can be later fed into an object classification framework [7]. Recent works also looked into jointly solving both segmentation and part labeling by training the system probabilistically (e.g. Conditional Random Field (CRF)[51]) from a collection of similarly segmented and labelled data [45, 48, 84].

Unlike systems that use segmentation to divide the data into classified region, for example, a 3D human model can be segmented and labelled as arms and faces, the motivation in this thesis is to provide highly repeatable segments from 3D surfaces, in which these segments may not necessarily contain any semantic meaning. They are instead required to facilitate robust registration of various object surfaces across different 3D data. The main characteristic of these segments is therefore that they are repeatably extractable, and robust to various expected conditions such as resolution, noise, and viewpoint. Robustness to occlusion is not a criterion, as will be revealed. The rest of this section will focus on reviewing various techniques designed specifically for segmenting 3D surfaces.
2.2.1 Segmenting 3D Surfaces from Range Images

Early investigations of 3D surface segmentation are based on range images (as defined in section 1.1), since they were more structured compared to other 3D data representations. Some algorithms depend on the 2D array structuring of the range image representation. A main focus was placed on objects composed of planar and quadric surface [16], where they can be segmented through surface parameterization or by second derivative analysis of 3D surface geometry.

By identifying and indexing the segmented surface primitives or classified surface types, objects of restricted surface types can be successfully detected and localized [10, 12, 38, 23, 102]. The method proposed by Hoffman and Jain [38] argued that by segmenting local surface patches and coarsely classifying them into only three categories of surfaces that are planar, concave, and convex, this approximate classification can be more robust to added noise to the 3D surfaces. Their work was inspired by the “clustering then region growing” method for 2D images. During the clustering stage, surface normals are estimated for each pixel using a least square plane fit at each $5 \times 5$ pixel window. A square error criterion based on the surface normal clusters neighbouring pixels into individual local patches. During the region growing stage, patches are classified into the aforementioned surface types by curvature analysis. Final segmentation is achieved by merging adjacent patches that belong to the same class, if their boundaries are non-crease edges.

Elaborating on the same idea of classifying surfaces based on differential geometry, Besl and Jain [10] proposed the HK map that allows a more precise analysis of surfaces. The HK map specifically analyzes the sign of the Gaussian curvature ($H$) and mean curvature ($K$) at each pixel, that can be $'+'$,$'-'$, or $'0'$, i.e. concave, convex,
or planar, resulting in 8 surface types. An initial stage of pixel clustering based on the HK surface classification is first applied. Rather than using region growing from identified surface class and neighbour edges, each local seed region is refined by applying an optimization process that iteratively fits a variable-order function to a growing 3D surface. An assumption of this approach therefore requires that a sufficiently large region be modelled as a piecewise-smooth surface.

For objects containing known simple quadric surfaces such as cones, cylinders, and spheres, the more direct approach of primitive fitting can be deployed. The parameters of these primitives can be estimated through error regression and the Hough Transform approach [64], and will be less sensitive to noise compared to differential estimation for surface classification. Similarly, the RESidual Consensus (RESC) [102] uses primitive fitting coupled with a RANSAC sampling strategy. This iterative algorithm randomly samples 3 points to estimate a plane and 9 points to estimate a quadratic surface at each iteration. To seek the parameters that best fit these two types of surfaces, the residual errors of each fit is measured. Subsequently, segmentation performance is maximized by searching for the largest continuous region where the residuals tend to be minimum, with a high tolerance of up to 80% outliers were reported.

An adaptive-scale parametric model estimation [99] that improves upon the RANSAC approach of RESC was recently proposed for primitive fitting. This algorithm is comprised of two stages; the Two-Step Scale estimator (TSSE) first applies nonparametric density estimation and density gradient estimation techniques, to robustly estimate the scale of the inliers. Then, the Adaptive Scale Sample Consensus (ASSC) simultaneously estimates the parameters of a model and the scale of the inliers belonging to
that fitting model. This approach is more robust to randomized primitive fitting as the prior of inlier scale is not needed. To allow the segmentation of free-form objects, Uckermann [95, 96] recently employed the edge-based and region growing approach. The edges are specifically derived from measuring angular differences between surface normals of neighbouring pixels. However, a disadvantage of such approach is that it neglects the scale dependency in estimating the normal to each pixel.

2.2.2 Segmenting 3D Surfaces from 3D Meshes

Synthetically created 3D meshes composed of interconnected triangle faces is the primary representation of 3D data in computer graphics applications [19], such as data compression for real-time rendering [46], texture mapping [58], and object modelling [28]. Each triangle face consists of 3 neighbouring vertices joined by 3 edges, and a 3D surface is simply a network of interconnected faces, where no more than 2 faces share a common edge. Most algorithms for segmenting 3D meshes therefore takes advantage of the connectivity offered by the vertices (i.e. 3D points) and the faces.

A popular solution to segmenting 3D meshes is by clustering a connected set of faces into \( K \) clusters, where \( K \) is pre-determined. In Shlafman's work [83], the K-means clustering algorithm that functions by minimizing the mean error as each observation is added to each cluster, was applied on 3D meshes. The initialization phase first chooses \( K \) seed faces to represent the cluster centers. These seed faces were selected such that their pairwise distances are maximized in a greedy fashion. Then, an iterative process that simultaneously assign faces to a segment according to traversal costs and recomputing the seed faces that minimize the sum of distances to faces in each cluster. This process repeats until the mean errors of all clusters
converge.

Hierarchical face clustering (HFC) [61, 29, 52] is another popular clustering approach, in which the data is segmented such that the level of detail can be scaled. This means that at a lower detail level, segmented surfaces can be represented by fewer 3D planes, gradually sacrificing the details of the surface geometries in exchange of a much simpler representation of the surface. HFC relies on first transforming a 3D mesh into a dual graph, representing each face as a single node, with two nodes joined by an edge if the two corresponding faces are adjacent. By assigning an error metric to these dual graph and iteratively contracting these edges, faces are clustered until K faces are left. To further simplify the representation, the HFC is also demonstrated to couple with a primitive fitting functions where plane, sphere, and cylinder [6] can be used to approximate the surface properties of each HFC cluster. This idea is similar to the continuous surface fitting and region growing approach discussed in the previous section.

A randomized approach based on an existing sets of segmentation results, was proposed via the randomized-cut [32] approach. The method was shown on randomizing the parameters of various mesh segmentation algorithms, including both HFC and K-mean clustering Segmentation, to produce a function that captures the probability that an edge lies on a segmentation boundary (a cut). This produces a ranked set of the most consistent cuts based on how much cuts overlap with others in a randomized set. In other words, this hierarchical decomposition procedure uses a set of randomized minimum cuts to guide placement of segmentation boundaries. They first decimate the mesh, for example, to 2,000 triangles in their work, and then
proceed top-down hierarchically, starting with all faces in a single segment and iteratively making binary splits. For each split, a set of randomized cuts for each segment is computed.

Recently, the contour-following strategy was proposed for 3D meshes, by reconstructing a contour on the 3D surfaces that satisfy a salience criteria [100]. Specifically, instead of transforming the mesh into a dual graph as in clustering, this algorithm treats the 3D mesh as a graph itself and extract a maximum spanning tree, that is a graph containing all the mesh vertices but only some of the edges. Subsequently, each remaining edge (i.e. excluding the edges in the spanning tree) is assigned a weight defined by three saliency terms: ridge, valley, and curve based on PCA. These remaining edges are added to the spanning tree if their weight meets the salience criterion. By defining these surface ridge structures, closed surfaces can be separated into regions to achieve mesh segmentation.

### 2.2.3 Segmenting 3D Surfaces from 3D Point Cloud

Similar segmentation methods for range images discussed, such as surface fitting or region growing for simple 3D surface, are also commonly used in segmenting 3D point cloud data. However, the lack of structured connectivity information from the image grid means that efficient neighbourhood estimation is necessary for efficient data processing. It is also possible to reconstruct 3D meshes from 3D point clouds and apply graph-based segmentation methods, as used in many reverse-engineering applications. Unfortunately, this may not be ideal for object recognition given the polygon reconstructed from the scene and object itself can vary due to the difference in depth resolution and sparseness.
Point clouds collected by current 3D sensors contain dense real-time 3D data, and the work by Schnabel [78] aimed to derive an efficient algorithm to handle the problem of large quantity of range data. Specifically, a segmentation driven by high performance RANSAC fitting of data into a combination of pure shape primitives was proposed. RANSAC was chosen because of its robustness in dealing with data containing a high number of outliers (e.g. more than 50%), as existed in many acquired point cloud data. The key to the proposed optimized RANSAC is by locally sampling the data, exploiting the fact that shapes are local phenomena. This means that the a priori probability of two points belonging to the same shape is higher if the distance between the points is smaller. An octree representation of the data is also used to efficiently process each iteration of RANSAC fitting, such that points are only sampled from a single cell of the octree. To increase the likelihood of finding the correct scale for a cell, the probability of the octree level that achieves the highest score is constantly updated and used to estimate the next shape candidate. A variety of primitive shapes including planes, spheres, cylinders, cones and tori were adapted into this framework. Therefore, at each iteration of the RANSACs fitting, the best fitted primitive type with maximal score is obtained by randomly sampling the minimal subset of points for each primitive type.

For scenes containing simple object-to-table relationship, Rusu [73] proposed to transform all 3D points into the world coordinate frame where the z-axis is always pointing upward. Clustering based on Euclidean distance and planar best-fit can therefore be applied efficiently to filter out possible table points. All remain point clusters situated on top of the extracted table are then fed into the object segmentation module, where 3D geometric primitives fitting is applied.
2.2.4 Discussion

The possibilities of how to segment 3D surfaces are multiple and we can draw inspirations from a number of key points from studying these segmentation methodologies, to synthesize a repeatable segmentation technique for our object recognition framework:

1. It is important to handle the usually noisy data presented in raw 3D data acquired from 3D sensors. Method that are robust to outliers such as RANSAC are powerful in handling such data.

2. Surface fitting is extremely common in many segmentation approaches, because they can be used to express a surface in a compact format. However, this limits the application to many non-free form surface objects. The proposed method in this thesis must overcome this obstacle.

3. Saliency measures such as PCA are important techniques when dealing with 3D data analysis, and region growing is a simple but effective technique in dealing with surfaces of various geometry. Both of these techniques will play a crucial part in the segmentation algorithm proposed in this thesis.
Chapter 3

The Detection of 3D Interest Points

Two important traits that identify a point as potentially interesting are its distinctiveness and repeatability [77]. Most importantly, both traits directly affect the matching ability of the LSD constructed at that point, that in turn determine the performance of object recognition. Distinctiveness is a measure of how a point from any region can be uniquely distinguished from a point from a different region. Measuring the distinctiveness of an interest point is challenging since it is also tightly coupled with its associated local neighbourhood. Distinctiveness therefore cannot be measured without also considering the type of the LSD used. A second trait is the repeatability, which is the measure of how reliably the same interest point is detected despite noise corruption and various deformations of the data.

In this thesis, we draw inspiration from this second trait by exploring possibilities of extending the interest points’ repeatability into a repeatable surface segmentation method, to be explained in chapter 4. Thus, we will first discuss variants of interest
point detectors and provide some insight into the detector performance. We will then discuss the differences between interest points extracted for improving LSDs matching, as compared to our application in facilitating surface segment extraction. Lastly, the repeatability of interest points will be demonstrated using both the original and noise-corrupted 3D models of various objects.

3.1 Interest Point Detector Variants

Most interest point detector select points by measuring the saliency within a local point neighbourhood, that is points locating at geometrically interesting regions. A detailed evaluation on various 3D interest point detectors was recently provided by Tombarai [94], in which detectors were categorized into the fixed-scale [18, 63, 91] and multi-scale detectors [63, 74, 97]. Fixed-scale detectors measure the saliency of a point based on a fixed radius, while the multi-scale approach is an optimization algorithm that measures the saliency at variable supporting radii, and selects the specific scale for a point that yield the maximum saliency. Although the multi-scale approach can yield more robust performance as the saliency differs due to the variation in size between and within an object, they are more computationally expensive. An effective compromise to these two categories that was not discussed in the report [94] is the scale-differences detector [40], where the measurement is based on the variation in saliency as the supporting radii changes.
CHAPTER 3. THE DETECTION OF 3D INTEREST POINTS

3.1.1 Fixed-scale Detector

The supporting radius in a fixed-scale detector has to be pre-determined and is tuned based on two factors: the scale of the object, and the sensitivity to occlusion. A larger object requires an equally large supporting radius to measure the saliency within a local neighbourhood. On the other hand, in cluttered scenes, if the radius is too big, the supporting radius may include points contributed by other objects that will affect the saliency measurement.

A common saliency measure of a 3D surface applies PCA to a local neighbourhood, decomposing the data into its principal, secondary, and minimal axes (a review of PCA is provided in Appendix A). The aforementioned VD-LSDs [91] and also in the work by Mian [63], both utilize a detector based on measuring the dispersion (i.e. Eigenvalue) ratio of the principal axes of a PCA frame (also used for extracting 9 basic properties for building the VD-LSD).

Eigen-analysis of the PCA frame yield the Eigenvalues \( \lambda_1 > \lambda_2 > \lambda_3 \), which are the magnitude associated with each of the Eigenvectors \( \vec{i}, \vec{j}, \vec{k} \). The PCA frame
can be viewed as a Cartesian coordinate, and for a geometrically flat surface, the magnitude $\lambda_3$ of the surface normal $\vec{k}$ will be significantly lower compared to the magnitude of the first principal axes, that is $\lambda_1 >> \lambda_3$. Therefore, an uninteresting flat region can be filtered when defining the threshold $Th$ is exceeded:

$$\left| \frac{\lambda_3}{\lambda_1} \right| > Th$$  \hspace{1cm} (3.1)

The absolute value of the ratio is needed because the PCA frame can be orientation ambiguously. Figure 3.1 demonstrates applying this saliency measure on the angel object. If a high threshold is used (e.g. $Th = 0.2$), only the smaller details like the crests on each side of the wings on the object are identified as interest points. As the threshold is lowered to $Th = 0.05$, even more subtle changes such as the contour of the eyes and the hairline are also included. A more selective criterion was defined in the LSD called Intrinsic Shape Signature (ISS) [103], replacing the single threshold by applying two thresholds $Th_{12}$ and $Th_{13}$ successively to the multiple axes of the PCA frame as:

$$\frac{\lambda_2}{\lambda_1} > Th_{12} \hspace{1cm} AND \hspace{1cm} \frac{\lambda_3}{\lambda_2} > Th_{23}$$  \hspace{1cm} (3.2)

This is because the 3D reference frames of the Eigen basis can become ambiguous when two of the Eigenvalues from the scatter matrix are equal. By imposing the additional second constraint on the ratios of the Eigenvalues, this can exclude frames of ambiguous axes at points of local symmetries. However, as data qualities vary, an interest point detector that is based on thresholding at a fixed-radius scale may not be sufficient to generate the repeatability required.
3.1.2 Multi-scale Detector

Detectors based on fixed-scale neighbourhoods overlook interest points that may belong to a different neighbourhood size. The goal of a multi-scale detector is to adaptively search for the optimal scale for each individual point. Using the same PCA-based detector as discussed, Mian [63] proposed to detect interest points using the same saliency measure, but by varying the size of the radii. Consider for a particular point, a function plotted such that its x-axis is the scale and the y-axis is the Eigen-value ratio $\frac{\lambda_3}{\lambda_1}$. Then the best scale of a specific interest point can then be selected by finding the local maxima of this function.

A multi-scale detector inspired by the Laplacian detector was proposed by Unnikrishnan and Herbert [97]. The Laplacian of an image highlights regions of rapid intensity change. Similarly, the 3D Laplacian detector is a second order differential detector that highlights rapid geometric curvature change. More specifically, this 3D Laplacian detector $A(p, t)$ calculates the displacement of a point along its normal by a quantity proportional to the mean curvature, where $p$ is the point of interest and $t$ is a scale instance. Similar to PCA-based techniques, the dependency on the local curvature means that $A(p, t)$ is invariant to rotation and translation. Formally, $A(p, t)$ is defined as:

$$A(p, t) \approx p + \frac{t^2}{2} L_M p.$$  \(3.3\)

where $L_M$ is the Laplace-Beltrami detector, a natural analogue of the Laplacian operator from Euclidean space, but operates in an intrinsic coordinate system defined on a manifold, measuring the divergence of gradient.
The automatic scale selection function $F$ is then formed by considering the exponential damping of $A(p, t)$, as:

$$F(p, t) = \frac{2 \| p - A(p, t) \|}{t} e^{-\frac{2 \| p - A(p, t) \|}{t}} \quad (3.4)$$

Due to this highly selective filtering process, the Laplacian-based multi-scale detector produces fewer interest points compared to other detectors discussed.

### 3.1.3 Difference-scale Detector

An effective compromise to the more computational expensive multi-scale detector is a difference-scale detector. To the author’s knowledge, the Difference of Normal (DoN) operator [40] is the only 3D detector proposed in the literature that follows this principal. The DoN operator functions similarly to the popular Difference of Gaussian (DoG) operator for 2D images. In DoG, the operator measures the Gaussian response corresponding to two different radii, whereas the DoN operator measures the angular difference between two normals $\{\vec{n}_1, \vec{n}_2\}$ of a point based on two different neighbourhood radii $\{r_1, r_2\}$.

To estimate a surface normal, a straightforward approach is to perform a plane fitting in the least squares sense within a neighbourhood. This would minimize the sum of Euclidean distances of all points within a small neighbourhood to the fitted plane. Note that more accurate quadric surface fitting is not preferred as the normal estimated has a smaller response to changes in the supporting radii. Surface normals from quadric surface fitting are more accurate because it consider a wide range of 3D surfaces such as ellipsoid, sphere, or a paraboloid. Both plane and quadric fitting are prone to errors caused by outliers, therefore, one of the most common practice in the 3D computer vision community is to employ PCA on a small neighbourhood around
the point where surface normal is to be estimated. A surface normal is simply the
Eigenvector that exhibits the smallest variance, associated the smallest Eigenvalue.
Another advantage of using the Eigenvector as the surface normal is that it does not
assume any underlying parametric surface as compared to plane and quadric surface
fitting.

Once the surface normals are obtained for different scale of radii, if the solid angle
$\Theta$ between $\vec{n}_1$ and $\vec{n}_2$ exceeds a predetermined threshold, then the 3D point is declared
as an interest point. By doing so, DoN realizes points where the surface geometry
changes significantly as the supporting neighbourhood radius increases from $r_1$ to $r_2$.
At any point, only two normals of different radii have to be computed, and so the
DoN interest points are simple but fast to obtain assuming that the 3D data has been
organized using a data structure for efficient neighbourhood retrieval such as the k-d
tree.

### 3.1.4 Selecting the Appropriate Detector

A key difference in our use of interest points compared to their more common use
of improving the effectiveness of LSD matching, is the omission of the non-maxima
suppression (NMS) of interest points. NMS of interest points for LSD matching is
important because LSDs constructed within close vicinity of each other are indistinct
and contribute to redundant matching, since they share similar geometric information.
There are various ways of achieving NMS of interest points. For example in the
PCA-based detector, additional saliency constraints are applied such as thresholding
the magnitude of the eigenvalue of a principal axis, or computing additional quality
measures using principal curvatures [63].
CHAPTER 3. THE DETECTION OF 3D INTEREST POINTS

In contrast to this requirement, the estimation of surface interest segments requires dense sampling of interest points, to later facilitate accurate segment boundary reconstruction (Section 4.1). A choice of interest point detector must therefore provide a dense set of repeatable interest points, and is relieved from the requirement and computational expense of suppressing non-maxima points. Multi-scale detectors as mentioned carry out NMS of interest points across various scales of the neighbourhood, and therefore do not suit our application in extracting surface segments.

Although fixed-scale detectors in comparison will be a better fit, their sensitivity to noise corruption if only a small neighbourhood is used may undermine the performance of the proposed recognition framework. The neighbourhood can be increased to reduce noise sensitivity, but the resulting interest points can be too abundant to be meaningful. For these reasons, a more appropriate choice will be applying the DoN difference-scale detector using two sufficiently distinct radii to adapt to data of various noise corruptions. In the next section, the repeatability of interest points of DoN and PCA fixed-scale detector will be evaluated experimentally.

3.2 Evaluation: Repeatability of Interest Points

The experiment aims to analyze the repeatability of dense interest points in realistic 3D data and addresses two major issues in the problem of object recognition: a) The sampling density of the scene - how densely or sparsely the surface is sampled compared to the 3D model; and b) The depth quantization of the scanner - how accurately the depth is measured at each point.
3.2.1 Data Set and Parameter Selection

To empirically measure the performance of each interest point detector against the two outlined issues in a controllable way, synthetic data (labelled as $M_{syn}$) are generated from the 3D point cloud model of various objects (labelled as $M_{org}$) as follows:

a) Depth quantization: To simulate different depth quantizations of different sensors, the depth value is rounded to the next most accurate value according to a specified depth accuracy. Sample generated data are shown in Figure 3.2.

b) Sampling density: To simulate different sampling densities of 3D surface, an approximation method is employed here. The object is first voxelized so that
each voxel is a 1 cm\(^3\) cube. Since it is assumed that only the surfaces are present in a 3D model, each voxel therefore only contains points on the object surface. Given that the voxel is small enough (i.e. 1 cm\(^3\)), \(n\) random points are sampled within each voxel if the targeted sampling density is \(n\) points per 1 cm\(^2\). Sample generated data are shown in Figure 3.3.

For the fixed-scale PCA detector, the less restrictive thresholding outlined in equation 3.1 is used to extract a denser sets of interest points. A high ratio of \(Th = 0.05\) is used to detect as many salient points as possible, and classify them as interest points. The neighbourhood radius for estimating each PCA frame will be evaluated using both 5 mm and 10 mm in order to understand the trade-off between small and large neighbourhood size for PCA calculation.

On the other hand, three parameters have to be defined for the DoN operator, comprising \(r_1\), \(r_2\), and \(\Theta\). In the original work [40], the DoN operator was applied on large scale 3D urban data, and different scales of \(\{r_1, r_2\}\) were required to estimate edges on structure of various sizes. For example, a lower range of \(\{r_1 = 0.1 m, r_2 = 1.0 m\}\) was used for building window edges and a higher range of \(\{r_1 = 0.8 m, r_2 = 8.0 m\}\) for larger structures such as building facades. In this work, it is assumed that objects to be detected are of similar size, and a single scale range therefore is sufficient to detect geometrically interesting points across multiple objects. Although in theory the angular threshold could be fine-tuned for different objects, in this experiment, a single scale range of \(\{r_1 = 5 \text{ mm}, r_2 = 15 \text{ mm}\}\) with a solid angular threshold \(\Theta = 15^\circ\) is used. The ratio of three between \(r_1\) and \(r_2\) accommodates a wide range of sampling densities and depth quantizations. In our implementation, the normals of a DoN operator are estimated using a least square plane fit to a local neighbourhood.
3.2.2 Measuring Repeatability

It is unfair to evaluate points detected in point cloud that may not even exist in the other. For comparing two different point cloud data, the intersection between $M_{org}$ and $M_{syn}$ must first be computed, to ensure that only points visible to both data participate in the repeatability calculation. Due to noise corruption of data, a more relaxed correspondence search radius is set at 2 mm, allowing interest point to be detected at an offset of ±2 mm. This means that a one-to-one correspondence between $M_{org}$ and $M_{syn}$ is not strictly enforced, the reason being that we are not measuring the precise location of interest points extracted, but rather the repeatability of dense interest points that can possibly be extracted within a close proximity of each other.

From all visible points that constitutes $M_{org} \cap M_{syn}$, our repeatability measure is based on the likelihood of detecting an interest point in $M_{org}$ that is also in $M_{syn}$. Likewise, it is also based on the likelihood of an interest point being detected in $M_{syn}$ that is in $M_{org}$. Let’s define the set of interest point found in $M_{syn}$ as $P_{M_{syn}}$ and that found in $M_{org}$ as $P_{M_{org}}$. Then repeatability measure is then simply defined as:

$$\text{repeatability score} = \frac{P_{M_{syn}} \cap P_{M_{org}}}{P_{M_{syn}} \cup P_{M_{org}}}$$ (3.5)

When two measured data are identical, the repeatability score is simply equal to the maximum of 1.

3.2.3 Results and Discussion

In this experiment, synthetic data are generated from a subset of 3D models used in experiments in later chapters, including the angel, elmo, bird house, and pipe objects (Figure 6.2).
Figure 3.4: Repeatability vs. Sampling Density plots for DoN interest points detected on various objects.
Figure 3.5: Repeatability vs. Depth Resolution plots for DoN interest points detected on various objects.
CHAPTER 3. THE DETECTION OF 3D INTEREST POINTS

Figure 3.6: DoN interest points detected on various objects. *Top Row:* Interest points detected on objects of original depth resolution. *Bottom Row:* Interest points detected on objects of depth resolution at 4 mm.

**Interest Point Repeatability vs. Sampling Density**

In the first experiment where sampling density is tested, we focused only on $r = 5 \text{ mm}$ for PCA fixed-scale detector and the DoN operator (Figure 3.4). Given that the lowest sampling rate tested is $\sim 10$ points per $1 \text{ cm}^2$, which represents a fairly low sampling rate on 3D surfaces compared to many real 3D sensing techniques, it is fair to conclude that both PCA estimation at $r = 5 \text{ mm}$ and the least square plane estimation at $r_1 = 5 \text{ mm}$, $r_2 = 10 \text{ mm}$ for DoN operator is sufficiently robust in extracting repeatable interest points. It can also be observed the angel and bird house objects maintain a higher repeatability as compared to the elmo and pipe objects when the sampling density drops. This is because the latter pair of objects have more salient points with less abrupt changes (e.g. the slow curving of the pipe circumference and the low geometric texture of the elmo object), and thus the lowered sampling density introduces slightly more error to the estimation of the normal and
principal component analysis.

**Interest Point Repeatability vs. Depth Quantization**

In the second experiment, different PCA neighbourhood radii \( r \) were used. Figure 3.5 demonstrates the respective repeatability using \( r = 5 \text{ mm} \) and \( r = 10 \text{ mm} \), in comparison with using a DoN operator. Compared to the change in performance versus sampling density, the change of depth quantization has a much higher impact on the measured repeatability. As the depth quantization is reduced, repeatability of a PCA fixed-scale detector drops significantly when \( r = 5 \text{mm} \). This is because the reduced depth quantization alters the principal components within a small neighbourhood (Figure 3.7, first and second rows). To counter this problem, a large neighbourhood at \( r = 10 \text{ mm} \) shows a major improvement in repeatability over the smaller neighbourhood (Figure 3.7 bottom row). However, as the local neighbourhood becomes larger, more points can be classified as interest points regardless of the depth quantization and the detection of interest points becomes noisy. On the other hand, qualitatively, the DoN operator offers more robust result towards the reduction in depth quantization, without introducing the selection of of non-salient points (Figure 3.6).

**Summary and Conclusion**

Different interest point detectors were studied in this chapter by studying the performance of dense interest point extractions on synthetic data. Through a repeatability evaluation, it was found that the scale-difference detector offers an attractive performance even against data with high noise corruption including changes in depth quantization and surface sampling rate. Empirically, its repeatability rate is about
Figure 3.7: Fixed-scale PCA interest points detected on various objects. *Top Row:* Interest points detected on objects of original depth resolution, $r = 5 \text{ mm}$. *Middle Row:* Interest points detected on objects of depth resolution at $4 \text{ mm}$, $r = 5 \text{ mm}$. *Bottom Row:* Interest points detected on objects of depth resolution at $4\text{mm}$, $r = 10\text{mm}$.

90% for data with sampling density at $\sim 20$ points per $1 \text{ cm}^2$ and depth quantization of $2 \text{ mm}$.

To further support the above observation, Figure 3.8 illustrates the extracted interest points from a fixed-scale PCA operator and from a DoN operator on scene data gathered from a LiDAR scanner up to an accuracy 220 microns [50] and a 1$^{st}$ generation Kinect camera. The objects in these scenes are placed at a distance of $\sim 1$ meter away from the depth sensors, where the depth resolution of data collected from
a Kinect is at $\sim 2\text{mm}$ [47]. It can be observed that if the same $Th$ for fixed-scale PCA operator is used, then many of the points in the Kinect data will be identified as interest points due to a much lower depth quantization. In contrast, the DoN detector can detect much more reliable interest points in both the accurate LiDAR data and the noisy Kinect data.
Figure 3.8: Detecting Dense set of Interest Points in 2.5D scenes from different sensor modalities. Left column: LiDAR data. Right column: Kinect data. First Row: 2.5D point cloud scene data. Second Row: Interest Points extracted using fixed-scale PCA operator. Third Row: Interest points extracted using the DoN detector.
Chapter 4

Interest Segment Extraction

A novel technique for segmenting 3D free-form surfaces is presented in this chapter. Unlike solutions that require noiseless CAD models or polygonal meshes, the proposed algorithm is aimed at point cloud data acquired from various grades of depth sensors, providing reliable segmentation results for 3D data of different depth quantization and surface sampling density. Note that there are publicly available segmentation algorithms for 3D point cloud data provided in the popular open source Point Cloud Library [1]. Although only algorithms with limited capabilities were available at the time the segmentation algorithm in this thesis was developed, including a Euclidean-clustering approach that segments objects based only on the closeness of any two points, and a simple planar segmentation of table-top from objects. These algorithms cannot achieve our goal of segmenting objects comprising 3D free-form surfaces into different surface segments.

A challenging aspect of segmenting free-form objects is that their surface normals vary continuously and their geometric boundaries are not represented by rigid edges, and so technique such as filtering based on clustering normals will simply fail.
Our proposed solution is to inherit the repeatability of interest points extracted via saliency measures of surface geometry, discussed in chapter 3. Following the detection of dense interest points, 3D interest segment boundaries are first reconstructed by locally optimizing the smoothness constraints imposed by the orientation of a collection of interest points. These boundaries provide an effective mean to later facilitate efficient 3D region growing to cluster points within the boundaries and form the interest segments. Figure 4.1 illustrates the results obtained at each intermediate step for a 3D model of the gnome object and a 2.5D scene containing the object. This example visually demonstrates the repeatability between the extracted interest segments across the data, including the coned head, the body, and the frog segments.

The rest of this chapter is as follows: First we discuss the algorithmic detail of the proposed segmentation technique. This is followed by an evaluation of segment repeatability using synthetic data of models with different depth quantization and sampling density. Then, the interest segment repeatability of across dense 3D models and sparse 2.5D scenes included occlusion and clutters is considered.

### 4.1 Reconstruction of Interest Segment Boundaries

In this section, the reconstruction of 3D closed-form boundaries enclosing each interest segment is explained. The key in reconstructing the segment boundaries is recognizing a smoothness constraint imposed by a small collection of interest points locally, and connecting them into individual piece-wise linear curves, defined as interest curves.

Recall that an interest point detector is in fact a salient feature point detector, therefore, the interest points are extracted along geometric edges, to be represented as interest curves here. Let’s consider each individual interest curve to be continuous
CHAPTER 4. INTEREST SEGMENT EXTRACTION

Figure 4.1: Intermediate steps of extracting interest segments from the 3D model of the gnome object and from a 2.5D scene containing the same object.
and smooth, such that no abrupt change in angle can be observed along the interest curve. An abrupt change in angle is simply defined as the intersection between two or more interest curves. Growing an interest curve can be roughly divided into four steps:

1. Estimate an orientation at the current interest point \( p_i \);

2. Find the next potential interest point \( p_j \) to connect;

3. Enforce a angular constraint between the section \( \vec{p}_i \vec{p}_j \) with other sections of the current interest curve;

4. Removing redundant interest points along an interest curve.

Following interest curve extraction of the point cloud, the interest curves are then connected to form 3D closed-boundary enclosing; each individual interest segment.

### 4.1.1 Estimate an orientation at the current interest point

To grow the interest curve so that it obeys the smoothness constraint, interest points are grouped such that they locally optimize the smoothness of each curve section in a greedy fashion. This requires first estimating the local interest curve orientation at each interest point, which can be efficiently achieved by applying PCA within a local neighbourhood of the extracted interest points. The orientation at the interest point is simply the eigenvector with the largest eigenvalue. Formally, let \( \mathbf{P} \) be the set of all such interest points \( p_{test} \) within the local neighbourhood of \( p_i \), then its tangential orientation is \( d_i \). Consider the 3D model of the gnome object again in Figure 4.2, which illustrates the tangential orientation estimated at the interest points of partial sections of the object.
4.1.2 Find the next interest point to connect

Having the orientation $\vec{d}_i$ estimated at $p_i$, the next step is to estimate the best neighbour point $p_j$ to be connected to $p_i$ and form a linked section $\overrightarrow{p_ip_{test}}$. A 3D vector $\vec{l}_i$ originating from $p_i$ and pointing in the direction $\vec{d}_i$ with a length of the local neighbourhood radius is constructed. The best $p_j$ is selected by iteratively searching the next available $p_{test}$ that is within $P$, such that the perpendicular distance $\tau$ from $p_{test}$ to $\vec{l}_i$, that is $\tau = \text{proj}_{\vec{l}_i} \overrightarrow{p_ip_{test}}$ is within a pre-defined scalar threshold $\delta$ and is the smallest amongst all other qualified $p_j$s:

$$p_j = \arg\min_{p_{test} \in P} \tau := \text{proj}_{\vec{l}_i} \overrightarrow{p_ip_{test}}, \quad \tau < \delta$$  \hspace{1cm} (4.1)
CHAPTER 4. INTEREST SEGMENT EXTRACTION

(a) Figure 4.3: Connecting interest points via direction estimated by PCA. This is achieved by finding \( p_j \) (purple) within \( \mathbf{P} \) based on \( \vec{d}_i \) (red) at \( p_i \) (green). If \( p_j \) is the neighbour point, its projected distance \( \tau = \text{proj}_{\vec{p}_i \vec{p}_j} \) (orange) must be within \( \delta = 0.5 \text{ mm} \).

In addition, the PCA orientation \( \vec{d}_i \) is ambiguous in that it can point in one of the two arbitrary directions, thus both directions \( \vec{d}_i \) and \( -\vec{d}_i \) will have to be considered (see Figure 4.3).

4.1.3 Enforce a smoothness constraint

Since the definition of our interest curve does not include branching, each interest point \( p_i \) is only connected to a maximum of two other interest points. Consecutive sections of the interest curve are further constrained by a smoothness angle \( \vartheta \). At each iteration of growing interest curve at \( p_i \), a maximum of 4 potential consecutively connected interest points denoted as \( p_h, p_i, p_j, p_k \), and its corresponding piecewise linear sections \( \vec{p}_h \vec{p}_i, \vec{p}_i \vec{p}_j, \) and \( \vec{p}_j \vec{p}_k \) are considered (Figure 4.4). Precisely, four different scenarios can transpire:

Case 1: both \( p_i \) and \( p_j \) are not connected to any interest points. A smoothness constraint is not applied.
Figure 4.4: Enforce a local smoothness constraint: 3D boundaries are in general smooth and fall along the intersection between two surfaces (the intersection between the hat and eyebrow of the gnome is demonstrated here). They can be reconstructed by connecting interest point \( p_i \) to an interest point within its local neighbourhood \( P \), enforced by the angle between \( \overrightarrow{p_i p_j} \) and \( \overrightarrow{p_j p_k} \) and \( \overrightarrow{p_i p_j} \), which must be within \( \vartheta < 45^0 \).

**Case 2**: \( p_i \) is not connected \( p_h \), but \( p_j \) is connected to \( p_k \). The smoothness constraint is:

\[
\frac{\overrightarrow{p_i p_j} \cdot \overrightarrow{p_j p_k}}{||\overrightarrow{p_i p_j}|| ||\overrightarrow{p_j p_k}||} < \vartheta
\]

**Case 3**: \( p_j \) is not connected \( p_k \), but \( p_h \) is connected to \( p_i \). The smoothness constraint is:

\[
\frac{\overrightarrow{p_h p_i} \cdot \overrightarrow{p_i p_j}}{||\overrightarrow{p_h p_i}|| ||\overrightarrow{p_i p_j}||} < \vartheta
\]

**Case 4**: \( p_h \) is connected to \( p_i \), and \( p_j \) is connected to \( p_k \). The smoothness constraint
Figure 4.5: Remove redundant interest points (purple) along a newly constructed interest curve (blue arrows)

The interest curve is allowed to grow as long as the connecting interest point \( p_j \) does not violate the smoothness constraint \( \vartheta \) and \( \text{proj}_{\vec{l}} \vec{p}_i \vec{p}_j \) is below \( \delta \). An interest curve stops growing with either these constraints are broken, or when \( p_{\text{test}} \) within \( P \) that satisfies these constraints cannot be found.

### 4.1.4 Removal of Redundant Representation of Interest Curve

Recall that each interest curve is extracted from a dense set of interest points representing the same 3D geometric edge. This set of interest points has not been filtered using non-maxima suppression methods, and so it is possible that redundant interest curve can be constructed at close vicinity to each other, which would add ambiguity to later segmentation. To prevent this, redundant interest points are identified by traversing each newly constructed interest curve, and removing any interest points whose perpendicular distance to a newly constructed interest curve is within a distance threshold (see Figure 4.5). The reconstruction of all interest curves finishes once all the interest points have been visited, where they are either assigned to an
CHAPTER 4. INTEREST SEGMENT EXTRACTION

4.1.5 Interest Segment Boundaries from Connecting Interest Curves

Closed boundaries are created by joining multiple interest curves. Unlike the formation of interest curves, 3D boundaries do not have to obey any smoothness constraint, as the boundaries of any 3D surface can occur at arbitrarily abrupt angles. Let’s denote the beginning and the ending of a particular interest curve $C$ as $p_0$ and $p_e$. Then $p_0$ and $p_e$ are allowed to connect to any other section on any other interest curves. The junction point is calculated by considering the shortest distance to $p_0$ and $p_e$ within their point neighbourhood. Examples of the 3D boundaries reconstructed from a few 3D objects are illustrated in Figure 4.6. In this figure, the green lines are the interest curves and the red lines are the connected edges between interest curves.

Figure 4.6: Reconstructed interest segment boundaries (both green and red) for various 3D models. The interest curves are shown in colour green and the connections between interest curves are shown in red.

interest curve or removed as redundant interest points.
CHAPTER 4. INTEREST SEGMENT EXTRACTION

4.2 Interest Segments Extraction from Fast 3D Region Growing

Having determined the boundaries of all interest segments, it still remains to identify those points that reside uniquely within each segment. One approach to region growing is to start at a random seed point that is not a boundary point. The seed point first looks for its neighbours that are also not boundaries. The neighbours are assigned to the same region as the seed point. This continues with each newly acquired point in the region, and ends when no more points can be added. This process can be efficiently performed in organized data such as 2D images where boundaries are well defined in pixel coordinates.

In contrast, 3D point clouds are unorganized and therefore, render it more difficult to employ the above-described region growing strategy. Growing regions in 3D point clouds must ensure that two neighbouring points are not separated by a boundary. This requirement can be computationally expensive, and requires first estimating the closest boundary to each point, and second evaluating the angular difference between the vectors formed from the boundary to each point. A three-step strategy is therefore proposed here to avoid such computational burden by removing the need to check for neighbour boundaries. This involves:

1. Classify 3D points as either edge points or region points;

2. Perform region growing on all region points;

3. Assign edge points to the corresponding regions.
4.2.1 Edge and Non-Edge Classification

To allow the growth of a region at any point without checking its boundary condition, it is convenient to classify those points (denoted as $R$) that are far away from the boundaries and those (denoted as $B$) that are close to the boundaries. By isolating $R$ from $B$, points can freely grow without checking for the boundary condition at each step. This classification of points $B$ close to the boundaries can subsequently be achieved effectively by finding all 3D points where the perpendicular distance to their closest boundaries are within a distance threshold. All remaining points are classified as $R$. A graphical representation of this classification is demonstrated in Figure 4.7, where $B$ are shown as points in red and $R$ are shown as points in green.

4.2.2 Region Growing of Non-Edge Points

A method is described here for growing a region in an unorganized 3D point cloud. First, points within $R$ are randomly chosen as seed points to start the process of
region growing. A seed point is assigned a new region label if it does not belong to any previously labelled region. The neighbourhood radius of any 3D point must be less than \( b \), since that is the threshold separating \( B \) from \( R \). All of the adjacent neighbours within \( R \) that are not yet labelled are added to this same region. If an adjacent neighbour point has already been associated with another distinct region, then the two regions can be merged. The region growing ends once all points in \( R \) have been visited and their neighbours have all been considered.

### 4.2.3 Assigning Edge Points to Surface Segments

There are two ways to assign the points in \( B \) to a region. The more precise method is to consider the location of an edge point with respect to a boundary and a region. That is, an edge point that belongs to a region should reside in a position between a boundary and a region point. A second method is simply assigning the edge point to its closest region, ignoring its position relative to the region and boundary. In this work, the second method is preferred since it is more efficient, and the loss in precision for segmentation is minimal compared to any error contributed by over- and under-segmentation. The first row of Figure 4.8 illustrates the resulting region segmentation for the angel object.

### 4.2.4 Merging Segments

In order to use interest segments to facilitate registration, their size must be sufficiently large so that their geometry is more distinctive for matching. As can be seen in Figure 4.8, the wing section of the angel object consists of many sections representing the “feather”. These surface segments can be matched with any other
“feather” sections and offer little advantages for registering the object. Therefore, a merging mechanism is required to combine interest segments together, without greatly affecting reducing their repeatability.

The approach taken here is to assume a final desired number of segments, which can be either specified by the user or determined based on the approximate total surface area of the object. From the set of all $N_{\text{merge}}$ segments in the 3D data, at each iteration the smallest segment is merged with its best candidate neighbour, which is determined under two conditions:

1. If none of the neighbour segments share a sufficiently long border with the smallest segment, then the neighbour segment with the closest point to the smallest segment is the best candidate.

2. If there exists a neighbour segment that shares a sufficiently long border, and is also the longest border with the smallest segment, then this identified neighbour segment is merged to the current segment.

The second condition is crucial in maintaining a continuous area for each merged interest segment, which can be important for registration. After each merge operation, the segments are again re-ranked and the process repeats until only $N_{\text{merge}}$ segments are left. The second row of Figure 4.8 illustrates the final $N_{\text{merge}} = 8$ segments for the angel object.

4.3 Evaluation: Interest Segments Repeatability

The objective here is to evaluate the repeatability of the segmentation method, by examining whether interest segments extracted from a 3D model in the database can
be repeatably and correctly segmented in an unknown 2.5D input scene, including clutters and occlusion.

4.3.1 Measuring Interest Segment Repeatability $\Psi$

To define the repeatability measure $\Psi$, we have to first align the pair of data sets, by estimating the ground truth transformations between the models and the scenes. To ensure accurate ground truth, 10 correspondence points instead of 3 are first manually selected to determine an initial transformation, and then applying ICP on all of the data to refine the transformation. Let’s define our 3D model used for matching as $M$ and the testing 2.5D scene as $S$. After this one-to-one correspondence is established between any point on the model $p_M$ and any point in the scene $p_S$, and each scene interest segment is then assigned the most likely corresponding model segment according to the repeatability measure $\Psi$, computed as follows.

Let the set of established corresponding point pairs between a model and a scene be denoted as $P_{MS}$. Each point $p_{MS} \in P_{MS}$ is associated with both a ground truth 3D
model segment labelled $R_M$ and a 2.5D scene segment labelled $R_S$. For each segment $R_S$ represented by a scene point within the set $P_{MS}$, the most likely corresponding $R_M$ can be determined by finding the model segment that has the largest overlap with $R_S$. If $N_{R_S}$ is the number of points in $R_S$ and $N_{R_M}$ is the number of points in $R_M$ respectively, then $N_{R_S \cap R_M}$ will be the number of points that can be found in both $R_S$ and $R_M$. $\Psi$ can be formally described as:

$$\Psi = \frac{N_{R_S \cap R_M}}{\max(N_{R_M}, N_{R_S})}$$ (4.2)

4.3.2 Experiments

The experiment is separated into two major parts. In the first part, synthetic data generated using the methods described in Chapter 3.2 is used. Our objective is to measure $\Psi$ against varying depth quantizations and surface sampling density (also defined in Chapter 3.2). In the second part, real data of 2.5D scenes from a LiDAR sensor are collected and tested. A subset of 3D models from Figure 6.2 is used in this experiment, including the Angel, Gnome, BigBird, and WatermelonKid objects. A fixed set of parameters were used throughout the entire experiment. For simplicity, all objects were segmented using a DoN angle threshold of $15^0$, and $r_1 = 5\text{mm}$, $r_2 = 15\text{mm}$. The number of final segments output for each object was set at 8.

Synthetic Data: Results and Discussion

In this experiment, the baseline of models used consist of 100 pts/cm$^2$ for sampling density and at the original depth resolution of the model. Each of the 8 interest segment is ranked and labelled by its size, then their repeatability $\Psi$ is computed with data of sampling density at 80, 60, 40, 20 pts/cm$^2$ and depth quantization at 1,
2, 3, and 4 mm. The average $\Psi$ for all segments, the average of the top 3 highest $\Psi$, and the top segments are shown in Figure 4.9 and Figure 4.10.

From these plots, it can be observed that neither decreasing the surface sampling nor increasing the depth quantization have a significant impact on $\Psi$. Across all different sampling densities, in general the average $\Psi$ of all 8 segments for object Angel, BigBird, Gnome, the Watermelonkid object are maintained at $\sim 55\%$, $60\%$, $55\%$, $45\%$ respectively. When considering the top 3 repeatable segments, $\Psi$ are maintained at $\sim 75\%$, $80\%$, $70\%$, $65\%$. For only the highest repeatable segments, their repeatability reached $\sim 85\%$, $90\%$, $85\%$, and $75\%$ for each object respectively for all sampling densities.

Similar performance can also be observed for depth quantization. For the same order presented, the average $\Psi$ are: $\sim 50\%$, $60\%$, $55\%$, $45\%$; top 3 average $\Psi$: $\sim 70\%$, $80\%$, $70\%$, $60\%$; and finally the top $\Psi$ are: $\sim 80\%$, $90\%$, $85\%$, $70\%$. The similarity in performance means that our segmentation algorithm is robust against the variation in data resolution, similar to the performance observed in interest points repeatability discussed in Chapter 3.

Another observation is that $\Psi$ differs amongst different objects and amongst different segments of the same object. For instance on average, both the Angel and the BigBird objects exhibit a higher $\Psi$ value compared to the Gnome object, followed by the WatermelonKid object. To understand this, Figure 4.11, 4.12 4.13, and 4.14 visually demonstrate the interest segments extracted for each object at different sampling densities. From these figures, it is interesting to note that the segments with high $\Psi$ are the object parts surrounded by boundaries where geometry varies significantly, such as the wings (segment 2,3) and the left arm (segment 4) of the Angel object, the
Figure 4.9: Interest segment repeatability ($\Psi$) of different objects from different surface sampling vs. 100 pts/cm$^2$. 
Figure 4.10: Interest segment repeatability of different objects from different depth quantization vs. original data.
tail (segment 6) of BigBird object, and the frog (segment 6) of the Gnome object. Compared to other object parts where the geometric boundaries are less defined, in which more interest points are extracted, geometrically defined boundaries can be more accurately estimated and reconstructed.

**Real Data: Results and Discussion**

The second experiment compares extracted model segments with 9 different 2.5D scenes collected by a LiDAR sensor. Since there are roughly 5 objects in each scene, where occlusion from other other objects and self-occlusion can occur, the number of merged segments in each scene is set at 20. The segmented data can be seen in Figure 4.15. Instead of using the full resolution model, we opt to use the model with a sampling density of 20 pts/cm$^2$. The reason for this is based on the observation in the previous experiment, where $\Psi$ are maintained across different data resolutions (both depth and surface). Data of high resolution that is more computationally to process, therefore offers no clear advantage over lower resolution data. The use of smaller data becomes even more relevant when it is required to register multiple interest segments in the scene, as to be discussed in Chapter 5.

Shown in Figure 4.16 are the average $\Psi$, max of top 3 average $\Psi$ (since there can be less than 3 segments found in a scene per object), and the top $\Psi$ of each object in each scene. If the average of all segments and all objects are considered, then the overall segment repeatability is 64.1% with a standard deviation of ±17%. If only the top 3 segments of all objects are considered, the the overall segment repeatability is 67.8% with a standard deviation of ±16.8% Finally, if only the top repeatable segment for each object in each scene is considered, than the repeatability is 80.2%
with a standard deviation of ±12.5%. Interest segments extracted for both the models and the scenes are shown in Figs. 4.17, 4.18, 4.19, and 4.20, where their respective \( \Psi \) are provided in Table 4.5, 4.6, 4.7, 4.8.

**Summary**

In conclusion, the interest segments maintain the repeatability throughout various test data of varying data resolution including depth and sampling rate. We found that our segmentation technique is inevitably affected by the type of 3D surfaces on different 3D objects. However, as can be seen in the experiment, the average segment repeatability is still maintained at a respectable level of at least 60%, which usually provides sufficient overlap for most robust registration methods. In addition, for the purpose of recognition by registration where only 1 repeatable segment is enough to facilitate a successful recognition event, our top final segment repeatability reaches on average a high 80% accuracy for complicated free-form surfaces, providing a solid platform to build a robust recognition system based on 3D surface registration.
Figure 4.11: Interest segment extracted for the angel object from data of different surface sampling.

<table>
<thead>
<tr>
<th>Segment Repeatability (Ψ)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 Pts/cm²</td>
<td>0.51</td>
<td>0.87</td>
<td>0.64</td>
<td>0.88</td>
<td>0.68</td>
<td>0.54</td>
<td>0.69</td>
<td>0.26</td>
<td>0.63</td>
<td>0.81</td>
</tr>
<tr>
<td>60 Pts/cm²</td>
<td>0.4</td>
<td>0.86</td>
<td>0.51</td>
<td>0.45</td>
<td>0.71</td>
<td>0.20</td>
<td>0.66</td>
<td>0.49</td>
<td>0.54</td>
<td>0.74</td>
</tr>
<tr>
<td>40 Pts/cm²</td>
<td>0.28</td>
<td>0.56</td>
<td>0.83</td>
<td>0.64</td>
<td>0.40</td>
<td>0.34</td>
<td>0.66</td>
<td>0.24</td>
<td>0.49</td>
<td>0.71</td>
</tr>
<tr>
<td>20 Pts/cm²</td>
<td>0.29</td>
<td>0.78</td>
<td>0.66</td>
<td>0.86</td>
<td>0.35</td>
<td>0.49</td>
<td>0.72</td>
<td>0.37</td>
<td>0.57</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 4.1: Interest segment repeatability of the final 8 segments for object angel with different surface sampling. The highest Ψ for each data sampling is highlighted in green.


Figure 4.12: Interest segment extracted for the bigBird object from data of different surface sampling.

<table>
<thead>
<tr>
<th>Segment Repeatability</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 Pts/cm²</td>
<td>0.33</td>
<td>0.55</td>
<td>0.53</td>
<td>0.97</td>
<td>0.51</td>
<td>0.39</td>
<td>0.93</td>
<td>0.30</td>
<td>0.56</td>
<td>0.82</td>
</tr>
<tr>
<td>60 Pts/cm²</td>
<td>0.46</td>
<td>0.83</td>
<td>0.48</td>
<td>0.71</td>
<td>0.37</td>
<td>0.72</td>
<td>0.76</td>
<td>0.64</td>
<td>0.62</td>
<td>0.77</td>
</tr>
<tr>
<td>40 Pts/cm²</td>
<td>0.40</td>
<td>0.64</td>
<td>0.57</td>
<td>0.72</td>
<td>0.72</td>
<td>0.39</td>
<td>0.90</td>
<td>0.67</td>
<td>0.63</td>
<td>0.78</td>
</tr>
<tr>
<td>20 Pts/cm²</td>
<td>0.61</td>
<td>0.70</td>
<td>0.55</td>
<td>0.79</td>
<td>0.64</td>
<td>0.68</td>
<td>0.94</td>
<td>0.30</td>
<td>0.65</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 4.2: Interest segment repeatability of the final 8 segments for object bigbird with different surface sampling. The highest $\Psi$ for each data sampling is highlighted in green.
CHAPTER 4. INTEREST SEGMENT EXTRACTION

(a) 100 Pts/cm²  (b) 80 Pts/cm²  (c) 60 Pts/cm²  (d) 40 Pts/cm²  (e) 20 Pts/cm²

Figure 4.13: Interest segment extracted for the gnome object from data of different surface sampling.

<table>
<thead>
<tr>
<th>Segment Repeatability</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 pts/cm²</td>
<td>0.81</td>
<td>0.26</td>
<td>0.39</td>
<td>0.34</td>
<td>0.47</td>
<td>0.89</td>
<td>0.71</td>
<td>0.38</td>
<td>0.53</td>
<td>0.80</td>
</tr>
<tr>
<td>60 pts/cm²</td>
<td>0.52</td>
<td>0.34</td>
<td>0.45</td>
<td>0.63</td>
<td>0.49</td>
<td>0.76</td>
<td>0.43</td>
<td>0.27</td>
<td>0.49</td>
<td>0.64</td>
</tr>
<tr>
<td>40 pts/cm²</td>
<td>0.73</td>
<td>0.51</td>
<td>0.65</td>
<td>0.75</td>
<td>0.83</td>
<td>0.90</td>
<td>0.42</td>
<td>0.34</td>
<td>0.64</td>
<td>0.83</td>
</tr>
<tr>
<td>20 pts/cm²</td>
<td>0.45</td>
<td>0.51</td>
<td>0.19</td>
<td>0.54</td>
<td>0.44</td>
<td>0.86</td>
<td>0.51</td>
<td>0.47</td>
<td>0.50</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 4.3: Interest segment repeatability of the final 8 segments for object gnome with different surface sampling. The highest $\Psi$ for each data sampling is highlighted in green.
Figure 4.14: Interest segment extracted for the watermelonKid object from data of different surface sampling.

Table 4.4: Interest segment repeatability of the final 8 segments for watermelonKid with different surface sampling setting. The highest Ψ for each data sampling is highlighted in green.
Figure 4.15: Segmented 2.5D LiDAR scene data used in this experiment.
Figure 4.16: Interest segment repeatability of different objects in different 2.5D scenes.
CHAPTER 4. INTEREST SEGMENT EXTRACTION

(a) Interest segments from 3D model, labelled by size

(b) Interest segments from 2.5D LiDAR scenes

Figure 4.17: Interest segment extracted for the angel object from 3D models and 2.5D scenes. (a) Interest segments extracted for the angel object at sampling density of 20 pts/cm². (b) Corresponding interest segments extracted from 9 2.5D LiDAR scenes.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td></td>
<td>0.79</td>
<td>0.51</td>
<td></td>
<td>0.95</td>
<td></td>
<td></td>
<td>0.48</td>
<td>0.68</td>
<td>0.75</td>
</tr>
<tr>
<td>Scene 2</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.54</td>
<td>0.73</td>
<td>0.83</td>
<td>0.82</td>
<td>0.94</td>
<td></td>
</tr>
<tr>
<td>Scene 3</td>
<td>0.72</td>
<td>0.97</td>
<td></td>
<td></td>
<td>0.51</td>
<td></td>
<td></td>
<td>0.73</td>
<td>0.73</td>
<td></td>
</tr>
<tr>
<td>Scene 4</td>
<td>0.57</td>
<td>0.36</td>
<td>0.61</td>
<td></td>
<td>0.67</td>
<td>0.55</td>
<td>0.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 5</td>
<td></td>
<td></td>
<td>0.6</td>
<td></td>
<td></td>
<td>0.41</td>
<td>0.51</td>
<td>0.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 6</td>
<td>0.6</td>
<td>0.68</td>
<td></td>
<td>0.44</td>
<td></td>
<td>0.57</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 7</td>
<td>0.99</td>
<td>0.75</td>
<td>0.5</td>
<td></td>
<td>0.67</td>
<td>0.69</td>
<td>0.72</td>
<td>0.81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 8</td>
<td>0.8</td>
<td>0.85</td>
<td></td>
<td>0.46</td>
<td>0.42</td>
<td>0.63</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 9</td>
<td>0.41</td>
<td></td>
<td></td>
<td>0.75</td>
<td></td>
<td>0.30</td>
<td>0.49</td>
<td>0.49</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.5: 3D model Interest segment repeatability Ψ of the angel object in 9 LiDAR scenes. The highest Ψ for each data sampling is highlighted in green.
CHAPTER 4. INTEREST SEGMENT EXTRACTION

Figure 4.18: Interest segment extracted for the bigbird object from 3D models and 2.5D scenes. (a) Interest segments extracted for the bigbird object at sampling density of 20 pts/cm$^2$. (b) Corresponding interest segments extracted from 9 2.5D LiDAR scenes.

Table 4.6: 3D model Interest segment repeatability $\Psi$ of the bigBird object in 9 LiDAR scenes. The highest $\Psi$ for each data sampling is highlighted in green.
CHAPTER 4. INTEREST SEGMENT EXTRACTION

(a) Interest segments from 3D model, labelled by size

(b) Interest segments from 2.5D LiDAR scenes

Figure 4.19: Interest segment extracted for the gnome object from 3D models and 2.5D scenes. (a) Interest segments extracted for the gnome object at sampling density of 20 pts/cm\(^2\). (b) Corresponding interest segments extracted from 9 2.5D LiDAR scenes.

<table>
<thead>
<tr>
<th>Scene</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td>0.76</td>
<td>0.94</td>
<td></td>
<td>0.54</td>
<td>0.72</td>
<td>0.78</td>
<td>0.75</td>
<td>0.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 2</td>
<td>0.64</td>
<td>0.39</td>
<td>0.45</td>
<td>0.80</td>
<td></td>
<td></td>
<td></td>
<td>0.57</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>Scene 3</td>
<td>0.52</td>
<td>0.59</td>
<td>0.73</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
<td>0.62</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>Scene 4</td>
<td>0.44</td>
<td></td>
<td></td>
<td></td>
<td>0.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 5</td>
<td>0.77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.63</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>Scene 6</td>
<td>0.4</td>
<td></td>
<td></td>
<td>0.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene 7</td>
<td>0.48</td>
<td>0.7</td>
<td>0.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.60</td>
<td>0.60</td>
<td></td>
</tr>
<tr>
<td>Scene 8</td>
<td>0.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.46</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>Scene 9</td>
<td>0.52</td>
<td>0.74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7: 3D model Interest segment repeatability $\Psi$ of the gnome object in 9 LiDAR scenes. The highest $\Psi$ for each data sampling is highlighted in green.
Figure 4.20: Interest segment extracted for the watermelonKid object from 3D models and 2.5D scenes. (a) Interest segments extracted for the watermelonKid object at sampling density of 20 pts/cm$^2$. (b) Corresponding interest segments extracted from 9 2.5D LiDAR scenes.

<table>
<thead>
<tr>
<th>Scene</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Avg</th>
<th>Top 3 Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td>0.80</td>
<td>0.65</td>
<td></td>
<td>0.53</td>
<td></td>
<td></td>
<td>0.55</td>
<td></td>
<td>0.65</td>
<td>0.69</td>
</tr>
<tr>
<td>Scene 2</td>
<td>0.57</td>
<td>0.76</td>
<td></td>
<td></td>
<td>0.39</td>
<td></td>
<td></td>
<td>0.57</td>
<td></td>
<td>0.57</td>
</tr>
<tr>
<td>Scene 3</td>
<td>0.77</td>
<td>0.51</td>
<td></td>
<td>0.41</td>
<td></td>
<td></td>
<td>0.52</td>
<td></td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>Scene 4</td>
<td>0.65</td>
<td>0.69</td>
<td>0.54</td>
<td></td>
<td></td>
<td></td>
<td>0.58</td>
<td></td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>Scene 5</td>
<td>0.65</td>
<td>0.85</td>
<td></td>
<td>0.94</td>
<td>0.54</td>
<td></td>
<td></td>
<td>0.75</td>
<td></td>
<td>0.81</td>
</tr>
<tr>
<td>Scene 6</td>
<td>0.57</td>
<td>0.52</td>
<td>0.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.60</td>
<td></td>
<td>0.60</td>
</tr>
<tr>
<td>Scene 7</td>
<td>0.64</td>
<td>0.84</td>
<td></td>
<td>0.74</td>
<td>0.48</td>
<td></td>
<td></td>
<td>0.68</td>
<td></td>
<td>0.74</td>
</tr>
<tr>
<td>Scene 8</td>
<td>0.50</td>
<td>0.98</td>
<td></td>
<td>0.49</td>
<td></td>
<td></td>
<td>0.64</td>
<td></td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>Scene 9</td>
<td>0.64</td>
<td>0.63</td>
<td></td>
<td>0.58</td>
<td>0.75</td>
<td></td>
<td>0.36</td>
<td></td>
<td>0.59</td>
<td>0.67</td>
</tr>
</tbody>
</table>

Table 4.8: 3D model Interest segment repeatability $\Psi$ of the watermelonKid object in 9 LiDAR scenes. The highest $\Psi$ for each data sampling is highlighted in green.
Chapter 5

Registration of 3D Interest Segments

Pairwise registration of 3D surfaces is the process of aligning a reference surface $Q_{ref}$ that is fixed and the target surface $Q_{target}$ that is to be transformed into the coordinate frame of the reference surface. Each surface starts at an arbitrary orientation and translation, and the goal is estimate an accurate rigid 6 DOF transformation that best aligns the two surfaces. In the case of registering a pair of interest segments extracted from the 3D model and 2.5D scene, the registration technique also needs to address two issues:

1. Repeatability of the interest segment ($\Psi$) due to over-segmentation or under-segmentation; and,

2. Completeness of the interest segment ($\varphi$) due to self-occlusion and occlusion by other objects, multiplied by the repeatability.
In this thesis, we specifically explored a robust registration technique named the 4-Points Congruent Set (4PCS) \cite{2} that is loosely based on the general RANSAC algorithm \cite{24} for robust data fitting. Our choice is based on the strength of 4PCS in handling any arbitrary starting positions of $Q_{\text{ref}}$ and $Q_{\text{target}}$, and its robustness towards partially overlapping data by restricting the selection of wide bases in $Q_{\text{ref}}$ and $Q_{\text{target}}$ to approximately congruent bases. A further improvement was added to the original 4PCS algorithm to make it more suitable for the application of registering interest segments. First, distance quantization of the 4PCS base from the model segment that can be preprocessed, is utilized to optimize the matching efficiency of the registration process. Second, instead of returning a single rigid transformation, a list of transformations that are clustered by their poses and ranked by segment matching score are reported. This allows further verification in a later stage of an object recognition system with the reprojection of all the points in the 3D model. Experimentally, parameters affecting the 4PCS performance, together with the repeatability and the completeness of interest segments were studied using various 3D model and 2.5D scene data.

5.1 Featureless Registration of 3D Interest Segments

Data registration can be considered as a sub-problem of object recognition, where the goal of both problems is retrieving a rigid transformation between two different data sets. The difference being in the former case, there are only 2 sets of data $Q_{\text{ref}}$ and $Q_{\text{target}}$, which partially overlap. On the other hand, object recognition also has
to handle outliers due to the presence of other irrelevant data, e.g. other objects. Naturally, Local Shape Descriptors (LSDs) discussed previously for the purpose of object recognition, is also an effective solution to quickly establish one-to-one correspondences between $Q_{\text{ref}}$ and $Q_{\text{target}}$. For instance, the VD-LSD was successfully demonstrated for the purpose of aligning different 2.5D views of objects in 3D model generation [89]. However, the use of LSDs to register interest segments contradicts the idea of a RbR framework, whose goal is to eliminate local features that are prone to noise corruption and data deformation. Hence, our focus here is investigating a solution that is independent of extracting local surface features.

A widely used method for registration without relying on local feature is the Iterative Closest Point (ICP) by Besl and Kay [11]. The concept of ICP is simple yet very effective. Under the basic assumption is that $Q_{\text{ref}}$ and $Q_{\text{target}}$ are roughly aligned, ICP proceeds as follows:

1. Establish point correspondences by using a nearest neighbour criteria;

2. Estimate the 6 DOF transformation $T$ in a least square sense from the established correspondences [4], which requires at least 3 correspondences to make the system rank sufficient;

3. Transform points in $Q_{\text{target}}$ using $T$, and calculate the overall Euclidean distance error between all points in $Q_{\text{target}}$ and $T$;

4. Iterate Steps 1 to 3 until either the distance error is within a pre-defined threshold, or a maximum number of iterations is met, or some other stopping criterion is satisfied.

One option to apply ICP to registering interest segments is to first align two segments
using their centroids and estimated normals. But this solution can easily fail in complicated scenes with significant amount of occlusion, which drastically changes the shapes of two segments even though they may be perfectly segmented, hindering ICP performance in aligning a pair of interest segments. The potential partial overlapping and imperfection in data segmentation may skew the registration result, and drive the transformation into the wrong local minima. Nevertheless, ICP remains an effective solution to finely register $Q_{ref}$ and $Q_{target}$ once an approximate rigid transformation has been retrieved by other methods and point correspondences can be established using nearest neighbour criteria.

Using a statistical approach, one can also model the point sets $Q_{ref}$ and $Q_{target}$ as individual Gaussian mixture models (GMM) [42]. The idea is that each point is represented as a Gaussian component in the GMM, derived from their spatial location and neighbour points. The registration is carried out by iteratively minimizing the L2 distance between the two GMM, similar to an iteration of ICP. The statistical approach in general reduces the sensitivity to missing correspondences or excessive outliers by avoiding explicitly establishing point correspondences.

Although both ICP or statistical-based techniques offer a closed-form solution, a shared weakness is their requirement for an initial transformation between the two surfaces, which is unknown when registering a scene interest segment $R_S$ with a model interest segment $R_M$. To this end, the 4PCS algorithm that is based on robust data fitting through RANSAC is utilized. Instead of establishing point correspondences and iteratively reducing the point-to-point distance error, a RANSAC-based algorithm iteratively fits the data by randomly choosing a minimum set of points to establish a candidate model. In this case, the model is a base consisting of four 3D
CHAPTER 5. REGISTRATION OF 3D INTEREST SEGMENTS

points that define a rigid transformation between $R_M$ to $R_S$. The best candidate model that best fits the data will be selected after a fixed number of iterations. The following sections proceeded by first reviewing some key points of the 4PCS algorithm, followed by the explanation of improvements to 4PCS that enable a more effective matching of interest segments.

5.1.1 The 4 Points Congruent Set (4PCS) Algorithm

Using a general RANSAC approach [24] to align $Q_{\text{ref}}$ with $Q_{\text{target}}$, a naive strategy is to randomly select a base composed of 3 points from $Q_{\text{ref}}$, and all possible combinations of 3-points bases from $Q_{\text{target}}$ to estimate the best transformation $T$ between the surfaces. This process is repeated $L$ times, that is the number of RANSAC cycles, until the $T$ that best aligns the data is accepted by measuring the reprojection of all points in $Q_{\text{target}}$ that are within a defined threshold distance to all points in $Q_{\text{ref}}$. Given that there are $n$ points in $Q_{\text{target}}$, then at each RANSAC iteration of selecting a 3-points base from $Q_{\text{ref}}$, it will have to consider $O(n^3)$ bases from $Q$, making this strategy computationally inefficient.

The 4PCS introduced by Aiger [2] drastically improves the efficiency over naive RANSAC by introducing a constraint when selecting the bases from $Q_{\text{ref}}$ and $Q_{\text{target}}$. Instead of using a base of 3 random points, it considers a congruent base $\Pi=\{p_a, p_b, p_c, p_d\}$ consisting of 4 coplanar points to be extracted from $Q_{\text{ref}}$. To tolerate noise in real data, $\Pi$ is actually selected by first randomly sampling 3 points from $Q_{\text{ref}}$, and then the last point is selected such that it is within a defined error distance threshold such that $\{p_a, p_b, p_c, p_d\}$ are approximately coplanar. A wide base is also preferred when selecting $\Pi$ since it offers higher stability to estimate the rotation component of $T$. 
A wider base can also increase the discriminative power of selecting a corresponding coplanar base, since coplanar bases of shorter distance are more likely to be formed compared to bases composed of longer distance point pairs.

Contrary to the general RANSAC approach, the selection of all corresponding bases in $Q_{\text{target}}$ is reduced to selecting from among those bases $U$ that are approximately congruent to $\Pi$. That is, $U$ can be effectively extracted by only selecting those point pairs where their Euclidean distance match the two distance: $\Pi_{\text{pair1}} = ||p_a - p_b||$, and $\Pi_{\text{pair2}} = ||p_c - p_d||$. The approximate congruent base allows some flexibility in choosing the potential base, up to an error threshold of $\delta$. The number of bases to be considered at each 4PCS iteration is therefore reduced from $O(n^3)$ to the order of $O(n^2 + k)$, where $k$ is the number of potential congruent sets in $Q_{\text{target}}$. Two affine invariant ratios from $\Pi$ can then be used to locate the intersection $p_e$ between the two vectors $\overrightarrow{p_a p_b}$ and $\overrightarrow{p_c p_d}$:

$$\text{ratio}_1 = \frac{||p_a - p_e||}{||p_a - p_b||}$$
$$\text{ratio}_2 = \frac{||p_c - p_e||}{||p_c - p_d||}$$

Using the ratios $\text{ratio}_1$, $\text{ratio}_2$ and all the qualified point pairs selected from $Q_{\text{target}}$ and substituting them into equation 5.1, a list of potential intermediate points are found. $U$ can be localized efficiently by searching two intermediate points that representing the same 3D point within a neighbourhood radius of $\chi$, and that are from two different point pairs satisfying the $\Pi_{\text{pair1}}$ and $\Pi_{\text{pair2}}$ constraint respectively.

The bases extracted this way still belong to an affine invariant superset, and thus a truly congruent base is further enforced by the distance $||p_a - p_c||$, $||p_a - p_d||$, $||p_b - p_c||$, $||p_b - p_d||$ or the angle between $\overrightarrow{p_a p_b}$ and $\overrightarrow{p_c p_d}$. In the original work, a range tree [5] was used to locate the nearest neighbour for the intermediate points. In this work,
we opt to organize the intermediate points in a voxelized space for more efficient neighbourhood search.

5.1.2 Distance Quantization of 4PCS Bases

At each iteration of the 4PCS algorithm where a congruent base is already extracted from $Q_{\text{ref}}$, it is still required to go through all the points in $Q_{\text{target}}$ and extract all those point pairs that satisfy the $\Pi_{\text{pair}1}, \Pi_{\text{pair}2}$ constraints before they can be further verified as a congruent coplanar set. For registering a known 3D model segment with an unknown 2.5D scene interest segment, one can further take advantage of prior knowledge of the model segments and pre-compute the distance between any point pairs. Precisely, the distance $\text{Dist}_{xy} \|p_x - p_y\|$ between any point pair $(p_x, p_y)$ on any model segment $R_M$ are pre-computed and stored at training time.

To provide fast access to any point pairs at run-time, $\text{dist}_{xy}$ can be quantized such that each quantized distance can be used as a hash key to locate each point pair efficiently. All the retrieved point pairs associated with the hash key constitute the sets of all potential congruent base set that satisfy the $\Pi_{\text{pair}1}, \Pi_{\text{pair}2}$ constraints, thus
generating all corresponding congruent bases. This improvement reduces the search space for all congruent bases from $O(n^2 + k)$ to $O(k)$, where $k$ is the number of pairs that satisfies the $\Pi_{pair_1}, \Pi_{pair_2}$ constraint.

To prevent loss in precision when using the quantized distance for computing the $r_1$ and $r_2$ ratio of a congruent basis, the actual distance $dist_{xy}$ is stored as a third tuplet associate with each point pair, as illustrated in Figure 5.1. By doing so, we make sure that the pre-computation of 4PCS point pairs does not affect the precision of $dist_{xy}$. In summary, the quantized distance now used as a hash key, drastically improves the time performance of 4PCS by providing a fast access to all point pairs that match the distance criteria, without any degradation in matching accuracy. However, our strategy does come with the cost of increased memory and offline pre-processing which makes this approach more suitable for object recognition, where preprocessing a database of object models is typical, than for online registration, which was the target application of the initial 4PCS work.

To address the disadvantage of memory requirement in pre-computation to store all the point pairs, a trade-off is to avoid computing and storing point pairs that are too close to each other. Given that there are $n$ points in a surface segment, then the number of point pairs is equal to $\frac{n(n-1)}{2}$. As stated in the original work of 4PCS, it is advantageous to avoid $\Pi$ with small base pairs. A wider base will offer a much higher stability in the estimated transformation. Illustrated in Figure 5.2 is the distribution for the number of point pairs versus its occurrence in a plane consisting of 900 equally sampled 3D points. For a plane that is equal in width and length, the number of point pairs peaked at around the median of $dist_{xy}$. After the peak, the number of point pairs per distance decreases. The same trend can be observed
for more elongated surfaces, while the distribution of point pairs is stretched to a larger distance, the number of point pairs peaked at about the same distance, then the number of point pairs per distance again decreases. Thus, as stated in Section 5.2.1, our current implementation ignores any point pairs that are below 60 mm, to take advantage of this observation and the usage of a wider coplanar base.

Lastly, since the interest segments from the 2.5D scene are of different resolution and are potentially corrupted by outliers, it is also necessary to inspect neighbour bins to provide more robust results. All of the point pairs within the bin $||\Pi_{pair1} \pm \Delta||$ and $||\Pi_{pair2} \pm \Delta||$, where $\Delta$ is the distance to the current bin, are also added to the list of point pairs when finding $U$. Lastly, when selecting $\Pi$ from a scene interest segment, we can further restrict the search of point pairs of $\text{dist}_{xy}$ to be underneath
the maximum point pair distance in the targeting model interest segment, to reduce
the number $\Pi$ to be selected and tested.

5.1.3 Ranking and Clustering of Transformation Candidates

Normally, pairwise registration only returns the optimal rigid transformation $T$ that
best aligns $Q_{ref}$ to $Q_{target}$. This registration score $\kappa$ can be computed using the same
fashion as measuring the repeatability $\Psi$ between two interest segments as shown in
Equation 4.2. However, depending on the size and shape of the interest segments,
pairwise registration based only on interest segments can be less distinctive compared
to registration of the entire model, where more points are available to support the
hypothesis. A way to avoid error in returning a non-optimal $T$ is to instead return a
list of $T$'s, ranked by their respective $\kappa$ values. The list of $T$ can later be verified using
the entire 3D model during the recognition phase of a RbR framework, as explained
in Chapter 6.

In addition to returning a list of candidates ranked by $\kappa$, $T$'s that occupy similar
6DOF space can be further clustered based on their rotation and translation differ-
ences. This offers the advantage of increasing the probability of returning the correct
$T$ by evenly spacing the retrieved $T$'s across the 6 DOF pose space.

In summary, the pose clustering and ranking of poses are performed in the follow-
ing order:

1. Check if $\kappa$ of the newly retrieved pose $T_i$ is greater than the candidate $T_j$ with
   the lowest $\kappa$ amongst the list of $T$'s to be returned;

2. If 1) is true, then compute the closest candidate $T$ in both rotation and trans-
   lation space, where they are also within the pose clustering threshold;
3. If 2) is true, \( \kappa \) is updated and its pose is updated by calculating the average of all clustered poses;

4. If 2) is false, then remove candidate \( T_j \) from the list if it is full, and replace it with \( T_i \).

## 5.2 Evaluation: Interest Segment Registration Success Rate \( \xi \)

When registering scene interest segment \( R_S \) with model interest segment \( R_M \), \( R_S \) will be designated as our reference surface \( Q_{\text{ref}} \) and \( R_M \) will be designated as our target surface \( Q_{\text{target}} \), where all the point pairs \( \text{dist}_{xy} \) for \( R_M \) are pre-computed and stored during training time. The objective of this experiment is to evaluate the success rate \( \xi \) of registering \( R_M \) with its corresponding \( R_S \) in real 2.5D scene data.

As the segmentation results differ between the model and scene data, it is required to first establish the corresponding \( R_M \) to \( R_S \) for all interest segments before \( \xi \) can be measured. The corresponding segments are found by first computing the interest segment repeatability \( \Psi \) as discussed in Section 4.3.1, where the \( R_S \) with the highest \( \Psi \) is assigned as the corresponding segment of \( R_M \). For continuity of this thesis, the experiment here will be built on the segmentation result already presented in Section 4.3.2. To recap, each 3D model of 4 different objects was segmented into 8 interest segments, while 9 different 2.5D scenes were segmented into 20 interest segments respectively. The \( \Psi \) value for each segment was measured after aligning the model and the scenes with the ground truth transformation. The segment registration success rate \( \xi \) will be measured for all \( R_M \) where a corresponding \( R_S \) is present, as
shown in Figs. 4.17, 4.18, 4.19, and 4.20, where their respective $\Psi$ are provided in Table 4.5, 4.6, 4.7, 4.8.

5.2.1 Parameters Selection

To measure $\xi$, we first empirically observe the minimum rotation and translation error for $T$ such that when $T$ is used to transform the entire 3D model and followed by ICP, $T$ can be refined within $10^0$ rotation error and $1 \, mm$ of the ground truth transformation. Based on this criteria, it is determined that $T$ must be within $15^0$ and $2.5 \, mm$ of the ground truth transformation in order to qualify as a successful interest segment registration.

Beside defining an error threshold for $T$, some other parameters have to be fine tuned for this evaluation. First, to provide more stable results using 4PCS, a wide base $\Pi$ of minimum length $60 \, mm$ is used. Therefore, all interest segment whose size is underneath $3600 \, mm^2$ are removed since no $\Pi$ can be extracted for matching. To allow some relaxation in finding all corresponding congruent bases $U$, $\delta$ is set at $10 \, mm$, and $\chi$ is set at $5 \, mm$. Our hash key $Dist_{xy}$ for fast access of 4PCS point pairs in $R_M$ is quantized into $2 \, mm$ per bin. The point pairs within the next neighbouring bins to the immediate left and right of the current bin is also included as all potential matching point pairs. Lastly, since $\delta$ and $\chi$ allow some flexibility in selecting an approximate congruent base, we further downsample $R_M$ so that it can reduce the number of $Dist_{xy}$ to be stored and checked during run-time. Accordingly, $R_S$ can also be downsampled using the same rate. Both $R_M$ and $R_S$ are downsampled to $2$ points per $1 \, cm^2$ in this test.

The following sections proceed by first studying the effect of changing two 4PCS
related parameters, namely $L$ and $K$ in affecting $\xi$. Then, the quality of the interest segment, including $\Psi$ and $\varphi$ are analyzed.

### 5.2.2 4PCS Parameters

The 4PCS is a RANSAC-based data-fitting algorithm, therefore an important parameter that affects $\xi$ is the number of RANSAC cycles $L$. In this case, $L$ is actually the number of bases $\Pi$ generated from $R_S$.

In an ideal scenario, let’s assume $R_S$ is perfectly segmented and its data sampling and depth resolution are all identical to that of the corresponding $R_M$. The required number of $L$ in order to successfully retrieve a correct bases in $U$ can be determined. This is because for any given $\Pi$ in $R_M$, the number of coplanar bases that match $ratio_1$ and $ratio_2$ are known. Therefore if $R_S$ is identical to $R_M$, then $L$ can be predetermined (i.e. $L = L_D$) and the randomness in the 4PCS algorithm is eliminated.

It is nevertheless unrealistic to visit all point pairs that satisfy the congruent coplanar criteria if the 3D surfaces are densely sampled and there exist abundant point pair candidates. An efficient trade-off that optimize the probability of retrieving the correct base in $R_S$ is to sample point pairs that are spatially distributed across $R_S$ evenly.

Unfortunately, due to data discrepancies between the model and the scene, segmentation repeatability and object occlusion, the number of RANSAC cycle to guarantee a successful registration is no longer bounded by $L_D$. For any given $\Pi$, a sufficient number of $L$ therefore is required to cover different spatial samplings of $\Pi$ in $R_S$.

In the following experiment, we simply empirically measure $L = 100, 200, 400, 800$ for
all coplanar bases in order to achieve a successful registration. To account for randomness in the RANSAC-based 4PCS algorithm, our success rate will be computed from the average of repeating each segment registration process 100 times.

A second parameter to be considered is the number of $T$’s to be returned, denoted as $N_T$ as discussed in Section 5.1.3. To count as a successful registration event, only a minimum of 1 out of $N_T$ T’s that is within the $15^0$ and 2.5 $mm$ threshold is required. In this experiment, $N_T = 1, 10, 50$ are evaluated. Again, each registration is repeated 100 times and its average is the final $\xi$.

Let’s first consider $\xi$ computed based on all interest segments. In total, there are more than 120 interest segments found from the 9 2.5D scenes, with each object having about 30 interest segments across these scenes. Based on all these interest segments, $\xi$ are evaluated against different $L$ and $N_T$ values as shown in Figure 5.3. The first observation is that by increasing $N_T$, we can drastically improve $\xi$ without even increasing $L$. This means that our hypothesis that ‘registration score based only on the interest segment reprojection itself is not sufficient’ is correct. $\xi$ is approximately doubled when $N_T$ is increased from 1 to 50.

Predictably, our second observation is that by increasing $L$, $\xi$ also increases according for each object. However, it can be seen that different $L$’s are needed for different objects to achieve a certain level of $\xi$. For example, interest segments from the Angel object achieve a higher $\xi$ at the same $L$ compared to other objects. Possible reasons include the shape uniqueness of an interest segment, the size of the segments as discussed in the next paragraph. This may also be due to the repeatability and completeness of the segment, as discussed in the Section 5.2.3.
CHAPTER 5. REGISTRATION OF 3D INTEREST SEGMENTS

94

(a) Results for Angel Object

(b) Results for BigBird Object

(c) Results for Gnome Object

(d) Results for WatermelonKid Object

Figure 5.3: Segment registration success rate for all interest segments $\xi$ vs the number of RANSAC cycles $L$ (i.e. number of coplanar base selected) and the number of retrieved transformation $N_T$

Since a RbR system only requires a minimum of one accurate segment registration to successfully recognize an object, let’s also consider the interest segment that recorded the highest $\xi$ per scene, as shown in Figure 5.4. At $L = 800$ and $N_T = 50$, an interest segment extracted from the Angel, Gnome, and WatermelonKid objects on average can successfully be registered with a high probability of over 95%, with the BigBird at a slightly lower rate of 80% due to its part having many shape ambiguity, which will be discussed shortly. Examples of successfully registered interest segments
CHAPTER 5. REGISTRATION OF 3D INTEREST SEGMENTS

(a) Results for Angel Object

(b) Results for BigBird Object

(c) Results for Gnome Object

(d) Results for WatermelonKid Object

Figure 5.4: Average segment registration success rate $\xi$ of only the best aligned interest segments per scene vs the number of RANSAC cycles $L$ (i.e. number of coplanar base selected) and the number of retrieved transformation $N_T$ per object in each tested scene are illustrated in Figure 5.5.

Another important observation is that the BigBird object has a much lower $\xi$ compared to other objects. A first reason, as shown in Figure 5.6, is because of unsuccessful registration due to shape ambiguity of different object parts. For instance, both the head and the tail are sphere-like parts, and the neck is a cylindrical-like part that can be be registered in many different orientations and positions. This results in 4PCS generating many bases in $U$ with high registration scores. A second reason
Figure 5.5: Selected examples of successfully registered model interest segments from each object in each tested scene
CHAPTER 5. REGISTRATION OF 3D INTEREST SEGMENTS

Figure 5.6: Low registration success rate due to shape ambiguity

is related to $N_{R_M}$, which is the number of points in each $R_M$. The BigBird object is much larger in size in comparison to other objects in this test. Hence, each $R_M$ will be relatively larger since all objects are divided into 8 segments. A disadvantage of having a larger interest segment is that it exponentially increases the number of $\Pi_{pair1}$ and $\Pi_{pair2}$, resulting in a higher $L$ value required to select the correct base pairs.

5.2.3 Interest Segment’s Repeatability and Completeness

To understand the effect of repeatability of an interest segments on $\xi$, note that the results here focus only on data using $N_T = 50$ since it has been proven in previous test that a higher $N_T$ will result in a more reliable registration. In this test, we first consider the average repeatability score from all interest segments of all objects collectively ($\sim 120$ interest segments in total), grouped based on their repeatability scores.

As shown in Figure 5.7, a highly repeatable segment can be successfully registered with a much lower RANSAC value. This is because when selecting bases from $U$ to match with $\Pi$, the chance of selecting the correct congruent base from a more
repeatable interest segment is much higher. Unfortunately, due to occlusion in the scene, a high repeatability does not always translate monotonically into a higher \( \xi \). Hence, the slight fluctuation in repeatability across different groups of increasing repeatability.

When the visibility or the occlusion is also considered as illustrated in Figure 5.8, where interest segment is now grouped based on completeness (interest segment visibility multiplied by repeatability), this slight fluctuation disappears and it can be seen that \( \xi \) increases monotonically according to increasing completeness of interest segments. An important observation here is that for interest segments with lower completeness, it will need a much higher \( L \) to increase \( \xi \). On the other hand, a relatively low \( L \) is needed to achieve \( \xi \) for more visible and repeatable segment as the chance of picking the correct corresponding coplanar base increases. This evaluation shows that on average if an interest segments is \( \geq 50\% \) complete, then it can be successfully registered with a corresponding \( R_M \) with a probability of 100\% at \( L = 200 \).
Figure 5.8: Average segment registration success rate $\xi$ of interest segments at different completeness percentage vs the number of RANSAC cycles $L$

5.2.4 Summary and Discussion

In this chapter, the 4PCS registration algorithm is explored, improved, and applied to the registration of 3D surface interest segments. Through our analysis using various objects and scenes, it can be seen that at least one interest segment per scene can be registered using the 4PCS with a high success rate. Furthermore, it can also be observed that more reliable and predictable registration of interest segments can be derived by considering a number of factors including:

- Increasing the repeatability of interest segment, which significantly lowers the number of RANSAC cycles $L$ required to yield a successful registration event;

- RANSAC cycles $L$ need to be higher when dealing with object parts having a high level of shape ambiguity;

- Size of the interest segments is also important as it exponentially increases the number of point pairs to be inspected, similarly increasing the number of RANSAC cycles $L$ required to find the correct corresponding base.
In general, it is shown here that our proposed interest segmentation results together with a carefully designed registration method, can provide a reliable system to facilitate interest segment registration. This enable the possibilities of recognizing objects based purely on registration without the use of traditional methods such as matching local or global shape descriptors.
Chapter 6

Recognition by Registration

Our final chapter presents and evaluates an object recognition framework that integrates the proposed interest segment extraction and the segment registration technique presented in previous chapters, called: Recognition by Registration (RbR). To demonstrate the robustness of the RbR system, we evaluate the object recognition success rate in 2.5D scenes collected from different sensors modalities, including an accurate LIDAR sensor and a consumer-grade noisy 1st generation Kinect sensor. To further understand the contribution of this work in the current literature, the RbR system will be be compared with other popular LSD-based approaches including Spin-Images [44] and VD-LSD [91].
6.1 3D Object Pose Retrieval from Registration of Interest Segments

An overview of the RbR framework is illustrated in Figure 6.1. During the offline stage, interest segments are first extracted from a 3D model where each 3D point is labelled with its corresponding interest segment $R_M$. To facilitate fast 4PCS registration at run-time, the distances $\text{Dist}_{xy}$ between all point pairs $(p_x, p_y)$ are computed and stored for each $R_M$. Both the labelled 3D points along with the computed point pairs can simply be stored as a binary file.

At run-time the process is mainly divided into the hypotheses generation phase and the pose verification phase. During hypothesis generation, the scene is first segmented into a set of respective $R_S$ segments. The system then exhaustively performs pairwise registration on all $R_S$ with all $R_M$ of a target object, by using the segmentation and point pair information precomputed during offline. Recall that to increase the segment registration rate $\xi$, multiple candidate transformations $T$ per segment registration are
returned, where the number of $T$’s returned is denoted as $N_T$. Let there be $s$ scene interest segments $R_S$, $m$ model interest segments $R_M$, and $R$ models. The total number of pose hypotheses to be later verified will then be $O(N_TRms)$. In practise, we have found $m$ and $s$ to be relatively small, on the order of 10 each for sufficiently complex models and scenes, and $N_T$ is of a similar size.

A verification score is calculated by estimating the number of point correspondences between the transformed 3D model using each $T$ and the 2.5D scene according to a closest neighbour criteria. However, it can be a time-consuming process to verify each reported $T$, and thus the actual number of $T$ to be verified is significantly reduced by first filtering out any $T$ that has a lower $\kappa$ (overlapping score between interest segments discussed in the previous chapter), for which the segment registration is likely to be wrong.

In addition, the verification process can be accelerated if a heavily down-sampled 3D model is used, granted that the down sampled subset maintains a good geometric and shape representation of the object. The assumption is that as long as the object instance is not heavily occluded, then the correct $T$ that gives the highest verification score will naturally be selected as the retrieved object pose. Optionally, the verification score can be improved by applying ICP on the down-sampled 3D model. Once the retrieved pose $T_{RP}$ with the best verification score is found, it is further refined using a full ICP applied on the full resolution 3D model.

### 6.2 Evaluation: Recognition Success Rate $\Upsilon$

#### 3D Models and Scene Data

Our list of objects tested includes 9 different free-form objects as shown in Figure 6.2,
Figure 6.2: Nine different 3D objects used in our experiments covering objects consisting of a variety of shapes. This ranges from garden ornament objects that are constituted of many different free-form surfaces, to some common man-made objects such as a pipe, a water-can, and a bird-house that consist of many uniform and primitive 3D surfaces such as cylinders and planes. All 3D point cloud models are accurately acquired using a Next Engine 3D laser scanner [75] up to a depth resolution to 100 microns. The 3D models are manually reconstructed from stitching multiple 2.5D scans acquired by rotating the objects using a turn table. For more efficient processing, the surface sampling densities for all 3D models are down-sampled to $\sim 20$ points per $cm^2$.

Two different sets of scenes data are used across the experiments, including 50
scenes acquired using a VIVID LiDAR scanner [50] (depth accuracy up to 220 microns) consisting of objects a) to e), and 50 scenes using the Kinect sensor (depth accuracy up to 2 mm at a distance of 1 meter) consisting of objects d) to i). An example of each LiDAR scene and a Kinect scene can be seen on the top row of Figure 3.8 in Chapter 3. Our scenes are arranged by randomly positioning and orienting the objects, with partial occlusion are allowed. Excessive occlusion (e.g. qualitatively more than $\sim 50\%$) are avoided since our current verification process uses the number of overlapping points between the model and scene to determine a final pose of an object. Ground truth pose for each object in the scene is computed by manually defining 10 correspondences between the object in the scene and its 3D model, generating an initial transformation from these correspondences, and then refining the transformation by ICP with a nearest neighbour search radius of 5 mm.

**Experiment Parameters**

In this experiment a true positive transformation $T_{TP}$ after ICP refinement, is defined as within $10^0$ and 1 mm of the ground truth pose. Note that the BirdHouse object is a symmetric object, and therefore its symmetric pose that rotates about the symmetric z-axis is also defined as a correct pose. Due to the non-deterministic nature of the 4PCS algorithm, similar to previous tests where the registration rate $\xi$ was evaluated, the object recognition success rate $\Upsilon$ per object per scene is the average computed from repeatedly retrieving the same object from the scenes 100 times. Note that $\xi$ and $\Upsilon$ are two different measurements because even in cases where registration fails for one interest segment, a success recognition event can still occur as long as there is another interest segment that is registered within an error distance that can later
be successfully refined using ICP.

To increase the effectiveness when registering interest segments, different $N_{merge}$ were determined qualitatively for each object to assert some control over the size of the interest segments (see Section 5.2.2) given that the objects themselves were of different sizes. Figure 6.3 shows the $N_{merge}$ interest segments extracted for each model. For our scene segmentation, we approximated the number of segments to be formed depending on the objects in the scene. For example in a LiDAR scene that include 5 objects, $N_{merge}$ is set at 28. See Figure 4.15 for selected segmented LiDAR scenes but with a different $N_{merge}$. For the Kinect scenes that include 6 objects, $N_{merge}$ is set at 26, as shown in the example scenes in Figure 6.4.

Other parameters related to the interest segment extraction and the 4PCS registration are based on the values used in previous experiments discussed in section 4.3.2 and section 5.2. The $\kappa$ used for filtering out $T$ to be verified is set at 0.2, which was empirically determined based on our observation in Figure 5.8 that even at $L = 800$, the successful registration rate $\xi$ is only less than 50 percent. All our code is implemented in C++, and the experiments were performed on a Intel Quad Core2 (2.3 GHz × 4) with 8 GB RAM running on a 32-bit Linux system.

A breakdown of the time needed at each step of the RbR system can be seen in Figure 6.5. It can be seen that the segmentation of the scene and the loading of the pre-segmented model only occupy a very small portion of the recognition pipeline. The majority of time is spent on the exhaustive registration of all $R_M$ and all $R_S$. It can be seen that per segment, our optimized 4PCS perform extremely fast, which takes only about 1 second to perform registration. This is a drastic improvement over the original 4PCS algorithm. However, the entire system that requires matching of all
Figure 6.3: Interest segments extracted for the nine 3D objects used in the experiment, the number behind indicates its $N_{merge}$. 

(a) BigBird,12  
(b) Gnome,12  
(c) Zoe,8  
(d) WatermelonKid,12  
(e) Angel,8  

(f) Pipe,7  
(g) WaterCan,5  
(h) BirdHouse,7  
(i) Elmo,10
Figure 6.4: Interest segments extracted from selected 2.5D Kinect scenes used in this experiment
PHASE I: Interest Segment Extraction
1. Load quantized point pairs and interest segments of model: 0.89 s
2. Interest segment extraction for scene point cloud: 2.58 s
   2.1. Interest point detection via DoN (include normal estimation): 0.63 s
   2.2. 3D boundaries reconstruction: 0.64 s
   2.3. Fast 3D region growing: 0.58 s
   2.4. Merging of interest segments: 0.73 s
TOTAL: 3.47 s

Phase II: Recognition by Registration with $L = 400$, $N_T = 50$ (time shown here per one $R_M$ vs. one $R_S$)
1. Select a co-planar base from $R_S$: 0.11 s
2. Find $U$ from $R_M$: 0.57 s
   2.1. Prepare voxel map to find intersection point: 0.02 s
   2.2. Retrieve all matching $\Pi_1$ from $R_M$: 0.20 s
   2.3. Retrieve all matching $\Pi_2$ from $R_M$: 0.17 s
   2.4. Calculate intersection between $\Pi_1$ and $\Pi_2$: 0.04 s
   2.5. Test if a candidate base in $U$ is congruent: 0.14 s
3. Overlap calculation, transform and re-project $R_M$ on $R_S$: 0.16 s
4. Pose clustering of all candidate $T$: 0.00 s
5. Re-projection verification using full model: 0.29 s
TOTAL: 1.13 s

*Example, if there are 10 $R_M$ and 20 $R_S$, then total time required $\approx 3.47 s + 20 \times 10 \times 1.13 s = 229.47 s$

Figure 6.5: Time breakdown of the RbR system.

segments is still computational expensive, in which potential solutions are addressed in our future work section in Chapter 7.

6.2.1 Recognition Rate $\Upsilon$ in LiDAR and Kinect Scene

To understand the performance of the RbR system, it is necessary to evaluate $\Upsilon$ against different RANSAC cycles $L$. Since our interest segments extracted per object are qualitatively customized, in theory, a smaller $L$ can be used to sample sufficient number of $\Pi$, while $N_T$ is fixed at 50. We tested $L = 10, 50, 200$ and $400$ in this experiment.

Figure 6.6 illustrates the $\Upsilon$ for the 5 objects present in our 50 LiDAR scenes,
as \( L \) is increased. As expected, \( \Upsilon \) increases drastically as \( L \) is increased from 10 to 400, in which on average \( \Upsilon \) is increased from \( \sim 40\% \) to \( \sim 95\% \). While both the Angel and Zoe object recorded a 100% recognition rate at \( L = 400 \). The slightly lower \( \Upsilon \) of Gnome, WatermelonKid, and BigBird is most likely due to the segment repeatability \( \Psi \) of the interest segments visible in the scenes. As we knew already that lower segment repeatability \( \Psi \) combined with the visibility of an object in the scene can have a negative impact on the registration rate even with a low \( L \). Overall at \( L = 400 \), the RbR is able recognize an object used in this experiment with an impressive \( \Upsilon = 95.8\% \).

For the results obtained from the Kinect scenes, Figure 6.7 illustrates the recognition rate \( \Upsilon \) for the 6 objects present in our 50 Kinect scenes, as \( L \) is increased. It can be seen that both the Angel and WatermelonKid object maintains similar performance in Kinect scenes compared to the LiDAR scenes, despite the drastically lowered depth resolution. Both objects reached a recognition rate of 100% at \( L = 400 \). However, it can be observed that in general \( \Upsilon \) in Kinect scenes outperformed that in
CHAPTER 6. RECOGNITION BY REGISTRATION

Figure 6.7: Results from recognizing various objects in 50 Kinect scenes using different number of RANSAC iterations ($L$).

LiDAR scenes at a lower $L$. This is most likely because our objects are placed at a closer distance to the Kinect sensors compared to the LiDAR scenes, resulting in a more dense 3D data. The denser data set allows a more accurate $\kappa$ to be estimated for each segment registration, which increases the probability of returning the optimal $T$.

A last observation is that the relatively lower $\Upsilon$ of the Pipe, BirdHouse, and WaterCan, as compared to the Angel, WatermelonKid, and Elmo objects ($\Upsilon > 99\%$ at $L = 400$) in the Kinect scenes. This is because of the shape ambiguity found in the former objects, which is an under-constrained parameters in our object recognition system. As shown in Figure. 6.8, the cylindrical part of the pipe object can easily be mismatched as the II selected can also has be matched with many other congruent bases. Similarly, the planar surface on the BirdHouse object can easily slip (i.e. translate) when registered. Nevertheless, these highly ambiguous objects that
CHAPTER 6. RECOGNITION BY REGISTRATION

(a) incorrect object pose  
(b) correct object pose

Figure 6.8: Difficulty in handling nearly symmetric objects in the absence of distinctive object part. In this case, the distinctive part that is missing is the third tubing of the pipe.

usually can be challenging for LSD matching due to the lower distinctiveness of the object parts, still recorded a respectable $\Upsilon$ at about 85%, 75%, and 90% respectively with $L = 400$. Figure 6.9 and Figure 6.10 illustrate objects that were successfully recognized in selected challenging LiDAR and Kinect scenes in this experiment.

6.2.2 Comparisons with other LSD-based Approaches

The second part of this experiment is to compare our RbR performance against the two LSD-based approaches discussed in Section 2.1.1: Spin Images [44], used for reference in many related works, and; VD-LSDs [91] that theoretically optimizes the performance of local shape descriptors by selecting the best subsets within the principal component space through training\(^1\). As in previous sections, all techniques are

\(^1\)A special thanks to the original authors of the VD-LSDs, who has provided the C++ code for comparison in this test. Spin image, known as a subset of VD-LSDs is derived from the same code.
Figure 6.9: Recognizing various objects in selected cluttered and occluded LiDAR scenes. The retrieved true positive object pose $T_{TP}$ are illustrated with an overlay model in green.
Figure 6.10: Recognizing various objects in selected cluttered and occluded Kinect scenes. The retrieved true positive object pose $T_{TP}$ are illustrated with an overlay model in green.
Table 6.1: Object recognition success rates ($\Upsilon$) of different approaches in LiDAR cluttered scenes

<table>
<thead>
<tr>
<th>Method</th>
<th>Bird</th>
<th>Gnome</th>
<th>Zoe</th>
<th>Angel</th>
<th>Kid</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spin Images</td>
<td>45.5</td>
<td>40.2</td>
<td>37.3</td>
<td>38.9</td>
<td>50.6</td>
<td>42.5</td>
</tr>
<tr>
<td>VD-LSDs</td>
<td>90.7</td>
<td>82.3</td>
<td>86.5</td>
<td>83.7</td>
<td>88.4</td>
<td>86.2</td>
</tr>
<tr>
<td>RbR</td>
<td>97.8</td>
<td>88.7</td>
<td>100</td>
<td>100</td>
<td>92.3</td>
<td>95.8</td>
</tr>
</tbody>
</table>

tested against the LiDAR and Kinect scene data sets. To ensure fairness in comparison, descriptors are computed and generated at interest points extracted using the DoN operator, which is the same operator used in constructing our interest segments. For VD-LSDs, 10 different scenes from both the LiDAR and Kinect data sets were used in training to obtain the optimized descriptors for each object. The number of RANSAC cycles used in selecting the correct 3-points base to estimate $T_{RP}$ is set at 5000. For RbR, all the parameters are kept the same as in the first part of the experiment, with $L$ fixed at 400.

**Comparisons using LiDAR Scenes**

Table. 6.1 shows the $\Upsilon$ of each object across all 50 LiDAR scene data, where our technique can be seen to consistently out-perform both local-based techniques. The VD-LSDs display a much improved recognition rate compared to Spin Images, but RbR offers even better recognition rate across all objects. The superior performance is most likely attributed by the technique’s robustness in dealing with noise and scene complexity, that is the ability to handle increasing number of objects in a scene. The effect of both factors become more apparent in the noisy Kinect data, as is studied in the next section.
CHAPTER 6. RECOGNITION BY REGISTRATION

(a) Recognition rate in scenes with different number of objects

Figure 6.11: Recognition rate of Spin Images, VD-LSDs, and RbR due to varying degree of scene complexity in Kinect scenes

Comparison using Kinect Scenes

Instead of directly measuring $\Upsilon$ as we did for the LiDAR data, we will study how $\Upsilon$ is affected by the number of objects in a scene, defined as scene complexity here. The new scenes are generated from the 50 Kinect scenes used in previous experiments. By using the ground truth of each object in a scene, objects (i.e. points within the nearest neighbourhood of the ground truth 3D model) are removed to generate the desired number of objects in a scene. 50 different scenes of 1, 2, 3, 4, 5, and 6 objects (i.e. a total of 300 scenes) are generated for each object. The intention to generate scene data in this fashion is to ensure the occlusion percentage of an object of interest remains the same in scenes of varying objects, and will not be a contributing factor in determining the recognition rate.
Figure 6.11 illustrates that even when there is only one object in a scene (i.e. simple pose determination), registration of RbR already offers a much more attractive recognition performance against local approaches. This is because the discriminative power of descriptor based methods greatly reduces when the data becomes noisy, making it hard to establish unique descriptors across different objects. As the number of objects increases, this problem become even more apparent in local approaches as the recognition rate dropped rapidly due to the confusion caused by other objects, where the noisy 3D surfaces offer little to generate informative descriptors regarding the object of interest. On the other hand, the performance of segment registration dropped only slightly from 94.2% in the one object scene to 91.9% in the six objects scene.

6.3 Summary and Discussion

The RbR framework was tested extensively using various 3D objects in 2.5D scenes collected from different sensor modalities. Despite significant differences between the accurate LiDAR sets and noisy Kinect sets, our proposed RbR system was shown to be extremely robust, recording high object recognition rates $\Upsilon$ for most objects in both sets of data. A deciding factor in our RbR performance is the number of RANSAC cycles $L$ required, in order to provide good sampling of the 4PCS $\Pi$ in generating our pose hypotheses. In general at $L = 400$, most of our objects can be recognized at a rate of $> 90\%$. We also found that for 3D surfaces where ambiguous parts exist, $L$ must be increased accordingly due to $\Pi$ that can be easily mismatched, in order to achieve a good recognition rate. When comparing RbR with systems using
local shape descriptors, it was found that RbR offered superior Υ in both LiDAR, and more obviously in Kinect data. While in noisy data, LSDs can be easily confused with other objects, the registration-based RbR frameworks ignores the missing finer depth information, and provide a much more robust solution across all data.
Chapter 7

Conclusions and Future Work

Recent years have seen significant advances in solving the difficult object recognition problem, mainly by establishing point correspondences using local features or by matching a compact global representation of the target object. Coincidentally, improvement in 3D data sensing technologies caused an equal surge in interest within the community to utilize geometric information from depth data, to compliment existing recognition systems based on image intensity, or even to use depth alone to facilitate object recognition. The latter case in particular motivated the use of local shape descriptors (LSDs) and global shape descriptors (GSDs), where both reached a certain level of maturity and achieved good results. However, when dealing with different depth resolutions and 3D surface sampling rate, LSD performance was inevitably affected by the corrupted shape descriptors due to loss in finer local depth information, while GSDs suffer from outliers due to other objects in cluttered scenes.

To tackle the shortcomings in existing object recognition methods based purely on depth data, we have introduced the Recognition by Registration (RbR) framework, which revisited two classic problems in computer vision:
1. Segmentation: Used often only in pre-processing of image so that relevant information can be extracted from specific regions; and

2. Registration: Used as a post-processing stage to refine object pose, or used to align two partially overlapped data sets in data reconstruction.

Precisely, RbR first extracts continuous surface segments from a 3D model. Then, pairwise registration between any surface segments extracted from a 3D model and 2.5D scene generates a list of 6 DOFs pose hypotheses, where the correct object pose is later verified using support of the entire 3D model. An attractive aspect of this approach is that it by-passes the matching of LSDs that may be prone to noise corruption, or matching GSDs that is prone to outliers caused by clutter and occlusion. A minimum requirement of the RbR system is that only one successful registration between surface segments can result in a successful recognition event.

One challenge of building a RbR system is the requirement to segment 3D surfaces (in particular free-form surfaces) both repeatably and robustly. To this end, we have proposed a novel surface segmentation algorithm that drew inspiration from the extraction of 3D interest points, that are more commonly used as a pre-processing tool in localizing the generation of LSDs. In our analysis, it was found that dense sets of interest points demonstrated high robustness to changes in both data sampling and depth resolution. To inherit this repeatability trait, a greedy approach was employed in connecting the interest points into continuous 3D surface segment boundaries, which was achieved by following a smoothness constraints imposed by the orientation estimated at each interest point. Then, a fast 3D region growing coupled with a merging algorithm formed our Interest Segments. Repeatability of interest segments were experimentally tested using 3D models with synthetically added noise
and realistic 2.5D scene data including object occlusions. It was observed that interest segments extracted using the proposed algorithm in general maintain similar repeatability across all different data. Inevitability, our segment repeatability varies depending on the geometry of the underlying 3D surface, due to the algorithm directly inheriting the type of geometric edges detected by the particular interest point operator employed. Overall, we found that interest segments derived from a dense sets of DoN interest points achieved an average repeatability of 60%, with the highest per scene averages at 80% accuracy using various objects with free-form surfaces.

Building on the high repeatability rate in surface segmentation, an equally robust registration method was also presented. In this work, the RANSAC-based 4 Points Congruent Set (4PCS) registration method was adapted into our framework with great success. Proposed strategies were presented in order to optimize the performance in selecting the 4 points coplanar bases in 3D model and corresponding 2.5D scene segments to improve interest segment registration rates. We found that a successful registration event between interest segments hinges on both the repeatability and the visibility of the interest segments. Since object occlusion cannot be manipulated, increasing the repeatability of the surface segmentation algorithm becomes a crucial factor. In our test, at least one interest segment extracted from each scene using our proposed segmentation algorithm can be registered with high probability. It was also found that the registration rate can be increased by: 1) Controlling the size of the interest segments which exponentially decreases the number of point pairs required to find a matching candidate pose, and; 2) When registering interest segments with high shape ambiguity, a higher number of RANSAC cycles is required to achieve the desired registration accuracy.
Finally, our presented segmentation and registration algorithms were combined with a verification process to retrieve the final pose of an object. We tested our RbR system using both accurate LiDAR and noisy Kinect data, and found that our algorithm is extremely robust in that it achieved similar performance across the two data using the same algorithmic parameters. For free-form objects with more distinctive surfaces, our average object recognition rate is $> 95\%$ at 400 RANSAC cycles for finding a coplanar base set. For objects containing more ambiguous surfaces that can easily cause confusion during surface registration, RbR still achieved an average object recognition rate of $> 80\%$. Further experimentation with two competing LSD-based approaches showed not only that RbR provides superior performance in accurate scene data, but that the advantage that RbR offers becomes apparent in noisy data where LSD-based approaches failed as an object can become easily mismatched with other objects in a cluttered scene.

In conclusion, the RbR framework proposed overcame some notable difficulties in the current literature of 3D object recognition. Based on the positive findings in this work, our hope is to challenge existing techniques to consider a different alternative, or by complimenting the information obtained through robust registration of segmented surfaces, towards developing a more intelligent and robust object recognition system for computer vision applications.

**Future Work**

There are number of possible directions that extend research discussed in this thesis:
CHAPTER 7. CONCLUSIONS AND FUTURE WORK

Improvement in Segmentation Repeatability:

Our proposed segmentation algorithm is based on the extraction of interest points. It was proven that in this work that repeatability of different segments vary due to the type of interest points being detected by a specific interest point operator. As our segmentation algorithm directly inherits the repeatability of the interest points utilized, a more robust operator can strongly benefit the type of interest segments being extracted. More importantly, this will eventually be translated into a higher segment registration rate and object recognition rate.

Another possible improvement is the formation of 3D interest segment boundaries. Instead of employing a greedy approach of optimizing the smoothness locally, a probabilistic approach such as using a Kalman filter to predict and observe the direction of growing a 3D interest curve can be applied. The use of Kalman filtering was demonstrated successfully in an earlier work by us for extracting road from 3D data [56]. It will be a logical choice to apply similar technique to achieve better 3D interest segment boundaries reconstruction.

Improvement in the RbR Framework:

The current pose verification method is based on counting the number of point correspondences within a nearest neighbour criteria when transforming the original 3D model into the 2.5D scene. This method proved effective in scenes exhibiting limited occlusion, however in cases where objects are heavily occluded, a verification score function that is more resilient to occluded points is needed. One possibility is adapting the Potential Well Space Embedding (PWSE) GSD as a verification score. Specifically, given the candidate pose of an object, an PWSE error function can be
calculated based on the 3D points in the scene and the 2.5D view of the reprojected model in the depth camera frame of reference. By comparing the recalculated PWSE error function with that pre-computed during training, a verification score that is independent of the number of overlapping points can be obtained.

A major limitation of the RbR framework is its efficiency, because in general registration is a computationally expensive process. In the original implementation of 4PCS, it takes seconds to perform one pairwise registration. Our work here has significantly improved this to the scale of milliseconds per registration, by utilizing pre-segmented model data, and by allowing the return of multiple hypotheses to offset the loss of accuracy in registration due to significantly down-sampled segment. Despite these improvements, RbR still cannot compete with the efficient matching of LSDs and GSDs based on compact mathematical vector representation. Beside optimizing our code implementation such as using Graphical Processing Unit (GPU) and parallel processing programming, it is our goal in the future to explore more efficient way to register partially overlapped data. One such way is to employ a GSD approach on matching our interest segment (to be discussed in the next section).

**Interest Segments for LSDs and GSDs:**

Recall that the purpose of interest points is to limit the construction and comparison of LSDs to more distinctive points, so that the burden in exhaustively matching these LSDs can be reduced. Naturally, the interest segments derived from interest points can offer a similar flavour to increase the effectiveness in matching LSDs. Traditionally, any three matched LSDs can form a candidate base that transform the 3D model into the scene. The process is iterated through RANSAC until the base
that best fits the model points is found. With the scene data now segmented into
different interest segments, the random sampling of LSDs can now be restricted to
the interest points that are within the same interest segment. This is because any
three points from the same model must come from the same interest segment, that
is if the interest segment is perfectly extracted. Since in reality an interest segment
may not be perfectly segmented, the sampling of LSDs can simply be expanded
to its nearest interest segments, which should still drastically reduce the search space
and improve the object recognition rate of a LSD-based system.

Lastly, interest segments can be adapted into a GSD-based system. For example
in the case of PWSE, we can dynamically combine neighbour interest segments and
compute the PWSE error function of the combined interest segments. A 3D surface
consisting of correctly combined interested segment will yield the best matching score
when compared with the error function of the corresponding trained 2.5D view. In
this way, it is possible to eliminate the limitation of GSDs where only unobstructed
2.5D view of the object can be matched by applying our interest segments.
Bibliography


[73] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz. Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation


Appendix A

Principal Component Analysis

The objective of principal component analysis (PCA) is to reorganize the data in its original basis into a new basis that better describe the data by finding the direction where data exhibit the greatest variance. Formally, given a $m \times n$ data matrix $X$ where $m$ is the dimension of the orginal data (3D in our case), and $n$ is the number of measured data, PCA finds the change of basis $P$ that transform the data $X$ to $Y$ such that:

$$PX = Y$$ \hspace{1cm} (A.1)

where the axis of $P$ is a linear combination of the original basis. Each row of the orthogonal matrix $P$ is the new basis that better describe the measured data, such that the first and principal axis of $P$ exhibits the greatest varience, the secondary axis has the second highest varience, and so forth.

In 3D surface analysis, PCA acts as an important tool in measuring the saliency of a local neighbourhood, providing information on which orientation of the 3D surface changes most abruptly by analyzing the norm associates with each principle axis of
APPENDIX A. PRINCIPAL COMPONENT ANALYSIS

the transformed basis.

An algebraic solution to solving PCA is extracting the eigenvectors of Covariance Matrix. Let’s consider the covariance matrix $S_X$ of $X$ in its original basis coordinate:

$$S_X = \frac{1}{n-1}XX^T$$  \hspace{1cm} (A.2)

where $S_X$ is a normalized square symmetric $m \times m$ matrix. Now by theory, a symmetric matrix $A$ is diagonalized by a matrix of its orthonormal eigenvectors:

$$A = EDE^T$$  \hspace{1cm} (A.3)

where $E$ is the diagonal matrix containing the eigenvalues, and $E$ is the orthonormal eigenvector matrix.

Let’s consider the covariance matrix $S_Y$ of the new data matrix $Y$ in the new basis coordinate, which is a diagonal matrix by theory since its variance are zeros for all off-diagonal elements. $S_Y$ can be rewritten in terms of $P$ and $X$ as follows:

$$S_Y = \frac{1}{n-1}YY^T$$

$$= \frac{1}{n-1}P(XX^T)P^T$$  \hspace{1cm} (A.4)

Now we select the matrix $P$ to be a matrix so that each row of $P$ is an eigenvector of $XX^T$. Since $XX^T$ is symmetric and can be rewritten as $P^TDP$, then:

$$S_Y = \frac{1}{n-1}P(P^TDP)P^T$$

$$= \frac{1}{n-1}D$$  \hspace{1cm} (A.5)

Therefore, the principal axes and their norm can be found by extracting the eigenvectors and corresponding eigenvalues of the covariance matrix $S_X$.

Similarly, PCA can also be solved through the more general solution called Singular Value Decomposition (SVD). By definition, the SVD states that any arbitrary
$m \times n$ matrix can be decomposed as follow:

$$X = U \Sigma V^T \quad (A.6)$$

where $U$ is a $n \times n$ orthogonal matrix, $\Sigma$ is a $n \times m$ diagonal matrix, and $V^T$ is a $m \times m$ orthogonal eigenvector matrix. Note that $X$ here is instead rearranged as a $n \times m$ matrix instead of $m \times n$ in Equation A.1, where the principal axes in this case span the columns of matrix $V$. 
Appendix B

RANSAC

RANSAC, abbreviated from RANdom SAmples Consensus proposed by Fischler and Bolles [24], is a general technique for robust fitting of a parameterized model to data containing noise and possible outliers. The performance of standard model fitting methods such as least squares estimation is prone to error introduced by data polluted with outliers. RANSAC on the other hand uses an iterative approach and a minimum number of required data to fit the model to estimate a robust fitting solution. A general RANSAC algorithm is composed of three steps:

1. Select $N$ points that is required to form the model. For instance two points for a line, and three points for a plane. The model itself can be selected by using a least square technique.

2. The model is then verified by evaluating how well it fits the remaining data points, classifying each point as inlier or outlier.

3. The above two steps iterate until the model with the largest number of inliers is selected as the best model. Usually, a final least squares fit is applied on all
the inliers to reduce the residual errors.