REAL-TIME OBJECT RECOGNITION IN SPARSE RANGE IMAGES USING ERROR SURFACE EMBEDDING

by

LIMIN SHANG

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

Queen's University
Kingston, Ontario, Canada

January 2010

Copyright © Limin Shang, 2010

Abstract

In this work we address the problem of object recognition and localization from sparse range data. The method is based upon comparing the 7-D error surfaces of objects in various poses, which result from the registration error function between two convolved surfaces. The objects and their pose values are encoded by a small set of feature vectors extracted from the minima of the error surfaces. The problem of object recognition is thus reduced to comparing these feature vectors to find the corresponding error surfaces between the runtime data and a preprocessed database.

Specifically, we present a new approach to the problems of pose determination, object recognition and object class recognition. The algorithm has been implemented and tested on both simulated and real data. The experimental results demonstrate the technique to be both effective and efficient, executing at 122 frames per second on standard hardware and with recognition rates exceeding 97% for a database of 60 objects. The performance of the proposed potential well space embedding (PWSE) approach on large size databases was also evaluated on the Princeton Shape Benchmark containing 1,814 objects. In experiments of object class recognition with the Princeton Shape Benchmark, PWSE is able to provide better classification rates than the previous methods in terms of nearest neighbour classification. In addition, PWSE is shown to (i) operate with very sparse data, e.g., comprising only hundreds of points

per image, and (ii) is robust to measurement error and outliers.

Acknowledgments

First and foremost, I would like to deeply thank my supervisor, professor Michael Greenspan, for inspiring me throughout my graduate career and keeping me on the right path to complete my research. I would have never achieved the completion of this work without the guidance and help from him. I would especially like to extend my thanks to my colleagues in the robotics and computer vision laboratory for having helped me along my study.

Devout thanks to my parents for their encouragement and support throughout my degree. Heart-felt thanks to my wife Junhua for her loving support, understanding and encouragement throughout my study.

Table of Contents

Abstract	i
Acknowledgments	iii
Table of Contents	iv
List of Tables	viii
List of Figures	ix
List of Symbols	xi
Chapter 1:	
Introduction	1
1.1 Object Recognition with Range Images	1
1.1.1 Range Images	3
1.2 Motivation	6
1.3 Contributions	7
1.4 Thesis Outline	9
Chapter 2:	
Literature Review	11

2.1	Object Recognition	11
2.1.1	Model-based Approaches	11
2.1.2	Summary and Discussion	17
2.1.3	Appearance-based Approaches	18
2.1.4	Summary and Discussion	20
2.2	Iterative Closest Point (ICP) Algorithm	20
2.2.1	The Basic ICP Algorithm	21
2.2.2	Local Minima Suppression	22
2.2.3	Speed Enhancements	23
2.2.4	Summary and Discussion	26

Chapter 3:

	Potential Well Space Embedding	28
3.1	Optimization	29
3.2	Global Versus Local Minimum	30
3.3	Object Views	32
3.4	Error Surface	34
3.5	Extraction of Embeddings	40

Chapter 4:

	Pose Determination	42
4.1	Problem Definition	42
4.2	Solution Approach	43
4.3	Experimental Results	44
4.3.1	Simulated Data	45

4.3.2	Real Data	56
4.4	Summary and Discussion	59
Chapter 5:		
	Object Recognition	61
5.1	Problem Definition	61
5.2	Solution Approach	61
5.2.1	Generic Model	62
5.2.2	Prefiltering for Performance Improvement	66
5.3	Experimental Results	68
5.3.1	Real Data Tests	69
5.3.2	Simulated Data Tests	70
5.3.3	Experiments with the Princeton Shape Benchmark	76
5.3.4	Real-time Stereovision Data Tests	78
5.4	Summary and Discussion	81
Chapter 6:		
	Object Class Recognition	83
6.1	Problem Definition	83
6.2	Solution Approach	85
6.3	Experimental Results	85
6.4	Summary and Discussion	91
Chapter 7:		
	Conclusions and Future Work	93

Bibliography	97
------------------------	----

List of Tables

4.1	Correctness vs. Number of Iterations For the Satellite	48
4.2	Correctness vs. Number of Iterations For the Molecule	51
4.3	Real Data, Satellite	58
4.4	Real Data, Freeform Objects	58
5.1	Confusion Matrix, Real Range Data. OR = average successful object recognition rate, PD = average successful pose determination rate. . .	70
5.2	Confusion Matrix, Real Range Data	70
6.1	Classification Rate On Test Set	90

List of Figures

1.1	Sample Lidar Range Image of Radarsat Satellite	4
1.2	Experiment Setup of Range Data Collection	5
1.3	Flash Lidar Range Image of Zoe	6
3.1	2D Example of Gradient Descent algorithm	30
3.2	Some Object Views	33
3.3	Views and Corresponding 3-D Error Surfaces. a) five views, point cloud b) corresponding 3-D error surfaces	36
3.4	Robustness of 3-D Error Surfaces to sparseness, sensor noise and out- liers. a) The error surface and the corresponding sparse range image that only contains of 125 points. b) The error surface and the corre- sponding range image with simulated sensor noise $\sigma = 15mm$. c) The error surface and the corresponding range image with simulated outliers.	37
3.5	Robustness of Error Surface. a) Robustness vs. Sparseness b) Robust- ness vs. Sensor Noise c) Robustness vs. Outliers	39
4.1	Algorithm Diagram	45
4.2	Test Objects	46
4.3	Correctness Vs. # of Iterations, Satellite	49
4.4	Failed Cases, Satellite	50

4.5	Correctness Vs. # of Iterations, Molecule	51
4.6	Pose estimate, Molecule	52
4.7	Results of Simulated Data, Satellite	54
4.8	Results of Simulated Data, Molecule	55
4.9	Pose Estimate, Outliers	56
4.10	Real Data, Satellite	57
4.11	Real Data, Freeform Objects	59
5.1	Generic Model	64
5.2	Test Objects, Simulated Data	65
5.3	Successful Cases, Real Data	66
5.4	Successful Cases, Real Data	67
5.5	Confusion Matrix, Simulated Data	73
5.6	Experimental Results, Non-ideal Data Tests. a) Robustness vs. Sparse- ness b) Robustness vs. Sensor Noise c) Robustness vs. Outliers . . .	74
5.7	Impact of Different Generic Models	76
5.8	Object Recognition, PSB	78
5.9	Test Objects, Real-time Stereovision	79
5.10	Recognition Result, Real-time. a) Angel b) Zoe c) Gnome d) Big Bird e) Watermelon Kid	80
6.1	Example of Object Class Recognition	84
6.2	Multiview Class Recognition	88
6.3	Class Recognition, Misclassified Models	89

List of Symbols

BHT	Bounded Hough Transform
CCA	Curvilinear Component Analysis
CCD	Charge-Coupled Device
DOF	Degree of Freedom
GHT	Generalized Hough Transform
HSI	Harmonic Shape Images
ICP	Iterative Closest Point
LFD	Light Field Descriptor
PCA	Principal Component Analysis
PDF	Posterior Probability Distributions
PSB	Princeton Shape Benchmark
PWSE	Potential Well Space Embedding
RANSAC	Random Sampling and Consensus
SI	Spin Image
UKF	Unscented Kalman filter
UPF	Unscented Particle Filter
VD-LSD	Variable Dimensional Local Shape Descriptor
E	Error Function

v_i	Independent Variable
$\mathcal{S}$	7-D Error Surface
ξ	Registration Error Function
Θ	Six Dimensional Parameter Space
$\mathcal{M}$	3-D Surface Model
$\mathbf{P}$	2.5-D Range Image
$\vec{\theta}$	Pose
R	3×3 Rotation Matrix
$\vec{t}$	3×1 Translation Vector
$Corr$	Correlation Coefficient
σ	Standard Deviation
P_i	View
$\{\vec{r}_i\}_{i=1}^N$	A Set of Discrete Rotation Vectors
r_M	Magnitude of the Perturbation
Δr	Increment of Perturbation
$\mathbf{E}_i$	Embedding
i_m	Index of the Closest Match View
$G(\cdot)$	Similarity Measurement Function
K	Number of Perturbations

Chapter 1

Introduction

Object recognition has been the subject of a tremendous amount of research for over thirty years. It is an important problem for industrial automation, and has a wide range of applications. For vision-guided robotic systems, the systems must identify the objects in a scene before they can handle them in any useful manner. Moreover, the system needs to recover the poses of recognized objects (i.e., positions and orientations) in order to perform tasks such as grasping, pick-and-place and assembly.

1.1 Object Recognition with Range Images

To recognize an object implies that the object model (i.e., 3-D model of the object, or a set of views of the object) are known a priori. Given an image of the scene taken by a sensor of unknown position and orientation, the goal of a recognition system is to identify objects in the scene by comparing them to a set of known objects in a database, and recovering their pose.

Although humans are capable of performing such vision tasks naturally and effortlessly in day-to-day life, the problem of object recognition remains challenging for artificial systems, and the main difficulties are:

1. High Dimensionality of Search Space

A three dimensional (3-D) object moving through a rigid transformation has a six degree-of-freedom (DOF) pose space. This pose space comprises 3 translational DOFs and 3 rotational DOFs. When an object moves within the pose space, its appearance varies with a fixed sensor viewpoint due to self occlusion, (i.e., the backside of the object is not visible from a particular viewpoint). For a 2-D sensor such as a charge-coupled device (CCD) camera, the shape of an object is also affected by perspective distortion.

Furthermore, an object recognition system may need to deal with hundreds or thousands of different object types, which adds an extra dimensionality to the problem, (i.e., object identity) and requires searching within a seven dimensional space. With no prior knowledge, object recognition is a global optimization problem, which requires exploration of a seven dimensional search space in order to identify objects and recover their poses.

2. Efficiency

Efficiency is one of the most important criteria for evaluating the performance of an object recognition system. However, object recognition is a computationally expensive process, and the performance of an object recognition system can decrease dramatically when dealing with large numbers of objects.

3. Background Clutter and Occlusion

Natural scenes rarely contain isolated objects, and the objects of interest may also be partially occluded by other objects. Ideally, an object recognition system should be capable of dealing with the case of background clutter and partial occlusion.

1.1.1 Range Images

Many approaches to object recognition from 2-D images have been studied, and have had some success [15,40,50,51,56]. However, these techniques are sensitive to shadows and illumination effects due to the limitations of the sensor. With the improvements of range sensors, object recognition from range images has attracted increasing interest over the past decade.

Compared with traditional 2-D image sensors, range sensors have several advantages. First and foremost, range sensors are able to provide accurate metric measurement data. In a range image, each individual pixel comprises X , Y , and Z coordinates and a signal intensity, which gives accurate metric information between the sensor and the surface points of objects in the scene. A sample range image collected by a Laser Radar (*Lidar*) system is shown in Fig. 1.1. Fig. 1.1(a) shows the original image from the viewpoint of the acquiring sensor, and Fig. 1.1(b) shows a rotated view that emphasizes the 3-D nature of the data.

Moreover, range sensors are insensitive to shadows and the effects of changing lighting conditions. This is especially true for active range sensors such as Lidar because they project their own illumination on the scene. Therefore, active range sensors are capable of working under harsh illumination conditions, such as a space environment, which can be extremely bright or dark due to its lack of atmosphere

Figure 1.1: Sample Lidar Range Image of Radarsat Satellite

to diffuse light. It can be seen from Figure 1.1 and Figure 1.2 that the Lidar not only offers 3-D information about the target, but it also exhibits a high degree of robustness to the extreme lighting conditions, as the scene has a high contrast and yet accurate and dense range data are still obtained.

Apart from these attractive characteristics, active range acquisition is slow. For a conventional Lidar system running in raster imaging mode, the data acquisition rate is $\sim 500,000$ points per second (pps), and it can take minutes to capture a dense range image. When the scene contains moving objects, the relative motion between the sensor and the target will corrupt the data with motion skew, which is the primary limitation of scanning Lidar sensors.

To deal with the problem of motion skew, the Lidar sensor can be set to a high speed mode. Instead of using raster-lines, a faster scan pattern (i.e., a Lissajous

Figure 1.2: Experiment Setup of Range Data Collection

pattern) is utilized in high speed mode. While it can increase the data acquisition rate, the range data collected under high speed mode tends to be sparse. Another attractive alternative is to use high speed range sensors (e.g., stereovision sensors and flash Lidar). Figure 1.3 shows a typical range image along with its corresponding intensity image captured by SR3000 *SwissRangertm* flash Lidar sensor. The sensor operates at video frame rates and the resolution of the image is 176×144 , which is quite low compared with that of conventional CCD cameras. In addition, it can be seen from Figure 1.3 that the quality of acquired data is far from perfect, as it has many data dropouts and contains considerable noise.

Figure 1.3: Flash Lidar Range Image of Zoe

1.2 Motivation

Many approaches to object recognition with range data have been proposed recently [3, 13, 14, 31, 32, 53, 66, 67, 71, 74]. However, the difficulty in solving the recognition problem, combined with limitations of current range sensors, leads to shortcomings in existing techniques:

1. Lack of Efficiency

Most existing object recognition techniques focus on dealing with background clutter and occlusion, and efficiency is usually a secondary consideration. Consequently, these techniques are computationally expensive, which makes them difficult to apply to time-critical tasks. We argue that efficiency is an important issue that needs to be fully addressed for industrial applications. Vision systems for industrial applications usually operate under a controlled environment, in which the background can be easily modelled and only minor or no occlusions

are present in the scene. Therefore, an efficient object recognition technique that is able to handle a small degree of background clutter and occlusion is preferred for industrial applications.

2. Data Density Requirement

Most existing object recognition techniques compute feature descriptors in the 3-D spatial domain, which require the use of dense range data [3, 44]. In addition, some techniques also need to preprocess the input range images to construct polygon meshes, which is time-consuming and sensitive to sensor noise and outliers.

3. Robustness to Sensor Error and Outliers

Most techniques involve the step of calculating surface normals in order to establish local coordinate systems, which is sensitive to sensor noise and outliers [21].

In this thesis, a novel object recognition algorithm, namely *Potential Well Space Embedding (PWSE)* [58, 59] is proposed, which fully addresses the above issues. The proposed algorithm is much more efficient than the existing techniques and is robust to a certain degree of noise, data sparseness and outliers. The goal of this work is to develop an alternative to the existing techniques, which is more applicable to industrial applications.

1.3 Contributions

In this thesis, several contributions are made to the field of object recognition with range images:

1. A new object recognition algorithm, namely PWSE, is introduced and systematically evaluated. The existence of local minima within the potential well space of the iterative closest point (ICP) algorithm has been known for some time. To the author's best knowledge, this is the first attempt to exploit the existence of local minima, and allows ICP, and potentially other local optimization algorithms used for registration, to be extended to solve the pose determination and object recognition problems.
2. The use of a generic model is proposed so that a single 3-D model can be used to compute the feature vectors for different objects during both preprocessing and runtime. The use of a generic model can dramatically simplify the algorithm as well as improve its efficiency. We also propose a practical method to construct an effective generic model, and examine the impact of different generic models on performance.
3. The PWSE algorithm is extended to include the solution to a more difficult problem, object class recognition. Both single-view and multi-view approaches are proposed. The performance of PWSE on object class recognition is systematically evaluated, and compared against existing techniques.
4. The proposed algorithm has been tested on both simulated and real data. The experimental results show the technique to be both effective and efficient. In addition, very few successful object recognition systems have been implemented in practice due to the difficulty of the problem. In this thesis, a complete object recognition and tracking system utilizing a commercial stereovision camera has been built, that is able to recognize and track at least 10 freeform objects in

real-time. To the author’s best knowledge, this is the first object recognition system that is able to recognize and track freeform objects in real-time.

1.4 Thesis Outline

The remaining chapters of this thesis are organized as follows.

Chapter 2: The chapter starts with a review of related research into object recognition techniques. As PWSE is based on optimization techniques, more specifically, the ICP algorithm, some important optimization techniques are reviewed in this chapter. In addition, ICP and its variations are also reviewed in this chapter due to its importance to the PWSE algorithm.

Chapter 3: PWSE is presented in this chapter. The chapter begins with the definition of object views followed by an introduction to the 7-D error surface and its properties. The use of the ICP algorithm to extract embeddings from these error surfaces are then discussed.

Chapter 4: In this chapter, the use of the PWSE algorithm to solve the problem of pose determination is discussed. The correctness of the resulting algorithm is verified by conducting experiments on both simulated range images and real data. In addition, the robustness to data sparseness, sensor noise, and outliers are quantitatively evaluated.

Chapter 5: By introducing a generic model strategy, which uses a single 3-D model to compute the feature vectors for different objects, PWSE is extended to solve the more difficult problem of 3-D object recognition. The PWSE algorithm for object recognition is discussed in this chapter, followed by experiments on both simulated and real range images. A practical method to build an effective generic

model, and parameter selection are also discussed in this chapter. In addition, a real-time object recognition and tracking system is described to further demonstrate that the PWSE algorithm is effective at recognizing rigid objects in real-time with sparse range data, and that it is robust to large variations in image noise.

Chapter 6 : In this chapter, we introduce applying PWSE to the problem of object class recognition. Single-view and multi-view approaches are presented. A set of tests were conducted to investigate the object recognition performance of PWSE using the Princeton Shape Benchmark (PSB) database.

Chapter 7: The thesis concludes with a summary of the work presented in the preceding chapters. The capabilities and limitations of the object recognition systems developed is reviewed, and the most promising avenues for future work on this topic are discussed.

Chapter 2

Literature Review

In this chapter, a review of existing object recognition techniques is presented. In addition, the ICP algorithm and its variations will be reviewed.

2.1 Object Recognition

Various approaches for object recognition using range images have been proposed in the literature. Based on the various ways that these algorithms represent 3-D objects, existing object recognition algorithms can be divided into two categories, namely *model-based* and *appearance-based* approaches.

2.1.1 Model-based Approaches

Model-based object recognition techniques consist of preprocessing and online recognition phases. In the preprocessing phase, a model library is first built by extracting descriptors (features) from the 3-D surface models of each object. Each descriptor, as indicated by its name, characterizes surface shape in a support region surrounding

a basis point on the 3-D model. During online recognition, the same descriptor set is extracted from the scene image, and the problem of object recognition is solved by matching the extracted descriptors with those in the library.

Ideally, only three point correspondences between the model and scene are required to identify the object and recover the transformation that aligns the model with the scene image, if these three correspondences are correct. Practically, as incorrect point matches may coexist with the correct ones, a larger set of correspondences are needed such that the object pose can be robustly resolved by utilizing robust statistically stable methods, such as Random Sampling and Consensus (RANSAC) [19] or the Generalized Hough Transform (GHT) [4]. The recovered object identities and pose estimates are then verified by aligning the models with the scene, and that which results in the maximum overlap is taken as the final solution, and refined by ICP or its several variations.

It can be seen that the model-based techniques are mainly discriminated by the manner in which local descriptors are computed. In this section, we will give a brief review of existing local descriptor techniques, grouped by the dimensionality of the descriptor.

One Dimensional Local Descriptors

Point signature, proposed by Chua and Jarvis [14], probably is the most well-known 1-D local descriptor technique. The main idea of the algorithm is to represent the surface geometry in the vicinity of a point by a 1-D contour. For each model point, a sphere is placed at the center of the point, which intersects with the surface of the object, and which creates a 3-D space curve. A plane can be obtained by fitting the

intersecting curve, and its normal serves as an estimate of the normal of the point. Another tangent plane, which is parallel to the first plane and passes through the original point, is then constructed, and the space curve is projected onto this tangent plane to form a second curve.

To deal with the ambiguity of rotating about the normal when matching the descriptor, an anchor point, which is a point on the second projected curve that is furthest from its original point, is defined for each descriptor, and the direction from the central point to the reference point is chosen as the reference vector. Then a directional frame is constructed by using the reference vector, normal vector and their cross product, and the point signature is built by computing the distances of the projected curve from the intersecting curve in a clockwise direction around the curve.

The authors tested the method with a database containing fifteen models, including four face masks, five terrain type models, five simple piecewise quartic shaped models and a propeller. A total of 16 scene images were used in their experiments, which included 15 single-object scenes, and one semi-cluttered scene. The authors reported that the method was able to correctly recognize all the objects in both single-object and semi-cluttered scenes, and the average recognition times are 44 seconds and 142 seconds, respectively, running on a SGI 4D/20 single-processor computer.

Other similar approaches include *Splash* by Stein and Medioni [66], which is defined by surface normals along contours of different radii, and more recently *Point Fingerprint* by Sun et al. [67]. In [67], a local descriptor is constructed from a sequence of contours formed by the projection of geodesic circles onto the tangent plane, and each descriptor carries information of the normal variation along geodesic circles.

The 1-D descriptors can provide a compact representation of local geometry and are efficient to compute. However, a limitation of the 1-D descriptors is their lack of desirable discriminating power. The cause of the problem is due to the loss of geometry information when encoding 3-D local geometry to a 1-D contour. For this reason, two dimensional and higher dimensional local descriptors have attracted increasing attention as they are able to offer a richer representation of local geometry than 1-D descriptors.

Two Dimensional Local Descriptors

Developed by Johnson and Hebert, the *Spin Image (SI)* [31,32] is the most well-known 2-D local descriptor technique. The idea of the SI is to represent the feature of a small surface patch around a point by a 2-D histogram. To generate the spin image, the normal to the point is first calculated, which serves as the axis of a cutting plane. While the cutting plane spins, the intersections between the plane and the surface are used to construct a set of 2-D histograms, which are called spin images and can be used to establish correspondences between scene and model points. During runtime, the spin images for scene points are generated in the same fashion and compared against the model spin images to find corresponding scene and model points. These corresponding points are then grouped based on both geometric position and orientations. The rotation and translation transformation between scene and model points are calculated from these grouped point correspondences and then further refined by a modified ICP algorithm.

To reduce the effect of self-occlusion, support angles are utilized to filter out the points that are not visible from the current viewing angle. If the support angle that is

formed by the surface normal of the point and the direction of the oriented point basis of a spin image exceeds a certain threshold, the point will be discarded. In addition, principal component analysis (PCA) is used to compress the spin images such that they can be represented in a more compact form. As the L_2 distance between two spin images in spin image space is the same as the L_2 distance represented in eigenspace, the problem of correspondence can be more efficiently solved in a lower dimensional eigenspace without sacrificing much accuracy.

Single-object scene recognition using spin images can be found in [57]. All the experiments were conducted on a 2 Ghz computer with 2 GB memory. For the simulated data test, the size of the database is 56 objects, and a total of 56 test images were used in their test. The reported accuracy of the simulated data test was slightly above 90%, and recognition time per query was 43 seconds on 50 objects. Their experiments on real data were conducted with 88 real queries against a 90 model database, and their reported accuracy was $\sim 40\%$.

Other similar approaches can be found in *Harmonic Shape Images (HSI)* by Zhang [74], *Spherical Spin Image (SSI)* by Correa and Shapiro [53], *Surface Signatures* by Yamany and Farag [71], and more recently *Local surface Patches* by Chen and Bhanu [13].

High Dimensional Local Descriptors

Mian et al. [44] proposed an object recognition and pose determination algorithm based on a tensor-based technique. In offline preprocessing, the input point cloud data is first converted into a triangular mesh, and decimated twice to construct three meshes with different resolutions. The coarsest one is used to select the feature points

in the next higher resolution mesh that is used to compute tensors, and the highest resolution mesh is used for registration refinement. The closest vertex pair constrained by an angle constraint and a distance constraint is used to define a 3-D coordinate basis, and each of these 3-D coordinate bases is used to define a $15 \times 15 \times 15$ grid centered at its origin. The area of intersection of the mesh with the grid is recorded in a third order tensor. The value of each element of the tensor is equal to the surface area intersecting its corresponding bin in the 3-D grid. The tensors are saved in a 4-D hash table.

For the single-object scene test, the tensor-based algorithm was able to achieve 95% accuracy using a total of 500 test images on a database with 50 objects. For the multi-object scene, the authors reported that the method can achieve a high recognition rate as the amount of occlusion increases. The average recognition rate of tensor-based algorithm was 96.6 percent with up to 84 percent occlusion. The time efficiency of the tensor-based experiments was not reported, although it was mentioned that their implementation was not optimized for time as it was developed in Matlab.

Frome et al. [21] introduced two high dimensional local shape descriptors, namely *3-D shape contexts* and *harmonic shape contexts*, which are designed for recognition of similar 3-D objects (i.e., different types of vehicles). The 3-D shape context is the straightforward extension of 2-D shape contexts [5], and the harmonic shape context is computed by using the monic transformation to the 3-D shape context.

More recently, Taati and Greenspan [3] proposed using variable dimensional local shape descriptors (VD-LSD) for recognition. The main idea of the VD-LSD approach is to use high (up to 9) dimensional descriptors for more accurate and robust point

correspondence. The authors generate a set of local shape descriptors for each point based on invariant properties extracted from the principal component space of the local neighbourhood around the points, and then select a set of optimal descriptors through preprocessing the models and sample training images.

The technique was tested on a total of 10 3-D models including the four models from the University of Western Australia [44], the Radarsat satellite model from MDA Space Missions, and five models from Queen's model database [22]. The test scene images include both Lidar and dense stereo images, 686 images in total. The authors reported that the average recognition rate of VD-LSD on Lidar data is 83.8%, which took 2,964 ms per image on a computer with Intel Core 2 Quad Q6600 CPU at 2.4GHz. For the tests on dense stereo images, VD-LSD was able to achieve 52.3% and 74.7% respectively when using 1,000 and 5,000 RANSAC iterations.

2.1.2 Summary and Discussion

Local descriptors play a decisive role in model-based techniques. A good local descriptor should be discriminating, robust, and computationally efficient. While low dimensional descriptors can be computed and compared more efficiently than high dimensional descriptors, they are in general not as discriminating, which leads to many incorrect point matches. To filter out these incorrect matches, a robust technique such as RANSAC or GHT, can be used. However, it is time-consuming as the computational complexity of RANSAC and GHT is a high degree polynomial in the number of incorrect point matches [55].

In contrast with low dimensional descriptors, high dimensional descriptors are able to offer more discriminating power such that more correct point matches can be

established. However, they are in general less efficient to compute and store, and in some cases require the construction of a triangular mesh first, which is a complex and time-consuming procedure.

2.1.3 Appearance-based Approaches

The appearance-based approach is more efficient than the model-based approach when the objects can be segmented from the scene image. An object is first encoded with a set of images collected from different vantages in an off-line training phase. In online recognition, the objects and their poses can be retrieved by searching for the best match between the input image and the database of stored images. As it is expensive to operate directly on image data, these images are first transformed into a lower dimensional space so that the comparison can be executed more efficiently within this lower dimensional space.

PCA is the most commonly used technique for forming the low dimensional space. The main idea of PCA is to effectively map high dimensional image data to a low dimensional subspace by reducing the redundancy while preserving as much information as possible. The directions with the largest variance of input data are first calculated in the high-dimensional input space, and then the dimension of the space can be reduced by discarding the directions with small variance of the input data. By doing so, the input high dimensional data can be approximated in a low dimensional space with minimal error among all linear transformations to a subspace of the same dimension.

Campbell and Flynn [11] were the first to apply the appearance-based technique on range data using PCA. In their work, eigenshapes are constructed based on a set of

range images in a training procedure, and these eigenshapes are utilized to construct a low dimensional subspace for object recognition and pose recovery. They tested the technique on two different databases: one contained 20 free-form objects and another contained 54 mechanical parts. The authors reported that an appearance subspace of 20 dimensions was sufficient for accurate object recognition, and the system was able to offer 91% accuracy on object recognition. The authors only considered the 2 DOF pose estimation problem in rotational subspace, and experimental pose determination results were not presented in the paper. In addition, only simulated range images generated from full 3-D models were used in their experiments, and so their range images can therefore be considered as ideal.

Scocay and Leonardis [65] extended the appearance-based technique to handle missing pixels and occlusion. The missing pixels are those pixels whose depth measurements are not available due to the architecture of range image sensors. Instead of computing the coefficients by a projection of the data onto the eigenimages, the authors addressed the problem of missing pixels by solving a set of linear equations in a robust manner to determine the coefficients. The technique was tested on simulated range data generated from six freeform objects. The experimental results showed that the algorithm was robust to missing pixels, noise, and occlusion in range images. However, their experiments were conducted on only one DOF and two DOF problems with a limited number of test objects.

2.1.4 Summary and Discussion

Robust and efficient object recognition has been tackled very rarely using the appearance-based approach. One challenge of the appearance-based approach is that it is sensitive to dropouts, sensor noise and outliers. Most real range images as illustrated in Figure 1.3 contain many erroneous regions, and these artifacts are difficult to avoid completely due to the limitations of current 3-D sensor technology. Therefore, it is impractical to directly apply the appearance-based techniques to real world object recognition problems.

In addition, the appearance-based technique suffers from the problem of combinatorial explosion. In essence, the appearance-based techniques are based on template matching, which require the training and scene images to be aligned in the same manner. For each pose of an object, a 3-D image needs to be captured and processed to index the pose of the object. To sample the 6 DOF parameter space with a certain resolution, a huge number of range images are required, which is very time-consuming. For a practical object recognition system, it in general contains tens or hundreds of objects, and the number of range images will grow linearly with the number of objects, and exponentially with the number of DOFs.

2.2 Iterative Closest Point (ICP) Algorithm

Since it was first introduced by Besl and McKay [8], the iterative closest point (ICP) algorithm has become the most prominent 3-D registration technique. The ICP algorithm works directly on 3-D points and solves the registration problem by iteratively minimizing an error function that registers the scene points to the underlying model.

2.2.1 The Basic ICP Algorithm

In the ICP algorithm, the registration error is defined with respect to the correspondences between points in the data sets. Let Θ denote the six dimensional parameter space comprising the 3 translations and 3 rotations of a rigid transformation. Given a 3-D surface model $\mathcal{M}$ of an object in an arbitrary canonical pose, and a range image $\mathbf{P} = \{\vec{p}_j\}_1^n$ of the object in a possibly different pose $\vec{\theta} \in \Theta$, the registration error function ξ between $\mathcal{M}$ and $\mathbf{P}$ at pose $\vec{\theta}$ is:

$$\xi(\mathcal{M}, \mathbf{P}, \vec{\theta}) = \sum_{j=1}^n \|\vec{q}_j - R\vec{p}_j - \vec{t}\|^2 \quad (2.1)$$

where R is a 3×3 rotation matrix, $\vec{t}$ is a 3×1 translation vector, $\vec{q}_j$ is the point on the surface of $\mathcal{M}$ that is closest to (*i.e.* corresponds to) the transformed $\vec{p}_j \in \mathbf{P}$, and $\|\vec{q} - \vec{p}\|$ denotes the Euclidean distance between two points $\vec{q}$ and $\vec{p}$.

By using closest points to approximate the true point correspondences, ICP is guaranteed to converge monotonically to a local minimum by iteratively finding the closest point sets and then solving Eq. 2.1 [8]. ICP can be stated as follows:

The iteration is initialized by setting $R = [I]$ and $\vec{t} = [0, 0, 0]^t$, with the transformation defined relative to $\mathbf{P}$ so that the final registration represents the complete transformation.

Algorithm 1 The Basic ICP Algorithm

- 1: For each point in $\mathbf{P}$, compute the closest corresponding point in $\mathcal{M}$.
 - 2: With the correspondences from step 1, compute the incremental transformation $(R, \vec{t})$ from Eq. (2.1).
 - 3: Apply the incremental transformation from step 2 to the data $\mathbf{P}$.
 - 4: Compute the change in total mean square error. If the change in error is less than a threshold or the number of iterations is beyond the predefined maximum number of iterations, terminate. Else goto step 1.
-

2.2.2 Local Minima Suppression

The convergence of ICP to the global minimum (i.e., the true pose) strongly depends on the initial pose estimate. When the transform between two data sets is large, it is well-known that ICP can be easily trapped by a local minimum resulting in an incorrect registration result, which is considered to be a limitation of the ICP algorithm. A number of solutions have been developed to improve the convergence of the ICP algorithm.

The most straightforward solution of this problem, as suggested by Besl in his initial paper [8], is that the ICP algorithm is initialized from several random locations and the best solution is chosen to be that pose with the minimum error. This method can work effectively in certain situations, but it still cannot avoid local minima completely. Moreover, it is difficult to decide the optimal set of initial states and a large number of states have to be used, which is computationally expensive.

Jason Luck et al. [41] proposed a hybrid algorithm that utilizes the simulated annealing algorithm [64] to aid ICP to converge to the global minimum. In the hybrid algorithm, the ICP is first invoked from an initial pose estimate, and it will converge to the nearest local minimum. If the local minimum is the global minimum, then the residual error should be below the error threshold and the hybrid algorithm will stop without executing the simulated annealing algorithm. Otherwise, when the residual error is larger than the threshold, the hybrid algorithm employs the simulated annealing algorithm to search about the error surface for a new start location for the ICP. The above process repeats until the error is below the threshold.

The hybrid algorithm is able to achieve the same level of accuracy as simulated annealing, but only using about one fourth the time that simulated annealing algorithm

consumed. However, the proposed technique is much slower than the ICP algorithm, and still needs a good initial pose estimation to begin with. In addition, when the error surface is relatively flat with many local minima, the hybrid algorithm may fail to find the global minimum.

An alternative approach is based on filtering techniques. Ma and Ellis [43] proposed the use of the Unscented Particle Filter (UPF) to solve the problem of 3-D registration. The UPF-based method is iterative, and is able to accurately register small data sets to the underlying 3-D model, which makes it especially useful for computer-assisted surgery. However, it requires a large number of particles (2,000) to effectively sample the Posterior Probability Distributions (PDF), which involves large computational costs. To address the problem, Moghari and Abolmaesumi [45] proposed an Unscented Kalman filter (UKF) -based technique, which replaces the UPF with the UKF. However, the UKF-based method assumes a unimodal probability distribution of the state vector, and it may fail when the assumption is invalid. In addition, both filtering-based methods still require a relatively good initial pose estimate to start with, and the convergence to the global minimum is not guaranteed.

2.2.3 Speed Enhancements

The most computational expensive processing step of ICP is the determination of point correspondences between the two data sets, which occurs at the beginning of each iteration. This correspondence determination is a form of the nearest neighbor (NN) problem, which is a classical problem in the field of computational geometry. A very common and reputedly efficient general solution is the k-d tree, which was developed by Bentley et al. [7]. If we assume that $\mathcal{M}$ and $\mathbf{P}$ are each of cardinality

N , then the ICP using a k-d tree executes in $O(N \log N)$ per ICP iteration.

Several authors have proposed solutions to accelerate the algorithm. Besides the hardware specialized techniques that use parallel computing to speed up the algorithm [38], these methods can be classified into three categories [33]:

- Reduction of the number of iterations;
- Reduction of the number of data points M and D ;
- Acceleration of the closest points computation.

The three types of acceleration techniques are quite independent and thus can be combined to further speed up the algorithm. Jost [33] also suggested that the last two methods are more effective than the reduction of the number of iterations. Reduction of the number of points trades off speed with quality of matching, since details can disappear when using only a subset of the data (control points). The acceleration of the closest points search generally has the biggest impact on the speedup. Projection methods, such as inverse calibration [9, 70] and Z-buffer projection [6] can be more efficient than the k-d tree approach [23, 25, 34] when an approximate pose estimate is available for initialization, which holds for the tracking problem.

In addition, several researchers have succeeded in tackling the real-time tracking problem by combining both high speed acquisition of 3-D data with high speed variations of ICP. Simon and Hebert [63] have developed an ICP-based real-time pose estimation system, in which several acceleration methods are applied to improve system performance including:

- k-d trees;

- Closest point caching;
- Closest surface point computation;
- Extrapolation of matching parameters by decoupling the acceleration of translation and rotation.

The system described in [63] gains much of its speed from the closest point caching algorithm which can reduce the number of necessary k-d tree lookups. The system can perform full 3-D pose estimation of arbitrarily shaped rigid objects at speeds up to 10 Hz with 32×32 cell ranged image sequences collected by CMU high speed VLSI range sensor.

Jasiobedzki and Abraham et al. [30] have developed an extension of the ICP algorithm which is capable of tracking at modest frame rates using stereo edge features as input. To speed up ICP, the distances between all pairs of points in both data sets are precalculated by creating efficient model representations off-line, which increases efficiency. The system is fairly robust to outliers and can reach an accuracy of millimeters at a tracking distance of several meters.

Rusinkiewicz [54] used a projection method with a selection of control points and a point-to-plane error metric to obtain a very fast ICP. In other stages of ICP, the author used the variation which required low computation, i.e., random sampling, constant weighting, and a threshold for rejecting pairs, to further improve the efficiency of the algorithm. Since the projection algorithm is more efficient than the k-d tree and a point-to-plane error metric has substantially faster convergence than the point-to-point metric, the system is capable of aligning two data sets in 20 ms.

More recently, Morency and Darrell [46] proposed new real-time tracking method using ICP and the normal flow constraint. By minimizing a hybrid error function

which combines constraints from the ICP algorithm and normal flow constraint, the technique is more precise than ICP alone for small movements and noisy depth data, and is more robust than the normal flow constraint alone for large movements. The hybrid tracker was tested with face tracking sequences obtained from a stereo camera using the SRI Small Vision System. The system can run at 2 Hz on a Pentium III 800 MHz when using 2500 points per frame.

2.2.4 Summary and Discussion

One important limitation of the ICP algorithm is its narrow domain of convergence to the global minimum. The main cause of the problem is that the point correspondences computed by nearest neighbor are a reasonable approximation of the real correspondences only when the displacement between two point sets is sufficiently small. Although alternative approaches as discussed above are able to improve the convergence of ICP, they all still require a good initial pose estimate, and are in general more computational expensive as random procedures, i.e., simulated annealing or Monte Carlo simulation, are involved in the methods.

Most of speed enhancement solutions imply a tradeoff between execution speed and the quality of matching. Therefore, there are potential risks that ICP gets trapped in local minima and that the tracking accuracy is degraded. In addition, a reduction of the number of points also increases the probability of occurrence of the situations.

Although, Besl suggested in his initial ICP paper that ICP could be potentially applied to the problem of object recognition, very little work has been done along these lines [36] due to its lack of efficiency and its susceptibility to the local minima. The ICP algorithm is usually used to refine the alignment of 3-D images, when the

initial pose estimation is available or is already solved by other coarse registration techniques.

Chapter 3

Potential Well Space Embedding

In essence, ICP is a nonlinear optimization algorithm, which also suffers from the problem of global versus local minima. For the 3-D registration problem, the error surface is a 7 dimensional hyper surface, which in general has a complex landscape. It is extremely difficult to avoid the local minimum completely when dealing with such a high dimensional error surface using ICP. Although many variations of ICP have been proposed to improve convergence to the global minimum, they are computational expensive, and convergence to the global minimum is not guaranteed. In this chapter, we will introduce our novel object recognition algorithm, potential well space embedding (PWSE), which in fact utilizes local minima as a set of effective feature vectors to solve the problem of object recognition in an efficient and robust manner.

3.1 Optimization

Optimization has been applied across a wide range of applications and is one of the most important mathematical techniques in the domain of engineering. Given a system with an error function E which depends on n independent variables $q_1, q_2, \dots, q_n$, the goal of an optimization solver is to find the values of q_i for which E is a minimum. For the error functions which have analytical forms, these minima may be found by calculus methods, that is, the first derivatives with respect to q_i of E are zero and the second derivatives are positive at a minimum point:

$$\frac{\partial E}{\partial q_i} = 0; \quad \frac{\partial^2 E}{\partial^2 q_i} > 0 \quad \text{for all } i \in [1, n] \quad (3.1)$$

When the error functions do not have an analytical form, gradient descent-based algorithms, such as the Gauss-Newton algorithm and the Levenberg-Marquardt algorithm, are the common approaches to solve the optimization problem. As illustrated in Figure 3.1, starting from the initial value v_1 which can be randomly chosen, if E is differentiable in a neighborhood of v_1 , then E decreases fastest if one varies from v_1 in the direction of the negative gradient of E at v_1 , $\Delta E(v_1)$. Then we can calculate the new point by:

$$v_2 = v_1 - \gamma \Delta E(v_1) \quad (3.2)$$

for $\gamma > 0$ a small enough number, then $E(v_1) \geq E(v_2)$. The process is iterated, which leads to a sequence of:

$$E(v_1) \geq E(v_2) \geq E(v_3) \dots, \quad (3.3)$$

and the gradient descent algorithm simply goes downhill in small steps until reaching

Figure 3.1: 2D Example of Gradient Descent algorithm

a local minimum $vm1$.

3.2 Global Versus Local Minimum

The gradient descent algorithm only converges to a local minimum, and the global versus local minimum is one of the most important issues to be addressed when using the gradient decent based technique. It is well known that the convergence of gradient descent methods depends highly on the initial value. For instance, there are two local minima coexisting with the global minimum in Figure 3.1. If the gradient descent algorithm is initialized with $v1$, it will converge only to the local minimum $vm1$ instead of the global minimum $vm3$. Consequently, the gradient decent algorithm

provides only a suboptimal solution in such a case.

An informal definition for a local minimum well is a region of the parameter space where the gradient descent function guides the search to a local minimum. Intuitively, if the gradient descent algorithm is initialized from any point located within a local minimum well, then the gradient descent algorithm will always monotonically converge to the local minimum in this local minimum well. As illustrated in Figure 3.1, the parameter space can be divided into 3 independent regions, specifically $w1$, $w2$ and $w3$ with each region corresponding to an unique local minimum.

The local minimum wells act as attraction regions in the parameter space. The gradient algorithm will converge to the corresponding local minimum depending on which local minimum well it is initialized within. The gradient algorithm will converge to the global minimum only when it is initialized within the local minimum well that encloses the global minimum. Otherwise it will end up at one of the local minima. If the local minimum well containing the global minimum is very small compared to the whole parameter space, a large number of samples have to be used in order to ensure that the local minimum well which the global minimum is located within can be sampled. Therefore, this will increase the computational burden dramatically as the error landscape has to be sampled extensively. For simulated annealing, if the error barriers between a local minimum are both deep, it is entirely possible that the simulated annealing may become stuck in a local minimum well which is not the global minimum, because the error barriers are too high to allow escape.

The local minimum can be very difficult to be avoid. As illustrated in Figure 3.1, the local minimum $vm1$ has a much wider minimum well than the global minimum $vm3$ and the error barriers are deep on both sides. The optimization algorithm has

much greater probability to be initialized within the local minimum well such that it will converge to $vm1$ instead of the global minimum $vm3$. Moreover, the simulated annealing will not be helpful due to the deep error barriers. For certain optimization problems, the error landscape could contain many local minima resembling $vm1$, which will make the problem even more difficult to solve.

3.3 Object Views

In PWSE, each object is represented as a set of discrete object views. We define a *view* of an object as a range image acquired from a particular sensor vantage with respect to the object's ego-centric coordinate reference frame. When an object is scanned with a conventional range sensor, only the front-facing surfaces of the object are visible from the sensor vantage. The remaining surfaces are *self-occluded*, and for this reason the resulting range images are called 2.5-D, with the object's surface bisecting a dimension along the sensor line of sight.

The object views are 2.5-D range images, acquired at a set of uniformly distributed discrete locations around the object's 3-D view sphere as illustrated in Figure 3.2. The set of object views are generated in simulation by transforming a virtual sensor to every location of a discretely sampled 3-D rotation space centered at the object model's origin, comprising a 2-D polar coordinate, and a rotation around the line of sight. By setting the three rotational increments to $(20^\circ, 20^\circ, 30^\circ)$ the rotation space is discretized into $18 \times 10 \times 12 = 2,160$ locations, and at each location a 2.5-D range image of the model is generated, representing a view. The second Euler angle ranges from 0° to 180° . As the view at 0° and the view at 180° are different, there are a total of 10 different views. The larger value of 30° was used for a rotation around the

Figure 3.2: Some Object Views

line of sight because it does not exhibit self-occlusion.

The concept of an object view used in PWSE is similar to that of aspects that are widely used in recognition from 2-D images such as [10, 16, 16, 18, 42, 69]. However, there are two main differences. First, the view defined in PWSE is 2.5-D which is able to provide more detailed geometric information of the object than the 2-D image. Second, these 2.5-D views are not organized into groups as in an aspect graph, i.e., by combining these views together based on their similarities. Although PWSE may benefit from using the concept of an aspect graph, constructing the aspect graph from 2.5-D range images is an unsolved problem, and is the subject of future research.

3.4 Error Surface

Each view corresponds to an error surface as defined by Equation 2.1. The error surface $\mathcal{S}$ is a 7-D hypersurface that is formed by convolving $\mathbf{P}$ over the complete 6-D pose space Θ , and computing the value of ξ for every transformation $\vec{\theta} \in \Theta$ of $\mathbf{P}$. Thus, $\mathcal{S} \in \Theta \times \mathbf{R}^+$, where $\mathbf{R}^+$ is the range of ξ , which is the set of non-negative real numbers. For asymmetric objects and sufficiently large point sets, $\mathcal{S}$ will have a single global minimum located at that value of $\vec{\theta}$ where $\mathbf{P}$ and $\mathcal{M}$ are correctly registered, as well as a number of local minima. Depending upon the initial pose, ICP will converge to either the global minimum, or to one of the local minima.

One interesting property about these $\mathcal{S}$ is the variety of their shapes. By examining Equation 2.1, we can note that the $\mathcal{M}$ is the same for all views of the object because the complete 3-D model is used, and the landscape of the error surface solely depends on the view of the object, which varies with the poses of the object due to self-occlusion. In model-based object recognition and pose determination, the 3-D models of the objects are already known such that the simulated views of objects can be generated. To solve the recognition and pose determination problem, we can precalculate and store the error surfaces for each view of the objects, and then search for the corresponding error surface during runtime.

In general, the error surfaces have very complex shapes and consist of many local minima. As PWSE is based on extracting features from the error surfaces, it would be interesting to examine the error surfaces directly. However, it is very time-consuming to compute entire error surfaces, and difficult to visualize such high-dimensional data. To simplify the problem, we produced the error surfaces by convolving over only the translational subspace of Θ which forms a 4-D error surface, and then used

Curvilinear Component Analysis [17] (CCA) to project the 4-D error surface onto 3-D space for visualization. Although some information is lost using this method, the projections are good enough to illustrate the basic characteristics of error surfaces.

An example of four projected error surfaces is illustrated in Figure 3.3. The plots clearly show differences among the error surfaces, which is the most important property of the error surfaces, that is, the variety of their shapes are related to their input views. In addition, it is important to notice that the result is only based on the translational subspace. For 7-D error surfaces, their shapes become more complex, and they will be even more distinctive.

For the error surface to be useful for object recognition, it has to be robust to data sparseness, sensor noise and a certain degree of outliers. In order to investigate the robustness of the error surface to data sparseness, noise and outliers, the first error surface in Figure 3.3 was regenerated using the same method, but on sparse range data, data with simulated sensor noise, and data with simulated outliers respectively. The range image used in Figure 3.3 consists of 1,000 points, and we randomly sampled 75 points from it to generate a new sparse range image. The sensor noise was simulated by introducing random zero-mean Gaussian noise to each data point. The size of the object was 200 *mm*, and the noise was set to $\sigma = 15$ *mm*. A total of 1,000 additional points were added to the original range images. To simulate outliers, spurious data points were randomly inserted into the original range images. The outliers were generated to lie near to surface points, ranging from 10% to 30% of the length of the original image's bounding box. Figure 3.4 shows the simulated range images, and their corresponding error surfaces. It shows that for the same view, the error surfaces are very similar regardless of the dramatic degradation of input range images.

Figure 3.3: Views and Corresponding 3-D Error Surfaces. a) five views, point cloud
b) corresponding 3-D error surfaces

a) Sparseness

b) Noise

c) Outliers

Figure 3.4: Robustness of 3-D Error Surfaces to sparseness, sensor noise and outliers. a) The error surface and the corresponding sparse range image that only contains of 125 points. b) The error surface and the corresponding range image with simulated sensor noise $\sigma = 15mm$. c) The error surface and the corresponding range image with simulated outliers.

The robustness of the error surfaces to data sparseness, noise and outliers was also quantitatively studied by computing the correlation coefficient between the error surfaces generated by the ideal data and the degraded data. The correlation coefficient ($Corr$) between two error surfaces $\{X_i\}_1^n$ and $\{Y_i\}_1^n$ is calculated as:

$$Corr = 1 - \sum_{i=1}^n \left\| \frac{Y_i - X_i}{X_i} \right\| \quad (3.4)$$

To measure the influence of data sparseness, a set of test images was generated by randomly sampling 1,000, 500, 250, 125 and 75 points from the ideal range images, and then computing the correlation coefficient between the error surface computed using the ideal range image, and those computed using the sparse ranges images. The result is shown in Figure 3.5 (a). It can be seen that the error surface is robust to data sparseness as when the number of points per image varied from 1000 to 125, the correlation coefficient only changed from 1 to 0.97, and when using only 75 points, the correlation coefficient was still over 0.9.

The evaluation of robustness vs. measurement error was computed by adding Gaussian noise to each point of the ideal range image. The noise was zero mean and the standard deviation (σ) varied between 0 *mm* and 20 *mm*, which is between 0% and 20% of the size of the ideal range image's bounding box. As shown in Figure 3.5 (b), the correlation coefficient barely changed when $\sigma \leq 10$ *mm*. Once $\sigma > 10$ *mm*, the correlation coefficient declined a little more rapidly, but it was still near to 0.9 when $\sigma = 20$ *mm*.

To simulate outliers, spurious data points were randomly inserted into the ideal range image with the number of outliers varying between 0 and 2000 points, which is between 0% and 200% of the number of points in each image. The result is illustrated in Figure 3.5 (c), and show a high level of robustness to outliers, as the correlation

Figure 3.5: Robustness of Error Surface. a) Robustness vs. Sparseness b) Robustness vs. Sensor Noise c) Robustness vs. Outliers

coefficient declined only from 97% to 90% when the outliers changed from 0% to 100%. The correlation coefficient is still near to 81% when outliers are at the 200% level, where there are twice as many outliers as true data points.

3.5 Extraction of Embeddings

The PWSE algorithm is motivated by the observation that each unique view P_i of an object will result in a distinctive error surface $\mathcal{S}_i$, with respect to a model $\mathcal{M}$ in a fixed canonical pose, and these error surfaces also show a certain degree of robustness against data sparseness, sensor noise and outliers. The essence of the method, therefore, is to precalculate and store representations of the $\mathcal{S}_i$ for all views of an object in a preprocessing stage, and then at runtime to compare the error surface of the acquired image against this database.

The error surface is 7-D, so it would be expensive to store and process a rich representation of $\mathcal{S}_i$, especially as there are a large number of $\mathcal{S}_i$ per object (one for each of the 2,160 views). In fact, the computation of the full error surface is unnecessary. As an alternative, we represent each $\mathcal{S}_i$ by a small set of pose values of its minima in some neighborhood of the origin of Θ . In preprocessing, the rotation space is quantized into $N = 2,160$ discrete rotation vectors $\{\vec{r}_i\}_{i=1}^N$, and a set of N views $\{P_i\}_{i=1}^N$ of the object are generated. For each P_i , the closest local minimum $\vec{\theta}_i^c$ to its centroid is first calculated by executing ICP from its centroid, and the translational component t_i^c of $\vec{\theta}_i^c$ is used as the origin of the local coordinate system. Here, the centroid is the geometric center of P_i , which is calculated by averaging all points of P_i . Each P_i is then perturbed to a standard set of K initial poses $\{\vec{\theta}_j^o\}_{j=1}^K$ around the calculated origin.

In our implementation, we have found a set of size $K = 30$ purely translational perturbations to be effective for a database of 60 objects. For the PSB database, K was set to 60 in order to deal with the large number of objects. The perturbations are chosen to be distributed uniformly in the translational subspace of Θ . For each translational dimension, the magnitude of the perturbation ranges from $-r_M$ to r_M with increments of $\Delta r = r_M/2$, which results in a total of $5^3 = 125$ 3-D perturbation vectors. Here r_M represents the maximum radius of the 3-D model, i.e. the furthest distance from the centroid of $\mathcal{M}$ to any point on its surface. A large Δr is preferred as it enlarges the distances among the perturbations and will result in more discriminative feature vectors. To deal with a larger database, K would need to be increased accordingly in order to improve the discriminative power of the feature set, which will, however, also decrease the efficiency of the runtime algorithm linearly.

After applying the perturbations, ICP is allowed to execute for a small number of iterations from each new initial state, resulting in K final pose values $\mathbf{E}_i = \{\vec{\theta}_j^i\}_{j=1}^K$ at the minima of the error surface. Each set $\mathbf{E}_i$ of K minima is called an *embedding* [26] of the error surface $\mathcal{S}_i$. In mathematics, an embedding is a representation of a topological object, manifold, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved. More specifically, an embedding is defined as a finite and small set of samples of a continuous error surface in this paper, which is used to represent and characterize the error surface in a compact format. As distances between embeddings approximate distances between error surfaces, similarity measurement of error surfaces can be computed using embeddings, and that searching with embeddings is more efficient than with original error surfaces.

Chapter 4

Pose Determination

4.1 Problem Definition

The goal of the pose determination (or pose estimation) is to find 3-D translations and orientations of the object that appears in an image, with respect to a known 3-D model at an arbitrary canonical pose. As the 3-D model of the object is known, the object can be represented as a set of views as defined in Section 3.3, and the problem of pose determination can be defined as one of view matching.

Let there be N views $\{P_1, P_2, \dots, P_N\}$ for the known object. As defined in Section 3.3, each view corresponds to a rotational vector in the rotational subspace of Θ . Given an arbitrary view P_r of the known object, the problem of pose determination can be solved by finding the closest match i_m of P_r .

$$i_m = \underset{i \in [1, N]}{\operatorname{argmin}} G(P_r, P_i) \quad (4.1)$$

where $G(\cdot)$ is a similarity measurement function.

4.2 Solution Approach

PWSE provides a way to represent views in a compact form, and to find the closest matching view in a robust and efficient manner. To do so, the process described in Section 3.5 is repeated for image data $\mathbf{P}$ at runtime. A local minimum $\vec{\theta}_p^c$ is first obtained by executing ICP from the centroid of $\mathbf{P}$. The image $\mathbf{P}$ is then translated by the translational term $\vec{t}_p^c$ of $\vec{\theta}_p^c$ so that this local minimum lies at the origin. It is further transformed to each of the K perturbations $\vec{\theta}_j^o$, $j = 1 \dots K$, from which ICP is invoked resulting in an embedding $\mathbf{E}_p$ of final pose values. $\mathbf{E}_p$ is then compared against the N embeddings $\mathbf{E}_i$ that were generated in preprocessing, by simply calculating the similarity, such as the minimum distance, between the embeddings. If we let $\vec{\theta} = (x, y, z, \theta, \phi, \psi)$, then the similarity between two poses $\vec{\theta}_a$ and $\vec{\theta}_b$ is calculated as:

$$\begin{aligned} f(\vec{\theta}_a, \vec{\theta}_b) &= \frac{1}{|D|}(|x_a - x_b| + |y_a - y_b| + |z_a - z_b|) \\ &\quad + \frac{1}{|360^\circ|}(|\theta_a - \theta_b| + |\phi_a - \phi_b| + |\psi_a - \psi_b|) \end{aligned} \quad (4.2)$$

where D is the magnitude of the translational pose perturbation. Two embeddings can be compared by summing the similarities over their corresponding pose sets:

$$g(\mathbf{E}_p, \mathbf{E}_i) = \sum_{j=1}^K f(\vec{\theta}_j^p, \vec{\theta}_j^i) \quad (4.3)$$

The view that most closely matches the current image is identified by summing the similarities of all corresponding poses in an embedding, and taking the minimum:

$$i_m = \underset{i \in [1, N]}{\operatorname{argmin}} g(\mathbf{E}_p, \mathbf{E}_i) \quad (4.4)$$

The final pose estimate can be then calculated as:

$$\vec{\theta}_{i_m} = (\vec{R}_{i_m}, \vec{T}_{i_m}) = (\vec{r}_{i_m}, \vec{t}_p^c + \vec{t}_{i_m}^c) \quad (4.5)$$

where $\vec{t}_{im}^c$ is the preprocessed translational component of $\vec{\theta}_{im}^c$.

Using this procedure, there may exist a few solutions that have the same or very close similarity measures. One way to handle this occurrence is to treat these solutions as multiple hypotheses. The correct pose estimate can then be verified by transforming $\mathbf{P}$ to the model frame, and the transformation that results in the smallest registration error is taken as the solution. Registration error is, however, not very effective in practice for finding the correct pose estimate because $\mathbf{P}$ can be only transformed near to the local minima due to quantization error. A smaller $\Delta\vec{\beta}$ magnitude might reduce this effect, but it will also serve to increase N exponentially.

For this reason, the Bounded Hough Transform (BHT) [24, 60] is used in the verification step. Each hypothesis acts as an initial pose estimate of the BHT, and the pose is transformed towards the local minimum by utilizing the BHT to perform one step tracking. Each BHT procedure results in a peak in the parameter space, and the peak with the largest value signifies the best hypothesis. A block diagram of the complete algorithm is illustrated in Figure 4.1.

4.3 Experimental Results

A set of experiments was conducted on both simulated and real range data to verify the concept as well as to evaluate the robustness and efficiency of the implementation. All experiments were executed on a Pentium 4, 3.2 GHz with 1,024 Mb of RAM running Windows XP. The algorithm was implemented in pure C++ with no assembly level coding. No hardware acceleration or software optimization (other than at the compiler flag level) were applied.

The ICP used in the implementation utilized a point-to-point error metric and

Figure 4.1: Algorithm Diagram

employed a k-d tree to determine the point correspondences. The underlying 3-D model was a point cloud consisting of $\sim 4,000$ points, which was obtained in preprocessing by randomly sampling a surface model of the object.

4.3.1 Simulated Data

The algorithm was first tested on simulated data to verify its correctness, as well as to select the optimal values for the maximum number of ICP iterations. In addition,

Figure 4.2: Test Objects

the robustness of the algorithm with respect to data sparseness, sensor noise, and outliers was also evaluated. A Radarsat satellite model and a model of a biological molecule, illustrated in Figure 4.2, were both tested.

The quantization vector was set to $\Delta\vec{\beta} = \{20, 20, 30\}$ degrees so that the rotational subspace of Θ was quantized into $18 \times 10 \times 12 = 2,160$ discrete rotation vectors. The larger value of 30 was used to quantize the rotation around the Z-axis, which does not exhibit self-occlusion. For each discrete rotation vector, a simulated range image

was generated by sampling the surface of the model in a given pose from the sensor vantage point. Self-occluded data were filtered out, so that the images were 2.5-D, as are typically acquired by conventional range sensors. A total of 2,160 simulated range images were generated in preprocessing to construct the set of $\mathbf{S}_i$.

For testing, a total of 6,000 simulated range images were generated by applying a random rotation vector $\{\theta, \phi, \psi\}$ to the object's canonical pose, where $\theta \in [0^\circ, 360^\circ]$, $\phi \in [0^\circ, 180^\circ]$, and $\psi \in [0^\circ, 360^\circ]$. As we were particularly interested in evaluating the performance of the algorithm on sparse range data, a total of 500 points were randomly sampled for each frame, and the tests were conducted on these sparse range images. For each test image, the pose estimate was calculated using the proposed algorithm, and the result was compared against the ground truth. When the pose error fell within the desired tracking precision of 10 degrees, which is generally sufficient to initiate the pose following algorithms, the trial was deemed to have been successful.

Correctness vs. Number of Maximum Iterations

The first experiment was conducted to decide an optimal value for the maximum number of ICP iterations. A total of six perturbations were applied, with the magnitude of $\pm r_M$ along each dimension. The maximum number of ICP iterations was varied for each trial to values of 5, 10, 20, 30, 50, and 100 so that the correctness vs. number of maximum iterations could be evaluated. The results of this test are plotted in Figure 4.3 and tabulated in Table 4.1.

As shown in the figure, the highest correctness rate was obtained by setting the

	<i>Num. of Suc.</i>	<i>Rate %</i>	<i>Fps</i>
5 iterations	5617	93.6	7.7
10 iterations	5614	93.6	5.8
20 iterations	5566	92.8	5.1
30 iterations	5583	93.0	4.6
50 iterations	5566	92.8	4.2
100 iterations	5566	92.8	3.8

Table 4.1: Correctness vs. Number of Iterations For the Satellite

maximum number of iterations to 5: more iterations served to slightly lower the correctness rate instead of improving it. The experimental results indicate that using approximate locations of the local minima can provide better performance than using accurate locations of the local minima. Views in the PWSE algorithm are discrete representation of an object and each training view actually represents an infinite set of continuous views of the object due to the effect of quantization. The accurate locations of the local minima can, in fact, only provide the best result for one particular view, but not for the set of infinite set of continuous views it represents. It is well known that ICP is a nonlinear optimization algorithm that converges to a local minimum quickly for the first few iterations, and then more slowly as it approaches the local minimum. Because ICP is used in PWSE only for the feature extractions, high registration accuracy is unnecessary. As long as these locations provide enough discriminating power for P_i , a good pose estimate can be achieved. This is a very important property of PWSE, which can be used to improve the efficiency of the PWSE algorithm. Based on this test, we chose 5 as a good value for the maximum number of iterations, and used this value for other experimental tests.

For the satellite model, most failed cases can be divided into two categories. One

Figure 4.3: Correctness Vs. # of Iterations, Satellite

is flip, as illustrated in Figure 4.4(a). Flip results from symmetries in the geometry of the satellite, and can be corrected by an extra verification (i.e., adding 180 degrees perturbation in ψ). Another category of failure occurs when the object is in some “difficult” pose with respect to the sensor, such that the pose cannot be resolved based on a single view of the object. A failure of this kind is illustrated in Figure 4.4(b). Only the small side facet of the satellite is visible, and its other parts are obscured by self-occlusion. Any available temporal information has to be considered in such a case to correct this kind of failure.

The algorithm was further tested on a model of a biological molecule. The same experimental procedure as was applied to the satellite object was reapplied on the

(a) Failure, flip

(b) Failure, difficult pose

Figure 4.4: Failed Cases, Satellite

molecule data set. The results are tabulated in Table 4.2, and illustrated in Figure 4.5. A successful pose estimate is illustrated in Figure 4.6.

The results obtained were consistent with those from the satellite tests, except slightly better correctness rates were achieved with the molecule data set. The reason for this better result is likely due to the geometry of the molecule model, whose error function ξ contains more distinct local minima than that of the satellite model, and these local minima are able to provide a higher discriminative power for pose determination.

	<i>Num. of Suc.</i>	<i>Rate %</i>	<i>Fps</i>
5 iterations	5917	98.6	4.3
10 iterations	5834	97.2	3.1
20 iterations	5736	95.6	2.5
30 iterations	5696	94.9	2.1
50 iterations	5670	94.5	1.7
100 iterations	5584	93.1	1.5

Table 4.2: Correctness vs. Number of Iterations For the Molecule

Figure 4.5: Correctness Vs. # of Iterations, Molecule

Figure 4.6: Pose estimate, Molecule

Sparseness, Sensor Noise and Outliers

To characterize the robustness of the algorithm, a set of new data was generated by degrading the previous data with data sparseness, additive noise and outliers. The method was then executed on this new data set and robustness was evaluated with respect to sparseness, sensor noise and outliers.

The satellite was tested first. The evaluation of robustness vs. sparseness was achieved by randomly sampling 1000, 500, 250, 125 and 75 points from each testing frame respectively, and then executing the pose determination on each sparse data set. The results of this experiment, shown in Figure 4.7(a), demonstrate that the proposed algorithm is extremely robust to sparseness. As the number of points per frame reduced from 1,000 to 500, the correctness rate varied less than one percent, from 94.7% to 93.9%. When using only 75 points per frame, the algorithm still

achieved an 86.7% correctness rate with a speed of 30 fps.

To evaluate robustness vs. sensor noise, the data quality was degraded with simulated Gaussian noise. Random, zero-mean additive Gaussian noise, with a standard deviation of between 0 mm and 200 mm (20% of r_M), was added to each image point. The results are illustrated in Figure 4.7(b), and show that the proposed algorithm is able to function well with simulated sensor noise. The rate of correct pose estimation declined fairly slowly when the sensor noise was less than 100 mm. Since the BHT was used for the verification, once the noise level exceeded 100 mm, which is larger than the voxel space resolution used in the BHT, the performance of the proposed algorithm degraded more rapidly.

Robustness vs. outliers was also tested. In this experiment, a total of 500 points were first randomly sampled from each frame, and spurious data points (i.e., *outliers*) were then randomly added to each data set. The outliers were generated to lie within the bounding box of each data set, with the number of outliers varying between 0% and 20% of the number of data points in each frame. The results are shown in Figure 4.7(c). The experimental results show that the proposed algorithm is fairly robust to outliers. When the number of outliers increased to 20% of the total number of points, the correctness rate degraded only by about 10%.

The same experimental tests were conducted on the molecule model, and the results are plotted in Figure 4.8. One result frame is also shown in Figure 4.9, in which the simulated range image was degraded by adding 20% outliers. The results are consistent with those from the satellite tests, which indicates that the algorithm is general and can function well with freeform objects.

(a) Robustness vs. data sparseness

(b) Robustness vs. noise

(c) Robustness vs. outliers

Figure 4.7: Results of Simulated Data, Satellite

(a) Robustness vs. data sparseness

(b) Robustness vs. noise

(c) Robustness vs. outliers

Figure 4.8: Results of Simulated Data, Molecule

Figure 4.9: Pose Estimate, Outliers

4.3.2 Real Data

Satellite Data

The algorithm was tested with a variety of real range data including:

- 20 frames, collected by an Optech RLS (Rendezvous Laser Vision) Lidar sensor;
- 21 frames, collected by an Optech ILRIS Lidar sensor;
- 10 frames, collected by a Point Grey *BumbleBeeTM* stereovision camera.

Figure 4.10: Real Data, Satellite

Since we intended to evaluate the algorithm on sparse range data, a new set of test images was generated by randomly sampling 500 points for each frame. The results are tabulated in Table 4.3 and a few frames are illustrated in Figure 4.10.

It is worth noting that each of the three sensors have different types of sensor noise. For the RLS and the ILRIS Lidar sensor, a fair amount of data dropped out from the satellite's solar panel. For the *BumbleBee*TM sensor, the tips of the solar panels are bent along the Z-axis, likely due to lens distortions. Despite these data-level errors, the proposed algorithm demonstrates the ability to accommodate these different kinds of sensor noise and provides good pose estimates. Indeed, the poses

	<i>Num. of Frames</i>	<i>Num. of Suc.</i>	<i>Rate%</i>	<i>Fps</i>
RLS	20	20	100	6.6
ILRIS	21	21	100	7.2
<i>BumbleBeeTM</i>	10	9	90	5.7

Table 4.3: Real Data, Satellite

were estimated with 100% correctness from the two Lidar sensors, and only one of the poses was incorrect for the stereovision sensor data.

	<i>Num. of Frames</i>	<i>Num. of Suc.</i>	<i>Rate%</i>	<i>Fps</i>
Chef	22	18	81.8	3.3
Parasaurolophus	16	14	87.5	2.7
T-rex	21	17	81.0	3.29
Chicken	16	15	93.8	4.6

Table 4.4: Real Data, Freeform Objects

Freeform objects

The four freeform objects from [44] shown in Figure 4.2 were also tested, namely a chef model, a parasaurolophus model, a T-rex model and a chicken model. For each frame, a total of 500 points were randomly sampled from the original range images. The results are tabulated in Table 4.4, and four result frames are shown in Figure 4.11. The experimental results show that the method functions well using sparse data for different types of freeform objects. The success rates varied from 80% to 94%, with an efficiency of 2.7 to 4.6 fps.

Figure 4.11: Real Data, Freeform Objects

4.4 Summary and Discussion

In this chapter, we have presented a novel model-based pose determination method for estimating the pose of an object in sparse range data. Instead of avoiding local minima, the algorithm takes advantage of these local minima and uses them as effective feature vectors. we found that the object shape has an impact on the convergence speed of iterative closest point (ICP) algorithm, and by using this property as well, the efficiency and robustness of proposed algorithm can be further improved.

The proposed method is both efficient and robust to data sparseness, outliers, and noise. Only a small number of iterations is needed for each ICP in both preprocessing and runtime, and the performance of the proposed algorithm is close to realtime on

standard hardware using a general ICP implementation. It can be further improved by utilizing more efficient ICP variations, which will be presented in Chapter 5.

Another attractive aspect of the technique is that it functions well in very sparse data. The pose can be successfully recovered from a sparse range image comprising only hundreds of points per frame. Experimental tests have been conducted on both simulated and real data with a variety of models including freeform objects, and results show that the proposed algorithm is effective and robust to data sparseness, sensor noise and outliers.

Chapter 5

Object Recognition

5.1 Problem Definition

The problem of object recognition can be regarded as a natural extension of the problem of pose determination. Let there be M objects, and each object has N views so that the full database of views then consists of $M \times N$ views $\{P_1, P_2, \dots, P_{M \times N}\}$. Then, the problem of object recognition can be solved within the same framework as the problem of pose determination, that is, finding the closest matching view in the database of an input view.

5.2 Solution Approach

It would be inefficient to directly apply PWSE to the problem of object recognition mainly because of the large number of views. The number of views increases linearly with the number of objects, and the cost of matching increases with the square of the number of objects. For instance, a total number of 2,160,000 views have to be

preprocessed and compared against the input view at runtime in order to recognize 1,000 possible objects, which is very computationally expensive.

To address this difficulty, the concept of the *generic model* was introduced to improve the efficiency of computing embeddings. Instead of generating the embedding by convolving the views of an object with a model of itself, all views are instead convolved with a single generic model. At runtime, only a single embedding of the error surface of the image is then required to be calculated and compared against the database. This reduces the cost of matching to be linear, rather than the square of the number of objects. In addition, a *prefiltering* technique was developed to reduce the cost of computing of the distances between runtime and prestored embeddings. The method is based on a voting technique, which is designed specifically for the embeddings used in PWSE, and is more efficient than the techniques based on partitioning the search space such as k-d tree [7] and projection [47].

5.2.1 Generic Model

In pose determination applications, it is assumed that there is a single object under consideration, and a reasonable choice for model $\mathcal{M}$ is a representation of that object. Alternately, in object recognition there may be multiple objects under consideration, and it is not known a priori which object is being imaged in the scene. The straightforward application of the method to object recognition would be to build the database using multiple models, and then at runtime to calculate embeddings of the image with each of these models and compare these embeddings to the database. This would be expensive, however, as not only would the database grow linearly with the number of models, but so too would the number of embeddings required to be

computed at runtime.

The use of a generic model is introduced to tackle the above the problem. In preprocessing, for each object, rather than generating the embedding by convolving the views of the object with a model of itself, they are instead convolved with a single generic model. At runtime, only a single embedding of the error surface of the image is then required to be calculated and compared against the database. For a generic model to be useful, it must generate error surface embeddings that are sufficiently distinctive so as to enable the discrimination of a number of objects. The utility of a given generic model at recognizing a collection of objects would depend upon the relative geometries between the objects in the collection, as well as that of the generic model itself.

Ideally, it would be desirable to automatically generate a generic model that is maximally discriminating for a given object collection, which is the subject of future research. Practically, we have found that it is not difficult to construct an effective generic model. The generic model illustrated in Figure 5.1 comprising 120 spheres was automatically randomly generated, and was found to effectively recognize and determine the pose of the collection of 60 objects shown in Figure 5.2, and the collection of nine objects shown in Figure 5.3 and Figure 5.4. The spheres are distributed randomly within a bounding box of size $D_b \times D_b \times D_b$, and their radii are randomly generated within the range from $D_b/10$ to $D_b/4$.

The dimension of the generic model is not necessarily similar to the dimensions of objects. In fact, a larger generic model is preferred (i.e., $D_b = 2 \times r_M$). This is because when applying the perturbations $\{\vec{\theta}_j^o\}_{j=1}^K$ to the scene image, it is preferred that the translated scene image is still located at (or near to) the generic model such

Figure 5.1: Generic Model

that most points in the scene image are able to find their corresponding closest points within a small range. In addition, using a large generic model allows larger distances between the perturbations $\{\vec{\theta}_j^o\}_{j=1}^K$, which make the feature vectors more distinctive. In experiments, we observed the algorithm to be fairly tolerant to the use of different generic models. As long as the complexity of the generic model exceeds a certain degree (e.g, number of spheres is large enough), the differences among the results using different generic models were minor.

Figure 5.2: Test Objects, Simulated Data

Figure 5.3: Successful Cases, Real Data

5.2.2 Prefiltering for Performance Improvement

Computing the distances between runtime and prestored features, as in Equation 4.2, is the most time-consuming step in the algorithm. For a database of 60 objects and $K = 30$, Equation 4.2 needs to be calculated a total of $2160 \times 60 \times 30 = 3,888,000$ times per frame. To accelerate the nearest neighbor search, an efficient voting technique has been developed specifically for the feature vectors used in PWSE. For these feature vectors, it is more efficient than the techniques based on partitioning the search space, such as k-d tree [7] and projection [47].

The technique is based on the fact that the embeddings $\mathbf{E}_i = \{\vec{\theta}_j^i\}_{j=1}^K$ used in PWSE are constructed by combining locations of K independent local minima together. To find the closest match to the runtime embedding, it is required only to

Figure 5.4: Successful Cases, Real Data

search among a small portion of prestored embeddings that have the majority of their local minima close to the corresponding local minima of the runtime embedding. In practice, we found that the results are sufficiently good by just using the translational component of each local minimum, which makes the technique even more efficient.

The translational subspace of Θ is first quantized into discrete grids using a quantization vector $(D_d/15, D_d/15, D_d/15)$, and then a total of K hash tables are built for each of the K local minima in a preprocessing step. In each hash table, the indices of the prestored embeddings share the same entry only if their K^{th} local minima are located within the same grid in the translational subspace of Θ . At runtime, the hashcodes for each of the K local minima of the runtime embedding are computed. Then, the prestored local minima that are close to each of the K local minima of the

runtime embeddings can be efficiently retrieved using the precalculated hash tables. Each hash table retrieval returns indices of a subset of prestored embeddings, and a total of K retrieved subsets vote on the possible solutions of the closest match of the runtime embedding.

Each prestored embedding will receive K votes at the most, and only the embeddings that receive votes exceeding a certain threshold will be used for the distance calculation. Efficiency and accuracy are best balanced by setting the threshold to about $0.5 \times K$. A larger threshold will increase the efficiency of the algorithm because more prestored feature vectors will be filtered out. However, the accuracy of the algorithm will be slightly decreased, and vice versa.

5.3 Experimental Results

A set of experiments was conducted on both real and simulated range image data to assess the effectiveness, robustness and efficiency of the proposed algorithm. All experiments were executed on a x86_64 quad-cores processor with 8GB of memory running Windows XP. In order to take advantage of the multi-core capabilities of the CPU, the code was parallelized using the Open Multi-Processing (openMP) API [52]. Other than this, no hardware acceleration or software optimization was applied.

The implementation of ICP used an efficient ICP variant [1] that uses a Euclidean distance map (i.e. a voxel map) to accelerate the nearest neighbour computation. The maximum number of iterations was set to 3, which was used in both preprocessing and runtime. The generic model illustrated in Figure 5.1 was used for all tests. The generic model was in point cloud format consisting of 4,000 points, which was obtained in preprocessing by randomly sampling a surface model of the object.

5.3.1 Real Data Tests

The first set of experiments was designed to evaluate the effectiveness of the method using real range data acquired from scans of a database of objects. The four freeform objects illustrated in Figure 5.4 and a set of associated range images were downloaded from [44] and used in these experiments. A total of 21 range images of the chef, 15 range images of the parasaurolophus, 20 range images of the T-rex, and 15 range images of the chicken were tested. For each of the 71 test images, a total of 1,000 points were randomly sampled from the original dense range images, and the experiments were conducted on this new sparse data set.

The result of each test image was first saved as a VRML file and then checked manually, as the ground truth was unavailable. When the object identity and pose estimate were correctly recovered, the trial was deemed successful. The resulting confusion matrix is shown in Table 5.1. Overall, 98.6% of the objects were correctly recognized, and the rate of correct pose recovery was 85.9%. Four successful test images, overlayed on the objects, are shown in Figure 5.4. The lower success rate of pose estimation, compared to object recognition, indicates that in 12.7% of tests, the objects were correctly identified but the recovered poses were incorrect. This was mainly caused by the cylinder-like shape of the chef model, as there were four images of the chef where the model was correctly identified, but the recovered pose was incorrect. In contrast, for the parasaurolophus, T-Rex, and chicken, the number of incorrectly recovered poses were 1, 3, and 1 respectively.

The same test was also conducted on another five freeform objects. Five successful test images, overlayed on the objects, are shown in Figure 5.3. The test images were collected by using a Konica-Minolta VIVID 3-D scanner. A total of 100 images, 20 for

Object	Chef	Para	T-Rex	Chicken
Chef	21	0	0	0
Para	0	14	1	0
T-Rex	0	0	20	0
Chicken	0	0	0	15
OR(%)	100	93	100	100
PD(%)	81	87	85	93

Table 5.1: Confusion Matrix, Real Range Data. OR = average successful object recognition rate, PD = average successful pose determination rate.

Object	Angel	Big Bird	Gnome	Watermelon Kid	Zoe
Angel	19	0	1	0	0
Big Bird	1	18	1	0	0
Gnome	0	1	18	1	0
Watermelon Kid	1	0	0	19	0
Zoe	0	0	0	0	20
OR(%)	95	90	90	95	100
PD(%)	80	85	85	90	95

Table 5.2: Confusion Matrix, Real Range Data

each object were used for recognition. The confusion matrix of these 100 recognition trials is reported in Table 5.2. The overall recognition rate was 94%, and the rate of correct pose recovery was 87%.

5.3.2 Simulated Data Tests

The above real data tests were encouraging, and we desired to experiment on a larger database of objects. Unfortunately, no such larger database of real range images and associated object models was available. As an alternative, we performed a set of experiments on a set of simulated range images of the 60 free-form objects shown in Figure 5.2. The use of simulated data allowed for the systematic evaluation of

the performance under varying degrees of data sparseness, measurement error and outliers. All object models were scaled to reside within a bounding box of size $1 \times 1 \times 1$, and the training and test images were both obtained by rendering these models into a depth buffer using OpenGL. A total of 30 pose perturbations were applied ($K = 30$), which were chosen from 125 3-D perturbation vectors generated as described in Chapter 4.

The rotational subspace of Θ was first quantized into $N = 2,160$ discrete rotation vectors $\{(\theta_i, \phi_i, \psi_i)\}_{i=1}^N$. For each object, a total of 2,160 range images were generated by transforming each model by $(\theta_i, \phi_i, \psi_i)$. Simulated range images (i.e., views) were then generated by sampling the surface of the model in its given pose from the sensor vantage point. Self-occluded data were filtered out such that the resulting views were 2.5-D. These images were used to construct the set of views $\{P_i\}_{i=1}^N$ in preprocessing to index the objects and their poses.

The test image set contained 12,000 images, 200 from each of the 60 objects. For each object, simulated range images were generated by applying a random rotation vector (θ, ϕ, ψ) to the object's canonical pose, where $\theta \in [0^\circ, 360^\circ]$, $\phi \in [0^\circ, 180^\circ]$, and $\psi \in [0^\circ, 360^\circ]$. As we were particularly interested in evaluating the performance of the algorithm on sparse range data, a total of 1,000 points were randomly sampled for each test image, and the tests were conducted on these sparse range images. For each test image, the object ID and pose estimate of the object were calculated using the proposed algorithm, and the result was compared against the ground truth. When the object was correctly recognized and the error of the pose estimate fell within the desired precision of 10 degrees, the trial was deemed to have been successful.

Ideal Data Tests

An experiment using ideal (i.e., non-noisy) range data was conducted to prove the effectiveness of the proposed algorithm. The resulting confusion matrix is illustrated in Figure 5.5. The experimental result shows that the algorithm was able to correctly recognize over 98% of the objects, and about 97% of the poses were correctly recovered. In addition, PWSE is extremely efficient, running at about 122 fps for a database of 60 objects. The worst case is the sports car (No. 41), whose pose determination rate and object recognition rate were 74.5% and 82.5% respectively. After examining the unsuccessful cases, we found that it was recognized as the jeep (No. 56, 25 cases, and No. 57, 4 cases), and the tank (No. 58, 2 cases).

Non-ideal Data Tests: Sparseness, Noise, and Outliers

A set of experiments was performed to study the robustness of the proposed method with respect to data sparseness, noise (i.e., measurement error) and outliers. To measure the influence of data sparseness, a new set of test images was generated by randomly sampling 1,000, 500, 250, 125 and 75 points from each test image from the ideal data tests. The same experiment as in the ideal data tests was then executed on this data set, and the results are shown in Figure 5.6 (a). It can be seen that the method is robust to data sparseness as the rate of correct recognition varied by only about 0.3% when the number of points per image was reduced from 1,000 to 500. When using only 125 points per image, the algorithm still achieved a 92.6% correctness rate, with an increased efficiency of over 200 fps.

Figure 5.5: Confusion Matrix, Simulated Data

The evaluation of robustness vs. measurement error was achieved by first randomly sampling 1,000 points from each test image from the ideal data test, and then adding Gaussian noise to each point. A Gaussian noise model was used in the tests as is conventional when the exact distribution is unknown. The noise was zero mean and the standard deviation varied between 0% and 20% of the size of an image's bounding box. The results are illustrated in Figure 5.6 (b), and show that the proposed algorithm has a good level of robustness to Gaussian noise below $\sim 10\%$. The

a) Sparsness

b) Noise

c) Outliers

Figure 5.6: Experimental Results, Non-ideal Data Tests. a) Robustness vs. Sparsness b) Robustness vs. Sensor Noise c) Robustness vs. Outliers

rate of correct pose estimation declined fairly slowly when the sensor noise was less than 10%.

A final test was executed to determine the impact of outliers. A total of 1,000 points were first randomly sampled for each image of the ideal data test. Spurious data points (i.e., *outliers*) were then randomly inserted into each test image. The outliers were generated to lie near to surface points, (i.e., range from 10% to 30% of the size of an image's bounding box), with the number of outliers varying between 0% and 200% of the number of points in each image. The results are illustrated in Figure 5.6 (c). The experimental results show a high level of robustness to outliers, as the rate of correct classification declined only from 97% to 90% when the outliers changed from 0% to 100%. The rate of correct classification is still near to 81% when outliers are at the 200% level, where there are twice as many outliers as true data points.

Impact of Different Generic Models

In order to investigate the impact of different generic models on performance, five generic models were generated. These generic models were completely different, with each generated independently by using 30, 60, 90, 120 and 150 spheres. Each generic model contained a total of 4,000 points which were randomly sampled from each surface model of the generic models. The same ideal data tests were then re-executed, using different generic models, and the results are shown in Figure 5.7.

The results show that when the number of spheres exceeded 60, the differences among the results using different generic models were minor. In fact, the recognition rate only varied within one percent when the number of spheres changed from 60 to

Figure 5.7: Impact of Different Generic Models

150, which shows PWSE to be fairly tolerant to the use of different generic models.

5.3.3 Experiments with the Princeton Shape Benchmark

To further investigate the performance of the method on a large data set, a set of experiments were conducted on the 3-D models in the PSB database [61]. The PSB database provides a set of 1814 classified 3-D models collected from the Web. The 3-D models were divided into the two sets: the training data set containing 907 models partitioned into 90 classes and the testing data set containing 907 different models partitioned into 92 classes. The PWSE algorithm was tested on both sets. For each data set, a total of 18140 test images, 20 for each of the 907 objects, were generated

using the same method as the simulated data tests, and the same generic model was used in the PSB experiments.

Object Recognition and Impact of K

The first set of tests was designed to decide the optimal value of K , in which PWSE was executed on the two PSB data sets with K varied from 30 to 120, in increments of 10. The results are illustrated in Figure 5.8. When $K = 30$, the recognition rate of PWSE was less than 2% for both data sets. However, this rate increased rapidly with the value of K , and the recognition rates peaked when $K = 60$ for both data sets. Further increasing K did not improve the recognition rate, which slightly decreased when K increased from 60 to 120. The results from the two different data sets were consistent, which indicates that the recognition rate mainly depends on K and the size of database, and that the shape variations of each individual model had only a minor effect on the overall performance of the method.

It is worth noting that the PSB was designed for the problem of class recognition, which contains many similar objects. For instance, it has 50 fighter_jet models, 50 human models, 17 head models, etc. In the above experiment, each test image consisted of only 1,000 points, so that the fine details of the models were often lost due to the sampling effect. Therefore, there are many cases where PWSE misrecognized the objects within the classes. In addition, the PSB also contains many symmetric objects (i.e., swords, tools, hourglass and etc.), which reduces the accuracy of the pose determination rate.

Figure 5.8: Object Recognition, PSB

5.3.4 Real-time Stereovision Data Tests

To further investigate the applicability of PWSE to challenging tasks of object recognition, we have developed a complete real-time object recognition and tracking application using a stereovision sensor, which is able to recognize and track the 10 objects shown in Figure 5.3 and Figure 5.9. The real-time tracking system was implemented in C++, and range data with subpixel disparity interpolation was acquired with a Point Grey Research *BumbleBeeTM* stereovision camera.

A simple background subtraction technique was used to separate the objects from the static background clutter. Once the number of foreground points in the subtracted image exceeded a certain predefined threshold, the system deemed that an object was

Figure 5.9: Test Objects, Real-time Stereovision

present in the scene and invoked PWSE for object recognition. To improve efficiency, a set of points was randomly selected from the subtracted image (e.g., 1,000 points), and this sparse range image served as input to PWSE. Once the object was recognized by PWSE, the output (i.e., Object ID and initial pose estimate) was fed into a more efficient tracking algorithm, BHT [60]. The results were visualized by rendering the 3-D model of the object with the estimated value of the translation and rotation.

The results were qualitatively evaluated by examining the 3-D model rendered using the estimated pose. The results confirm that a good recognition rate was obtained for all of the five tested models. Five screen shots of selected frames for the different objects being recognized by the system are shown in Figure 5.10. Each frame contains two views. The left view is used to display the intensity video, whereas the white circle is used to indicate the rough location of the tracked object in the current frame. The right view is used to render the 3-D model of the object using the estimated pose.

To the best of our knowledge, this is the first system which is able to perform real-time object recognition using a stereovision camera. The range data collected

Figure 5.10: Recognition Result, Real-time. a) Angel b) Zoe c) Gnome d) Big Bird
e) Watermelon Kid

by stereovision cameras are relatively noisy and have many dropouts due to depth discontinuities. Working with this type of data is extremely challenging for object recognition techniques. This is especially true for local methods such as spin images and point signatures that rely upon an estimate of the normal around each point, which can be very inaccurate. In addition, the objects being recognized were held by hand during the tests, which partially occluded surfaces of the objects and introduced a large number of outliers to the data and made the problem even more difficult. Despite these challenges, PWSE was able to correctly recognize all of the objects in an efficient and robust manner, which further demonstrated the effectiveness and robustness of the PWSE algorithm.

5.4 Summary and Discussion

In the previous sections, the performance of PWSE has been evaluated for both object recognition and object class recognition. From these results, it can be concluded that PWSE compares favourably to the previous object recognition techniques in terms of efficiency and accuracy. In a previous study [57], spin images was tested with a 3-D model database of 56 objects using a 2 GHz PC with 2 GB memory. The reported accuracy of the simulated data tests, which were similar to our ideal data tests, was slightly above 90%, and recognition time per query was 43 seconds on 50 objects. The experiments on real data were conducted with 88 real queries against a 90 model database, and the reported accuracy was $\sim 40\%$.

In more recent work, an experiment conducted on the tensor-based technique [44] was close to our ideal data tests. To the author's knowledge, the tensor-based technique achieved the best accuracy reported to date from among the model-based

approaches. Compared to their work, we used a slightly larger database (60 objects vs. 50 objects), and a much larger number of test images (12,000 images vs. 500 images). PWSE was able to achieve a better result than that reported for the tensor-based technique, at 97% vs. 95%. The time efficiency of the tensor-based experiments was not reported, although it was mentioned that their implementation was not optimized for time as it was developed in Matlab.

Although some objects used in our tests were the same that had been used in [44] to evaluate the feature-based method, more difficult objects were also added to our database, such as trees and flowers that contain many discontinuous surfaces. This kind of object would cause particular difficulties for the tensor-based technique, which requires the building of a surface mesh and calculating the normals of the points. In contrast, PWSE was able to deal with these objects very well because it works directly on 3-D points and doesn't require a mesh or normals. In addition, PWSE was also evaluated on the much larger PSB data set containing 1,814 objects, and achieved promising results.

Chapter 6

Object Class Recognition

6.1 Problem Definition

Instead of identifying an object at the instance level, object class recognition applies a higher level label to a collection of objects. Some training and test models in the PSB database are illustrated in Figure 6.1, which includes two classes: *plane* and *plant*. A class recognition classifier needs to first learn the common features of classes from a set of training models, and then it must assign a hierarchy of semantic labels to a set of testing models, which may not be present in the training sets.

The problem of object class recognition is more challenging than the problem of object recognition due to the inter-class variation, and has been investigated by many researchers [2, 12, 28, 35, 39, 48, 49, 73]. As shown in Figure 6.1, although some objects are similar, no two objects are exactly alike, and some of the models in the testing set are quite different from those in the training set even when drawn from the same class. In this chapter, we apply PWSE to this challenging problem and examine how well PWSE is able to deal with the inter-class variations.

A) Plane Models, Training Set

B) Plant Models, Training Set

C) Plane Models, Testing Set

A) Plant Models, Testing Set

Figure 6.1: Example of Object Class Recognition

6.2 Solution Approach

A straightforward approach for applying PWSE to the problem of object class recognition is to use a single view of the test objects. In preprocessing, the same procedure utilized in object recognition is used to generate the embeddings for each of the training models, where each embedding corresponds to a class label, rather than an object ID and a rotation vector. During runtime, a random 2.5-D view is generated from the complete 3-D model of a test object. The embedding that most closely matched the current view is identified by using the same technique as in object recognition, and its class label is assigned to the current test object.

A single 2.5-D view only provides a partial description of the complete 3-D object, and will therefore bias the performance of PWSE. As has been shown in previous research [27,37,62,68,72], using multiple 2-D images of 3-D objects is able to improve the recognition rate. Based upon this, we hypothesize that the performance of PWSE could be improved further using a multi-view classification scheme. For a given query model, rather than using a single 2.5-D image as the basis for classification, a small number of range images from different views are generated. Each image served as a distinct query image, and produced a nearest neighbor (NN) classification result. The results from all NN classifications of all images are then voted upon, and the object with the majority of votes is taken as the retrieved class.

6.3 Experimental Results

In order to determine the performance of PWSE for the object class recognition problem, a series of experiments were run on the PSB database. The use of the

PSB database allowed us to compare the results of PWSE against the published performance of several other state-of-the-art algorithms. The experimental setup and the generic model were the same as in the previous object recognition tests, and the parameter K was set to 60 for all of the tests based on the results of the previous chapter.

The first test was a straightforward extension of the object recognition tests. Instead of evaluating the results by object ID and their pose estimates, we examined the recognition results in terms of object class ID, which was conveniently included in the description of each object in the PSB. For any given query 2.5-D range image, if the retrieved model was from the same class as the query image, then the query was deemed successful; otherwise, it was deemed unsuccessful. This test is similar to the nearest neighbor experiments presented in [61]. One difference was that, whereas the NN tests were executed only on the 907 objects in the test set, we extended our experiments on both the 907 objects in each of the test and training sets.

For each object, a total of 2,160 simulated views were generated, and used to precalculate the embeddings for the object, which is the same as in object recognition. During runtime, using a single 2.5-D query image that was generated from each object model, PWSE was able to achieve a 90.9% classification rate on the training set, and a 87.8% classification rate on the test set. This significantly outperformed the twelve well-known shape descriptors that were tested in [61]. Of those twelve methods, the most successful was the Light Field Descriptor (LFD) [12], which achieved a 65.7% classification rate on the PSB training data set, 22.1% lower than that achieved by PWSE. It was interesting that this higher classification rate was achieved by PWSE using only 2.5-D range data as input, such as can be acquired by standard range

imaging sensors. In contrast, most of the other 12 methods that were tested in [61] required complete 3-D information about the object, without considering the effects of self-occlusion.

The above single-view recognition tests were promising, although they only utilized partial information of the objects. As discussed in Section 6.2, the performance of PWSE could be improved further using a multi-view classification, and so the above experiments were re-executed using a multi-view scheme. Figure 6.2 shows the results of multi-view classification. The classification rates were improved to 96.4% and 94.3% respectively on the training and test sets when using 5 views. When using 20 views, the classification rates increased further to 98.7% and 96.8% respectively, which was a significant improvement over tests on the same data set using single view PWSE (90.9%) or LFD (65.7%).

A more challenging test was performed, in which the training set was first used in the preprocessing stage to construct the offline embeddings, and the online object class recognition was then performed on the test set. The test set contains 907 different 3-D models partitioned into 92 classes. None of the particular models in the test set exist in the training set, although there is some overlap in the object classes. The results from this experiment are summarized in Table 6.1. For base classification, which is the primary and finest granularity of classes into which the PSB objects were categorized, the classification rate was 26.9%. This is good when considering that 71 of the 92 classes in the test set were not present in the training set, and were therefore outlier classes. If only the 21 classes containing the 370 objects present in both sets were considered, then the classification rate could be reinterpreted as 65.9% as shown in the second row, first column of Table 6.1.

Figure 6.2: Multiview Class Recognition

Interestingly, by considering the misclassified cases, we found that PWSE was good at capturing the overall geometric characteristics of the models. For instance, the outlier class `hot_air_balloons` (model 1337-1345) was misrecognized as `helmet` (model 1637), `snowman` (model 1758), `ice_cream` (model 751, 754, 759 and 760), `potted_plant` (model 1026) and `head` (model 347), which are all sphere-shell like models as illustrated in the first row of Figure 6.3. Similarly misclassified cases include: `flying_saucer` to `round`, `human_arms_out` to `fighter_jet`, `sword` to `flower_with_stem`, `fireplace` to `laptop`, `stealth_bomber` to `handgun`, `book` to `microchip`, `chess_set` to `rectangular` and `double_doors`, `desktop` to `TV`, `computer_monitor` to `roman_building`, `wheel` and `gear` to `tire`, and so on.

#1338	#1638	#1758	#760	#1028	#349

#1381	#924	#927

#241	#1195	#1189	#1183

#693	#988	#1681

#1798	#1761

#1295	#649	#275

#1791	#574	#579

#1590	#894	#1731	#1724

#1768	#563	#571

#552	#413

#733	#738	#927	#1638	#739	
Query	Retrieved models				

Figure 6.3: Class Recognition, Misclassified Models

Data Set	Base (92)	Coarse (44)	Coarser (6)	Coarsest (2)
Complete	26.9%	40.7%	51.6%	80.3%
Overlapping	65.9%	71.1%	77.8%	87.6%

Table 6.1: Classification Rate On Test Set

There are, of course, different ways that models can be classified, such as structural, functional and so on. The PSB recognizes this by offering a variety of different classification schemes, of varying granularity (i.e., coarseness), and the performance of PWSE on these different classification granularities was further evaluated. The “coarse” classification merges the 92 base classes to form 44 classes. The “coarser” classification merges these 44 classes further to form just six classes, and the “coarsest” classification merges these six classes to form just two classes: man-made objects and natural objects. The classification rates of PWSE with respect to these different classifications is listed in Table 6.1. As expected, PWSE performed better on the coarser classifications. This is particularly true when the results of complete non-overlapping data sets are considered, in which case the classification rate increased from 26.9% for the “base” granularity, to 80.3% for the “coarsest” granularity. In contrast, the improvement for the overlapping classes was less significant. This result indicates that PWSE is effective at discriminating between man-made and natural objects, possibly because man-made objects typically comprise primary shapes (e.g., sphere, disk etc.) which PWSE is good at recognizing.

6.4 Summary and Discussion

With little modification, we directly applied the PWSE algorithm to the more difficult problem of object class recognition, and tested it on the well-known PSB data set. The results show that PWSE outperformed LFD by 31.1% in terms of NN classification. PWSE is slightly less time-efficient and consumes more memory per object than LFD, mainly because the current version of PWSE was targeted at object recognition applications, which requires the storage of a large number of embeddings for each view of the objects in order to determine the pose of objects.

Most existing object recognition and object class recognition techniques extract features from local neighborhoods of the models or images, and then compress these local features into a compact and invariant representation. For instance, point signatures use curvature features and compress them into 1-D contours that are invariant to rotations; spin images represent a small surface patch around a point by a 2-D histogram that is invariant to rotations; the tensor-based technique represents a triangular mesh by a third order tensor. In extracting and compressing these features, geometric information is lost. Also, these features require a dense sampling of the surface around each point, and are sensitive to the effects of data sparseness. As a result, the descriptors resulting from these local features may lack discriminating power, which can lead to a high proportion of incorrect point correspondences between the scene and the model. To compensate for incorrect correspondences, robust techniques such as RANSAC or GHT can be used, which while effective, nevertheless increase the runtime expense of these methods.

In contrast, the performance of PWSE mainly derives from the use of highly descriptive and robust features, which are extracted from the 7-D error surface. Each

point on the error surface is formed by computing the RMS error using all of the image points, which makes it convey much more information than a single point in the image data, and therefore makes it very robust to data sparseness, noise and outliers. Like the local methods, PWSE also represents the error surface in a compact embedded form, but it nevertheless still carries more information than features extracted solely from the image domain. For this reason, PWSE features tend to be more discriminating than local features, and don't require additional statistic methods such as RANSAC to filter out mismatches. In addition, the 7-D error surface has a rich landscape, so that more distinctive and discriminating features can be extracted from it. For this reason, PWSE is able to achieve a good classification rate on both freeform objects as well as objects with simple regular geometries, such as sphere, cone and cube, etc.

Chapter 7

Conclusions and Future Work

We have presented a new approach to pose determination, object recognition and object class recognition using range images. The proposed algorithm is purely geometry based: it functions solely on surface geometry, and can therefore function on freeform objects, without requiring the presence of any particular identifying features or patterns. The algorithm has been verified and evaluated extensively using both simulated and real range images, and experimental results show that it is both effective and efficient.

ICP is among the most useful algorithms in range image processing, and many researchers have explored its properties. The existence of local minima in the well space is well-known, and the general strategy has been to avoid converging to a local minimum by ensuring that the initial pose lies within the global minimum well. In contrast, the proposed method explicitly explores the error surface in an attempt to converge to a set of local minima. To our knowledge, this is the first attempt to exploit the existence of local minima, and allows ICP, and potentially other local optimization algorithms used for registration, to be extended to solve the pose determination and

object recognition problems.

The experimental results show the proposed method to be both efficient and robust to data sparseness, outliers, and measurement error. Only a small number (e.g. 3) of iterations is required for each ICP invocation in both preprocessing and runtime. The performance of the proposed algorithm runs at 122 frames per second on standard hardware, and the object recognition and pose determination rates respectively exceed 98% and 97% with a database of 60 objects.

In addition, we have explored ways of applying PWSE to the problem of 3-D object retrieval, and have achieved promising results. These tests were executed on the PSB data set, which contains 1,814 3-D objects, divided into training and test sets of 907 objects each. PWSE was able to achieved a 90.9% classification rate on the training set, and a 87.8% classification rate on the test set using a single 2.5-D query image. By further applying a multi-view classification technique, which used multiple input 2.5-D images per query, we were able to further improve the classification rate. The classification rate was increased to 96.4% and 94.3% respectively on training and test sets using five input images from different views of the query object. It was further increased to 98.7% and 96.8% using 20 input query images. These recognition rates were a significant improvement over previously reported object retrieval methods on this standardized data set.

One limitation of the current PWSE implementation for object recognition applications is that it is a global method and cannot handle background clutter, and thus requires the foreground objects to be initially segmented from the background scene. Without modification, it is difficult to directly apply the proposed algorithm to static scenes, and local techniques may be more suitable for such applications. However,

when the object can be separated from background clutter (e.g., through motion segmentation and in 3-D object retrieval), the proposed algorithm is an attractive alternative to local methods due to its efficiency and robustness to data sparseness, sensor noise and outliers.

There is still room for improvement and many open problems still remain to be solved. In this very final section of the dissertation, we will identify some of these open problems and try to give directions for their solutions. First, we aim to extend the method to handle significant degrees of scene clutter, which in fact is fairly straightforward by borrowing elements of the model-based technique. Rather than calculating the feature vectors for each view of objects, PWSE can be employed to compute the feature vectors for a small patch (supporting region) surrounding basis points. As the embeddings are calculated only for a limited local support region, the modified PWSE will be able to deal with cluttered scenes [29].

In addition, we are investigating the possibility of automatically generating optimal generic models for a given database. Not only is the optimal generic model able to improve the accuracy, efficiency and robustness of proposed algorithm, but it also enables PWSE to deal with a large number of objects. The problem is difficult because the database may contains thousands of objects and each object also has 2,160 views, which is a tremendous search space. The adaptive boosting (AdaBoost) algorithm [20] seems promising for tackle the problem, which can be employed to constructed optimal generic models by combining a set of simple generic models.

A method that can determine the optimal number of views for a given object is also worth exploring. In the current implementation of PWSE, each object is represented by 2,160 views, obtained by evenly quantizing the rotational subspaces of

Θ . For objects with simple geometry and symmetric objects, such a large number of views may be redundant and will reduce the overall efficiency of the algorithm because computational resources are wasted on computing and comparing the embeddings of these redundant views. Conversely, the 2,160 views may not be sufficient for the freeform objects with complicated geometry, and will cause misrecognition. Organizing those views in a similar way as an aspect graph can further improve the performance of PWSE.

The PWSE algorithm is not limited by ICP, and it can therefore be substituted with other optimization techniques. In future work, we are planing to explore possibilities of improving the performance of PWSE by exploring properties of different types of optimization methods. In addition, applying PWSE to 2-D image data will also be

Bibliography

- [1] M. Abraham, P. Jasiobedzki, and M. Umasuthan. Robust 3D vision for autonomous space robotic operations. In *Proceedings of 6th International Symposium of Artificial Intelligence and Robotics in Space*, pages 2235–2241, June 2001.
- [2] C. B. Akgl, B. Sankur, Y. Yemez, and F. Schmitt. 3D model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1117–1133, 2009.
- [3] M. Bondy B. Taati, P. Jasiobedzki, and M. Greenspan. Variable dimensional local shape descriptors for object recognition in range data. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation for Recognition*, pages 1–8, 2007.
- [4] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

- [6] R. Benjemaa and F. Schmitt. Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique. In *Proceedings of International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 113–120, 1997.
- [7] J. L. Bentley. K-d trees for semidynamic point sets. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 187–197, New York, NY, USA, 1990. ACM.
- [8] P.J. Besl and H.D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [9] G. Blais and M. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1975.
- [10] A. Bottino and A. Laurentini. The visual hull of smooth curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1622–1632, Dec. 2004.
- [11] R.J. Campbell and P.J. Flynn. Eigenshapes for 3D object recognition in range data. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 505–510, 1999.
- [12] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. *Computer graphics forum*, 22(3):223–232, 2003.
- [13] H. Chen and B. Bhanu. 3D free-form object recognition in range images using local surface patches. In *Proceedings of International Conference on Pattern*

- Recognition*, volume 3, pages 136–139, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [14] C.S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.
- [15] Christopher M. Cyr and Benjamin B. Kimia. A similarity-based aspect-graph approach to 3D object recognition. *International Journal of Computer Vision*, 57(1):5–22, 2004.
- [16] Christopher M. Cyr and Benjamin B. Kimia. A similarity-based aspect-graph approach to 3D object recognition. *International Journal of Computer Vision*, 57(1):5–22, 2004.
- [17] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, January 1997.
- [18] D.W. Eggert, K.W. Bowyer, C.R. Dyer, H.I. Christensen, and D.B. Goldgof. The scale space aspect graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1114–1130, Nov 1993.
- [19] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [20] Y. Freund and R.E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

- [21] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision*, pages 224–237, May 2004.
- [22] M. Greenspan. Queen’s model database. <http://rcvlab.ece.queensu.ca>.
- [23] M. Greenspan and G. Godin. A nearest neighbor method for efficient ICP. In *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pages 161–170, May 2001.
- [24] M. Greenspan, L. Shang, and P. Jasiobedzki. Efficient tracking with the bounded hough transform. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 520–527, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
- [25] M. Greenspan and M. Yurick. Approximate k-d tree search for efficient ICP. In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 442–228, Oct. 2003.
- [26] G.R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):530–549, 2003.
- [27] D. Hoiem, C. Rother, and J. Winn. 3D layoutcrf for multi-view object class recognition and segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 0, pages 1–8, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

- [28] B.K.P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, Dec. 1984.
- [29] Y. Ioannou, L. Shang, R. Harrap, and M. Greenspan. Local potential well space embedding. In *Proceedings of International Conference on 3D Digital Imaging and Modeling, 2009.*, volume 1, pages 297–304, 2009.
- [30] P. Jasiobedzki, M. Abraham, P. Newhook, and J. Talbot. Model based pose estimation for autonomous operations in space. In *Proceedings of IEEE International Conference on Intelligence, Information, and Systems*, pages 211–215, 1999.
- [31] A.E. Johnson and M. Hebert. Surface registration by matching oriented points. In *Proceedings of International Conference on Recent Advances in 3D Digital Imaging and Modeling*, pages 121–128, 1997.
- [32] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.
- [33] T. Jost. *Fast Geometric Matching for Shape Registration*. PhD thesis, University of Neuchâtel, 2002.
- [34] T. Jost and H. Hugli. A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 427–433, Oct. 2003.

- [35] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of Symposium on Geometry Processing*, June 2003.
- [36] B. Krebs, P. Sieverding, and B. Korn. A fuzzy ICP algorithm for 3D free-form object recognition. In *Proceedings of the 1996 International Conference on Pattern Recognition*, volume 1, page 539, Washington, DC, USA, 1996. IEEE Computer Society.
- [37] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3D object recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 0, pages 1–8, Los Alamitos, CA, USA, 2007. IEEE Computer Society.
- [38] C. Langis, M. Greenspan, and G. Godin. The parallel iterative closest point algorithm. In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 195–204, 2001.
- [39] X. Li, A. Godil, and A. Wagan. Shrec’08 entry: Visual based 3D CAD retrieval using Fourier Mellin transform. In *Proceedings of International Conference on Shape Modeling and Applications*, volume 0, pages 235–236, Los Alamitos, CA, USA, 2008.
- [40] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 2, pages 1150–1157, 1999.

- [41] J. Luck, C. Little, and W. Hoff. Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3739–3744 vol.4, 2000.
- [42] Y. Luo, H.Ma, and F. Li. 3D object recognition technique based on multi-resolution aspect graph. In *Proceedings of International Conference on Neural Networks and Brain, 2005.*, volume 2, pages 1168–1172, Oct. 2005.
- [43] B. Ma and R. E. Ellis. Surface-based registration with a particle filter. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, pages 566–573, 2004.
- [44] A.S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1584–1601, 2006.
- [45] M.H. Moghari and P. Abolmaesumi. Point-based rigid-body registration using an unscented kalman filter. *IEEE Transactions on Medical Imaging*, 26(12):1708–1728, Dec. 2007.
- [46] L. Morency and T. Darrell. Stereo tracking using ICP and normal flow constraint. In *Proceedings of International Conference on Pattern Recognition*, volume 4, pages 367–372, 2002.
- [47] S.A. Nene and S.K. Nayar. A simple algorithm for nearest neighbor search in high dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, 1997.

- [48] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *Proceedings of International Conference on Shape Modeling and Applications*, pages 154–166, May 2001.
- [49] P. Papadakis, I. Pratikakis, S. Perantonis, T. Theoharis, and G. Passalis. Shrec’08 entry: 2D/3D hybrid. In *Proceedings of International Conference on Shape Modeling and Applications*, volume 0, pages 247–248, Los Alamitos, CA, USA, 2008. IEEE Computer Society.
- [50] A.R. Pope and D.G. Lowe. Learning 3D object recognition models from 2d images. In *Proceedings of AAAI Fall Workshop on Machine Learning in Computer Vision*, pages 35–39, 1993.
- [51] Arthur R. Pope and David G. Lowe. Probabilistic models of appearance for 3D object recognition. *International Journal of Computer Vision*, 40(2):149–167, 2000.
- [52] J. Michael Quinn. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Education Group, 2003.
- [53] S. Ruiz-Correa, L.G. Shapiro, and M. Melia. A new signature-based method for efficient 3-d object recognition. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001.*, volume 1, pages I–769–I–776 vol.1, 2001.
- [54] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

- [55] Ruwen S., Roland W., and Reinhard K. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007.
- [56] B. Schiele and J.L. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 2, pages 50–54, Washington, DC, USA, 1996. IEEE Computer Society.
- [57] Y. Shan, B. Matei, H. S. Sawhney, R. Kumar, D. Huber, and M. Hebert. Linear model hashing and batch RANSAC for rapid and accurate object recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 121–128, 2004.
- [58] L. Shang and M. Greenspan. Pose determination by potential well space embedding. In *Proceedings of Sixth International Conference on 3D Digital Imaging and Modeling*, pages 297–304, Aug. 2007.
- [59] L. Shang and M. Greenspan. Real-time object recognition in sparse range images using error surface embedding. *International Journal of Computer Vision*, Aug. 2009.
- [60] L. Shang, M. Greenspan, and P. Jasiobedzki. Model-based tracking by classification in a tiny discrete pose space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):976–989, 2007.
- [61] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Proceedings of International Conference on Shape Modeling and Applications 2004*, volume 0, pages 167–178, 2004.

- [62] S. Silvio and F. Li. View synthesis for recognizing unseen poses of object classes. In *Proceedings of the 10th European Conference on Computer Vision*, pages 602–615, Berlin, Heidelberg, 2008. Springer-Verlag.
- [63] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-D pose estimation using a high-speed range sensor. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2235–2241, May 1994.
- [64] Christopher C. Skiřcim and Bruce L. Golden. Optimization by simulated annealing: A preliminary computational study for the tsp. In *Proceedings of the 15th conference on Winter Simulation*, pages 523–535, Piscataway, NJ, USA, 1983. IEEE Press.
- [65] D. Skocaj and A. Leonardis. Robust recognition and pose determination of 3D objects using range images in eigenspace approach. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 171–178, 2001.
- [66] F. Stein and G. Medioni. Structural indexing: efficient 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.
- [67] Y. Sun, J. Paik, A. Koschan, D.L. Page, and M.A. Abidi. Point fingerprint: a new 3D object representation scheme. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(4):712–717, 2003.

- [68] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. V. Gool. Towards multi-view object class detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1589–1596, Los Alamitos, CA, USA, 2006. IEEE Computer Society.
- [69] T. Van Effelterre, L. Van Gool, and A. Oosterlinck. Visual recognition of CAD objects with aspect graphs. In *Proceedings of the 1992 IEEE International Symposium on Intelligent Control, 1992.*, pages 54–59, Aug 1992.
- [70] S. Weik. Registration of 3-D partial surface models using luminance and depth information. In *Proceedings of International Conference on Recent Advances in 3-D Digital Imaging and Modelling, Ottawa*, pages 93–100, 1997.
- [71] S.M. Yamany and A.A. Farag. Surface signatures: an orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1105–1120, 2002.
- [72] P. Yan, S.M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *Proceedings of IEEE 11th International Conference on Computer Vision, 2007.*, pages 1–6, Oct. 2007.
- [73] T. Yang, B. Liu, and H. Zhang. 3D model retrieval based on exact visual similarity. In *Proceedings of International Conference on Signal Processing*, pages 1556–1560, Oct. 2008.

- [74] D. Zhang. *Harmonic shape images: a three-dimensional free-form surface representation and its applications in surface matching*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2000. Chair-Hebert, Martial.