Automatic 3D Registration using Range and Colour Data

By

Andrew Miles

May 6, 2004

A thesis submitted to the

Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science

School of Computer Science

Carleton University

Ottawa, Ontario

Canada

© Copyright Andrew Miles May 2004

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-93962-6

Our file Notre référence

ISBN: 0-612-93962-6

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

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

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

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

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

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

Canada

PAGINATION ERROR.

TEXT COMPLETE.

ERREUR DE PAGINATION.

LE TEXTE EST COMPLET.

Abstract

Automatic 3D Registration using Range and Colour Data refers to the process of automatically joining a pair of overlapping colour 3D images acquired using a laser range finder. Currently there are no methods that can automatically and reliably register overlapping 3D images. This Thesis presents a solution to the problem of automatic registration, which uses both the 3D data, along with co-registered 2D colour information. Features are found in the 2D data, but both 2D and 3D descriptors are attached to these feature points. The correspondences between these feature points in the two overlapping images are determined, and the transformation between the overlapping range images is computed from the best correspondences. The combination of 2D and 3D data exploits both the object geometry, as well as the object colour. Both the feature finding and correspondence process are automatic, and do not require any user intervention. A number of experiments are conducted to demonstrate the effectiveness of the approach.

Acknowledgements

I would like to thank my supervisor Dr. Gerhard Roth and the National Research Council of Canada for supporting me during my studies. Dr. Roth and the National Research Council were instrumental in my interest in this subject and they provided me with resources and data for success.

I would also like to thank my co-supervisor professor John Oommen for his support. Special thanks to Dr. David Lowe for making his keypoint software available to me. Thank you to Dr. John Williams for his comments and his MVR software. Thanks to Anthony Whitehead for making his PIL library and findcorner application available to me, as well to Dr. Jason Morrison and Weixuan Li for their help and interest in my Thesis.

Thank you to my beautiful wife, to Hildegard O Briain, my mother, and especially to Frau Lisbeth Prawitt, my grandmother, who provided me with support and strength for my education.

Table of Contents

Abstract	iii
Acknowledgements.....	iv
Figures.....	viii
Acronyms	xi
1 Introduction.....	1
1.1 Motivation.....	2
1.2 Problem Statement.....	4
1.2.1 Input Data.....	4
1.2.2 Manual Matching.....	6
1.2.3 Semi-Automatic Matching.....	7
1.2.4 Automatic Matching	8
1.2.5 Output Data.....	8
1.3 Thesis Contributions	9
1.4 Thesis Outline	10
2 3D Registration Survey.....	12
2.1 Fundamentals of 3D Registration	12
2.2 Purpose.....	13
2.3 Classification Methods.....	14
2.4 Geometry.....	16
2.5 Intensity and Colour.....	17
2.6 Direct Registration	19
2.7 Spin Images.....	21
2.8 Automatic Registration	23
3 Automatic Matching and Correlation.....	25
3.1 Overview.....	26
3.2 Image Data	28
3.2.1 Active Range Scanners	28

3.2.2	Active Sensor Data	30
3.2.3	Representation of 3D Transformations	32
3.2.4	Image Views	34
3.2.5	Range Image Errors	36
3.2.6	Data Conversion.....	38
3.3	2D Point Features.....	39
3.3.1	Feature Detection	39
3.3.2	Harris', Susan's, and Lowe's Keypoint Generators.....	40
3.3.3	Accuracy of Feature Points	42
3.4	2D Point Descriptors.....	44
3.4.1	Colour Correlation Window Method	44
3.4.2	Correlation Methods	46
3.4.3	Correlation Windows	47
3.4.4	Colour Spaces	48
3.5	3D Point Descriptors.....	50
3.5.1	Spin Images.....	50
3.5.2	Comparing Spin Images.....	54
3.6	Matching	55
3.6.1	The Matching Problem	56
3.6.2	Combining 2D and 3D descriptors	57
3.7	The Correspondence Problem.....	58
3.8	Correspondence Filtering.....	61
3.9	Triangle Alignment.....	63
3.10	Verifying the Transformation	65
4	Experiments and Results.....	67
4.1	Implementation Details.....	69
4.1.1	Generating Descriptors	69
4.1.2	Matching and Correlation Implementation.....	72
4.1.3	The Graphical User Interface.....	74
4.2	Manual Registration.....	75
4.3	Semi-Automatic Registration.....	79

4.3.1	View correspondences and overlap definitions	80
4.3.2	Colour correlation experiments.....	81
4.4	Automatic Registration	85
4.4.1	View correspondences and overlap definitions	86
4.4.2	Correspondences and thresholds.....	87
4.4.3	Automatic registration using spin images.....	90
4.4.4	Comparing feature detectors	93
4.4.5	Varying the window size	96
5	Conclusion	98
5.1	Summary	98
5.2	Future Work	103
	References	106
	Appendices.....	119
	Appendix A: 3D Transformation Matrices	119
	Appendix B: 3D Plane Fitting.....	121
	Appendix C: Combining descriptors example.....	124
	Appendix D: Maximum and Minimum values of a matrix.....	126
	Appendix E: Weighted Bipartite Graph Matching Example	128
	Appendix F: RGB to HSV Conversion.....	132
	Appendix G: Automatic Registration Results.....	133
	Appendix H: The Correlator GUI.....	141

Figures

Figure 1.2-1 Black box flowchart of the automatic registration problem; input are two 2.5D co-located images V_1 and V_2 , and the output is a transformation matrix T that aligns V_1 and V_2	4
Figure 1.2-2 Rendering of co-registered 2D duck image (top), the 3D duck image (bottom), and their correspondences drawn in red.....	5
Figure 3.1-1 The Automatic Matching and Correlation Input and Output.	26
Figure 3.1-2 The Automatic Matching and Correlation process using a pair of 2.5D images.	27
Figure 3.2-1 A laser range finder generating a 2.5D image.....	31
Figure 3.2-2 Sample view (V), and the overlap (S) configurations.....	34
Figure 3.2-3 (a) A set of duck images, the edges imply image overlap (b) the graph representation of the duck image views V_1 to V_5 showing the Transformation matrices T that align the image pairs.	35
Figure 3.2-4 Separating the 2.5D laser range data into 2D and 3D data files.	38
Figure 3.3-1 The input and output point feature procedure.	39
Figure 3.3-2The Harris, Susan, and Lowe feature detectors set to detect 40 feature points shown as (i), (ii), and (iii) respectively, where the arrows indicate potential matching points, and the green lines indicate automatically matched correspondences.	42
Figure 3.4-1 The Point Feature descriptor Process.....	44
Figure 3.4-2 A square 5x5 Colour Correlation Window taken from a duck image.....	45
Figure 3.5-1 (a) Circular window around an oriented point (b) definition of an oriented point V_0 on a plane π showing α and β distances to a point P_0	51
Figure 3.5-2 Spin Image Generation.....	52
Figure 3.6-1 The matching procedures input and output.....	55
Figure 3.7-1 The Correspondence methods input and output.....	58
Figure 3.8-1 The correspondence list is filtered and reduced to the best three correspondences.....	61
Figure 3.8-2 A set of Correspondences between image A and B.	61

Figure 3.9-1 The triangle alignment algorithm calculates the transformation matrix T computed from three correspondences.	63
Figure 3.9-2 Triangle Alignment algorithm steps (i) to (iv).	63
Figure 4.1-1 The process of generating descriptors from EXT images.	69
Figure 4.1-2 Programs and libraries used for the data conversion and feature point creation process. Note that programs denoted by (*) are existing programs and libraries written by other researchers.	70
Figure 4.1-3 Libraries implemented to generate feature point descriptors.	71
Figure 4.1-4 Implementation of the Matching process.	72
Figure 4.1-5 Libraries implemented to determine the correspondences and to generate the transformation matrix.	73
Figure 4.2-1 Manual selection of six pairs of correspondences of the duck image.	75
Figure 4.2-2 Manual identification of overlap of the duck image set.	77
Figure 4.2-3 Two duck images are registered and displayed.	77
Figure 4.3-1 A set of five overlapping totem pole images.	80
Figure 4.3-2 Spin Image matching (a), colour correlation (b), spin and colour (c).	81
Figure 4.3-3 Totem views V_1 and V_2 showing spin match (blue), colour match (red), and spin and colour results (green).	82
Figure 4.3-4 Registered VRML totem pole views V_1 and V_2 showing the side view (a) top view (b) and bottom view (c).	83
Figure 4.4-1 Four 2D views of the toupe image V_1 to V_4 shown as (a) to (d) respectively.	86
Figure 4.4-2 Toupe views V_3 and V_4 using a low threshold and 70 feature points.	87
Figure 4.4-3 Correspondences generated from Lowe's feature points using the toupe views V_1 and V_2 and 40 feature points. (i) shows the feature points, the arrows indicate points that do not have a correspondence, and (ii) shows the correspondences.	88
Figure 4.4-4 Three registered views (a) to (c) of a pair of toupe images.	89
Figure 4.4-5 Image view V_2 and V_3 of the toupe showing the feature points vs. the overlap of three feature detectors using only the spin image descriptor.	91

Figure 4.4-6 Image view V_2 and V_3 of the toupe showing the feature points vs. the accuracy of three feature detectors using only the spin image descriptor.	92
Figure 4.4-7 Image view V_1 and V_2 of the toupe showing the feature points vs. the overlap of three feature detectors.....	94
Figure 4.4-8 Results of the registration of views V_1 and V_2 of the toupe image using three feature detectors.	95
Figure 4.4-9 Toupe image V_1 and V_2 showing window size vs. the percentage overlap.	96
Figure 4.4-10 Toupe image V_1 and V_2 showing window size vs. percent accuracy.	97

Acronyms

BIRIS	BI-IRIS Range Scanner
CAD	Computer Aided Design
DoG	Difference of Gaussian
EXT	NRCC 3D file format
HSV	Hue, Saturation, Value Colour Space
ICP	Iterative Closest Point
MRI	Magnetic Resonance Imaging
MVE	Minimum Variance Estimator
MVR	Multiple View Registration
NRCC	National Research Council of Canada
PM	Progressive Mesh
PPM	Portable Pixmap Utility (2D image file format)
PRIME	Profile Imaging Module
RGB	Red, Green, Blue Colour Space
SVD	Singular Value Decomposition
VRML	Virtual Reality Modeling Language

1 Introduction

Model building refers to the process of taking 3D data acquired by a laser rangefinder and making a geometric model of an object using this data. An important open problem in model building is how to automate the registration step, in which 3D range data obtained from different viewpoints must be transformed to be in the same coordinate frame.

This Thesis describes a particular solution to the automatic registration problem that uses matching and correlation of image data points. The input consists of 3D data acquired by a laser range finder along with co-located 2D colour images. This means that there is a color value associated with each 3D data point. Such 3D images are called 2.5D images because the points are ordered into an image format and at the same time are actual 3D points. Our approach automatically finds appropriate feature points and then matches them between each pair of overlapping 3D range images. The most important contribution of this research is the use of both the 2D and 3D data to automate the matching process. Using this technique, the 3D data images can be registered directly, without an initial pose estimate. Along with a description of the algorithms that are used, a number of experiments are described that demonstrate the effectiveness of the method.

The research has implications in the areas of industrial product design, virtual museums, medical imagery, architecture, archeology, robotics, and imagery in film and multimedia.

1.1 Motivation

The invention of the low-cost commercial 3D-laser scanners and publicly available 3D application interfaces like OpenGL, along with display technologies like VRML have generated a great deal of interest in Model Building in the Computer Graphics and Computer Vision Community. Since computers are now very powerful and have considerable memory, it is feasible to work with dense 3D data acquired by a laser range scanner on an ordinary PC.

In the Model Building process a critical aspect of the process is the 3D Registration step, in which overlapping 3D images are placed in the same reference frame. Currently, this registration step is done manually. The goal in this Thesis is to create an automatic registration process that does not require any human intervention. To achieve this goal, a method that matches features between a pair of overlapping 3D images has been created.

There have been previous attempts to automate this registration process. The Iterative Closest Point algorithm [BeM91] can register images, but only if they are already approximately aligned. Thus this approach is not automatic. Spin images can be used as 3D point descriptors, and there are automatic registration methods based on spin images [JoH99]. However, methods that use only spin images, do not make use of any co-registered 2D intensity information that is available along with the 3D data and are not always successful. While the method we describe does use spin images, it also incorporates 2D information during the matching process, and is thus an improvement over spin images.

In summary, this Thesis describes a particular solution to the automatic Registration problem, which finds 2D and 3D features, and matches these features between overlapping 3D images using different types of correlation algorithms.

1.2 Problem Statement

We assume that we are given two overlapping 2.5D images that are not aligned, in other words, the two images are not in the same coordinate system. The aim of this study is to attempt to find the transformation matrix, T , that aligns the two 2.5D images (Figure 1.2-1).

Figure 1.2-1 Black box flowchart of the automatic registration problem; input are two 2.5D co-located images V_1 and V_2 , and the output is a transformation matrix T that aligns V_1 and V_2 .

1.2.1 Input Data

The images used by the matching process were generated by a colour laser range finder [Rio94]. The 2.5D images have the advantage of having 2D as well as 3D data. Every point on the 2D image has a corresponding 3D data point, the 2D data is said to be *co-registered* with the 3D data (Figure 1.2-2). Thus 2D imaging techniques can be applied to the 2D data while still having the underlying 3D data available.

Figure 1.2-2 Rendering of co-registered 2D duck image (top), the 3D duck image (bottom), and their correspondences drawn in red.

The input data consists of a pair of 2.5D co-located data sets that are overlapping.

Two overlapping 2.5D images can be registered in a number of different ways. The registration process requires two things; the selection of feature points and their correspondence across the images. The registration methods discussed in this Thesis are called manual matching, semi-automatic matching and automatic matching.

1.2.2 Manual Matching

Manual Matching refers to the process of selecting image feature points by-hand. This can be done by displaying two images side by side, and permitting the user to manually choose correspondences in a point-to-point manner.

Manual Matching is a time consuming task which relies on human judgment and accuracy. The interesting aspects of manual registration are:

- What registration algorithm is used,
- How the algorithm deals with multiple points, and
- How successful the procedure is.

The Thesis implements manual matching to determine the amount of overlap between the image pairs. This value is considered to be the ground truth to which the results are to be compared.

1.2.3 Semi-Automatic Matching

Semi-automatic Matching refers to the process of automating the matching process. The idea is to improve the accuracy and speed of the registration process by limiting the user's manual input. If we were to display two images side-by-side, the user can manually select a number of features in both image I_1 and image I_2 . Then an automatic process will determine the correspondences between the selected points. Since the feature selection process is manual, and the feature matching process is automatic, the entire process is semi-automatic.

Semi-automatic registration deals with:

- How to automatically determine correspondences between a set of features in two point sets,
- How to measure the performance of various automatic matching methods,
- How to use colour information during correlation, and
- How to register and verify the alignment of the data.

The Thesis then resorts to semi-automatic registration in order to test the success of the automatic matching and automatic correspondence methods.

1.2.4 Automatic Matching

Automatic registration attempts to automate both the feature point creation as well as the matching process. The matching process can be automated by using 2D keypoint generators such as the corner or edge detectors. This would save the user the time consuming task of determining accurate features. Given a pair of 2.5D images the process will automatically generate features in both images, generate 2D and 3D descriptors, match the descriptors, choose the best three correspondences for registration¹, calculate the transformation matrix, and finally register the data. The Thesis will also implement a strategy for automatic registration using a pair of co-located 2.5D images, and outline the success of the approach.

1.2.5 Output Data

The result of our Automatic 3D Registration method using Range and Colour Data is a 4x4-transformation matrix that can transform the 3D data points from image V_1 to image V_2 . The transformation matrix T is comprised of six basic transformations: translation in X,Y and Z as well as rotations about X,Y and Z. These six basic transformations constitute *six degrees of freedom*².

¹ Note that our method requires only three correspondences to register the data.

² Please refer to Appendix A for a detailed description of the basic 4x4 3D transformation matrices.

1.3 Thesis Contributions

This Thesis will outline a complete method that can register two overlapping colour 3D images. The automatic registration method combines 2D feature point descriptors with 3D descriptors to determine reliable correspondences. It is shown that by using the 3D geometric information, along with the co-registered 2D intensity images that the overall success rate and accuracy of the automatic registration process is improved.

Further contributions include:

- Described a novel method to register a pair of 2.5D images by matching feature points,
- Implementation of the automating matching and correlation method,
- Demonstrating that the use of colour correlation windows can increase the registration accuracy,
- Evaluating several automatic 2D keypoint generators,
- Automating keypoint generation,
- Automating correlation,
- Demonstrating that it is possible to accurately register a pair of images by matching feature points,
- Converting the feature matching problem into a graph problem, in particular a complete weighted bipartite graph,
- Comparisons of feature detectors when they are used in automatic matching.

1.4 Thesis Outline

The second Chapter gives a survey on 3D registration techniques. Existing registration techniques are described, including iterative, direct, and automatic methods. Several automatic and multiple view registration techniques are discussed.

In Chapter 3 we consider the problem of how to automatically register a single pair of overlapping range images. These images consist of a set of 3D data points ordered in an image format, which are called 2.5D images. The approach we describe requires 2.5D images which are co-registered. In other words, for each 3D data point in the 2.5D image, there is also an intensity value, in R, G, B form which represents the color of that associated 3D data point. Corner features are found in this co-registered intensity image using different corner operators; in particular the Harris, Lowe and Susan operators. A type of color correlation process is used to match these 2D features between overlapping images, and at the same time spin images are used to match the 3D features. The matching process uses the 3D information to compute the registration transformation by using the method of triangle alignment.

The experimental results are described in Chapter 4. The first section describes the implementation details. The implementation details include a description of the libraries used as well as a description of the graphical user interface. Section 4.2 describes how a number of overlapping range images are registered manually to provide the accurate ground truth transformations. The results of this manual registration are then compared to the results of the automatic registration process. The experiments are conducted using

different images, correlation window sizes, descriptors, the number of feature points, and the correlation thresholds. In Chapter 5 we describe our conclusions, and suggest some future work.

2 3D Registration Survey

The registration process attempts to find a set of transformation matrices that will align all the data sets into a world coordinate system. The question of how accurately and efficiently this is done, is dependent on how the resulting model is to be used.

2.1 Fundamentals of 3D Registration

The method of connecting two overlapping 3D range images together is called *3D registration*. An effective surface registration algorithm must be able to handle complex scenes, and not make any assumptions about the transformations between the views.

We formally define 3D registration as the problem of determining the transformation $\mathbf{T}$, in which we minimize:

$$e = \sum_{i=1}^{N_p} \|\mathbf{T}\mathbf{p}_i - \mathbf{q}_i\|^2 \quad (2.3.1)$$

where $\mathbf{T}$ is a 4×4 3D transformation matrix with six degrees of freedom, $\mathbf{p}_i \in P$ and $\mathbf{q}_i \in Q$ are the scanned 3D control points, and $N_p = \|P\|$. Solving the registration problem involves determining the transformation $\mathbf{T}$ that minimizes the distance between the points of the two data sets P and Q .

The goal of image registration, also known as the correspondence problem [ChM92], is to find the transformation between the data obtained from two different views.

2.2 Purpose

There are four basic parameters that quantify the success of a registration algorithm.

These parameters are its:

- Accuracy
- Efficiency
- Robustness
- Visualization

If we were registering images with the intent of manufacturing an object, we would want a high degree of accuracy. In this case, it would be advantageous to register a large number of data points, with low errors, and with little or no texture mapping.

Typically, there is a balance between precision and the complexity of a registration algorithm. In the CAD world, precision is paramount even at the cost of object visualization. Camara and Campos [CaC99] developed a method used by CAD designers to aid them in the visualization of the object using VRML. Their paper presented a study of the minimum number of captures required to obtain a complete description of an object.

Bernardini *et al.* [BMR01] recognized that integration of high-quality texture on geometric data has been a neglected aspect of the model building process. Using

inexpensive electronic camera-based systems, they outlined a high-quality texture reconstruction method. The key idea in their research was to produce a model of high visual quality, while they sacrificed a dense geometric representation. The registration procedure used local searches on small regions based on image correlation indices which were computed using sample pixels in areas rich of detailed image features, and which minimized the occurrence of false matches.

2.3 Classification Methods

There are several classes of registration algorithms. They are not necessarily mutually exclusive, although each one is associated with a different research area. In the interest of simplicity, we identify four distinct classes of registration algorithms [Bro92]:

- Multimodal Registration
- Template Registration
- Temporal Registration
- Viewpoint Registration

Multimodal Registration techniques are methods that integrate different sensor images. The idea is that Multimodal Registration will provide improved segmentation and pixel classification. For example, the Environment Modeling process can combine the output of a 3D-laser scanner with a colour video camera.

Template Registration methods attempt to match a reference pattern within an image window. Typically, template registration techniques are used to locate object models or patterns. This can give rise to model-based methods, and is particularly useful when *a priori* information is present in a scene. Geographical image analysis can use template matching to locate specific features and register images.

Temporal Registration techniques are methods in which images are taken at different times, and at different positions. Surveillance methods use temporal registration techniques [SiB97] [SzC94], and they are also used as in automated inspection systems [NGU99].

Finally, viewpoint registration techniques are methods in which images are taken from different vantagepoints, and then are registered together. Viewpoint registration techniques are typically used in model building and environment modeling.

2.4 Geometry

The classical problem of image registration deals with the geometry of 3D image points in order to register the images. The basic data unit is the pixel from which a wide variety of higher-level geometric models can be created. Edges, contours and surfaces can be created using a segmentation algorithm. As mentioned in [Bro92] many salient features can be generated such as:

- local maximum curvatures,
- centers of gravity of closed bounded regions,
- line intersections,
- Fourier descriptors.

These techniques also use statistical feature matching, and includes concepts such as:

- Moment Invariants,
- Centroid Calculation,
- Histogramming.

Additionally, higher level feature detection includes:

- Structural graphs,
- Feature point detection,
- Matching against models.

The most difficult problem in 3D registration occurs when the relative position of each scanned view is unknown. In this case, the camera position is unconstrained, and thus has six degrees of freedom. This class of problems is called *pose determination* [SLH98]. However, if there is prior knowledge of the camera position (say for example, from a robot holding the camera) then the problem is simpler, and is called *pose refinement*. The reason is that the pose has already been determined approximately, so that it must only be refined further, thus the term pose refinement.

2.5 Intensity and Colour

In addition to XYZ data, colour laser scanners can also acquire the RGB triplet of the scanned point, typically done by using an additional colour video or digital camera [Rio94]. In this case, it is possible to use the registered color information, to help solve the registration problem.

In 1997 Weik [Wei97] proposed a method to register 3D partial surface models using luminance and depth information. The paper describes a method that uses both camera-based depth information (e.g. from stereo) and the luminance image to find corresponding points between images. The author claims that depth information, in addition to luminance information, can speed up the process when registering sphere-like shapes. Weik described a stereo model that is used as part of the registration process. The *corresponding point set registration algorithm* calculates the average point-to-point

distance between two corresponding point sets. A cross-covariance matrix is then calculated involving the two point sets from which the optimal rotation and translation matrix is derived.

The main aspect of the *corresponding point set algorithm* involves registration using the luminance information of the corresponding points. The algorithm incorporates the stereo and quaternion methods to register the position of the destination camera with respect to the source camera. This is achieved by calculating intensity differences between the two point sets together with the projected image gradient to predict corresponding image points. The testing of the algorithm has several limitations: point sets were assumed to be evenly distributed, the point set size was about 3000 points, rotation angles of the two images was less than 30 degrees, and the point sets had high image gradients. Another limitation of the algorithm is that both image sets were forced to have exact overlap since the point sets had to lie in each of the two image planes.

2.6 Direct Registration

Direct Registration refers to techniques that solve the registration problem using a single algorithmic step rather than iteratively converging to the solution. The following briefly describes some existing direct registration techniques.

Early work by Arun *et al.* [AHB87] on estimating 3D rigid body transformations presented a solution using the singular value decomposition (SVD). The method requires a connected set of correspondences and accurately registers the 3D data.

Faugeras and Hebert [FaH86] employed the quaternion method to solve the registration problem directly. They required a set of correspondences, and then converted the registration problem into a problem that involved minimization of a cost function for rigid motion, which can be solved with a quadratic form of a unit quaternion. They then showed that the method is theoretically sound and that the solution converges [MaY95].

Eggert *et al.* [ELF97] compared four major 3D rigid body transformation algorithms:

1. Arun *et al.*'s algorithm based on the SVD of the standard rotation and translation matrix.
2. Horn *et al.*'s algorithm that exploited the orthonormal properties of the rotation matrix, and that computed the eigensystem of the derived matrix.

3. Horn's algorithm that computed the eigensystem of a matrix relating the rotational components of the unit quaternion.
4. Walket *et al.*'s algorithm that analyzed the dual quaternion representation of the eigensystem of the transformation matrix.

Eggert *et al.* based their comparative results on experiments determining accuracy, robustness, stability, and efficiency, and found that no single algorithm was superior in all cases.

Sun *et al.* [SDA00] presented a mesh-based integration of range and colour images. Their goal was to create photo-realistic colour 3D models from multisensor data. The colour and texture data was first processed separately and then integrated at the end of the procedure. Even though the explicit registration technique was not discussed in their paper, the ideas employed are clear. First, they created a triangular mesh of each view, registered them together, removed overlapping triangles from the final model, and then added colour and texture. Registering mesh representations of the original range data has many advantages since one can easily process prohibitively large data sets using such a mesh approach.

2.7 Spin Images

In 1997, Johnson and Hebert [JoH97] [JoH99] published a paper titled, *Surface Registration by Matching Oriented Points*. Their *spin-image* algorithm procedure is as follows:

1. Two views of a scene are acquired using a range scanner and then converted into a triangular mesh surface.
2. Point correspondences are found using their *spin-image* algorithm of matching points on surface objects.
3. Geometrically consistent point correspondences are used to compute a rigid transformation that aligns views.
4. The transformation is refined using a modified ICP algorithm.

The fundamental shape element used is an oriented point having a surface normal pointing towards the sensor. The oriented point, in turn, defines a *spin-map* which is a projection that maps 3D points to its 2D coordinates. The 2D spin-images thus represent the shape of the surface independent of its pose. The algorithm then attempts to compare spin-images to identify correspondences. Finally, a modified ICP algorithm finds the best transformation between the model and the scene by eliminating transformations that are inconsistent.

Environment Modeling requires the collection of a large amount of 3D data. Carmichael and Hebert used “spin-images” in order to find the correspondences between two 3D

points [CaH98]. By comparing the coordinate information from the spin-images of both points, they determined a measure of the local shape similarity. Their procedure converted the data into a low-resolution mesh and computed correspondences after which the image was registered.

2.8 Automatic Registration

So far, we have described methods that calculate a transformation matrix given a set of correspondences between the 3D points in each data set. The iterative methods use an initial approximate estimate of this transformation to get these correspondences. Direct techniques also require a set of accurate correspondences as input. Obtaining correspondences using *a priori* information such as data signatures, target locations, and approximate camera positions is difficult and time consuming if done manually. The ultimate goal in automatic registration is to automate all the steps in this process including the correspondence step. Automatic registration is a difficult problem and there are no robust methods that can automatically register all types of 3D images. Even the best automatic methods have difficulties in registering images that are symmetrical or have homogeneous geometry and intensity. However, there has been some success in automatic registration; the following paragraphs will briefly describe several automatic registration methods.

Chen *et al.* [CHC99] proposed a random sample consensus (RANSAC) scheme that is used to check all possible data-alignments of two data sets. First of all, control points are selected and the data alignments calculated. If the transformation falls above a threshold it is considered to be a solution to the registration problem. A statistical analysis showed that only a few random trials are required to solve the partially overlapping registration problem. The authors claim that their scheme works with featureless data and requires no initial pose estimate.

Roth [Rot99-2] proposed a scheme that automatically registers two overlapping range images using the 3D data itself without any *a priori* information. The only assumption is that the 3D data has more than 20-30% overlap, and that the 3D data contains co-located intensity data. Correspondences are determined automatically using points of interest found in the intensity data. A Delaunay Tetrahedrization of these interest points is constructed and all compatible triangle pairs are used to register the range images.

Andreetto *et al.* [ABC03] automatically registered archaeological objects by pairwise registering 3D views, and then globally aligning them. Two difficult approaches were defined to automatically detect common regions, which were:

1. Invariant statistics for each point of the mesh
2. Special features based on geometry or texture

Spin images were used to automatically detect common regions, and then an ICP algorithm was invoked to refine the spin image estimates. Spin images proved to be a robust method to identify 3D descriptors or shape characteristics, of the object. The spin image technique was able to determine an effective starting point of the ICP algorithm.

3 Automatic Matching and Correlation

Automatic Matching and Correlation is an approach to solve the pair-wise registration problem. That is, given two overlapping sets of 2.5D images, find the registration that aligns them. Given a 2.5D image, features are generated and then correspondences determined by comparing 2D and 3D descriptors. The best matches are used to generate a transformation matrix and then the pair of images is registered.

3.1 Overview

The goal of the Automatic Matching and Correlation procedure is to take a pair of overlapping 2.5D images V_1 and V_2 , performs some computations, and then outputs a transformation matrix, T , that aligns the two images (Figure 3.1-1).

Figure 3.1-1 The Automatic Matching and Correlation Input and Output.

Automatic Correlation and Matching align a pair of adjacent 2.5D images by using point features. Figure 3.1-2 shows an overview of the Automatic Matching and Correlation process.

Figure 3.1-2 The Automatic Matching and Correlation process using a pair of 2.5D images.

The first step is to generate features³ in each 2.5D image. After the features are found, 2D and 3D descriptors of these features are generated. The features must then be compared and then matched in order to create a set of correspondences. Once the correspondences have been determined, a correspondence filtering algorithm determines the best three correlating points. The process of triangle alignment takes these three data points and calculates the transformation matrix that can align the two images. The images are registered using the transformation matrix and the alignment is verified. The following Chapter will explain the Automatic Matching and Correlation procedure in detail.

³ The terms keypoints/corners and feature points will be used interchangeably.

3.2 Image Data

This Thesis uses 2.5D images that were created using an active laser range finder. The laser range finder is moved around an object acquiring scans at different locations. Each scan of the laser range finder creates a set of 3D data and colour 2D intensity data which is co-located. We call this particular combination 2D and 3D data *2.5D images* because each pixel of the 2D image has a corresponding 3D data point. A pair of 2.5D image views is used as input for our registration method.

3.2.1 Active Range Scanners

Active sensors project light onto the object using a source such as a laser beam. The most common principles of the technologies that use active sensors are [Rot99]:

1. Time of Flight,
2. Triangulation,
3. Structured Light [DeT95].

As we have stated, to create a model of an entire object it is necessary to acquire data over the entire surface of that object. Active sensor can achieve this goal using the following methods:

1. The object is fixed, and the scanner is translated and rotated, or
2. The scanner is fixed and the object is translated and rotated.

A common image acquisition technique used by model builders is to scan the object while it is on a turntable. The rotation of the turntable can be accurately determined, and the object can be reconstructed quite precisely [JoO00]. The main difficulty is encountered when the top and bottom of the object are to be modeled, since these areas cannot be scanned in a single pass.

Using a “coordinate measuring machine” [BeR00] it is possible to accurately track the camera position and orientation. However, such accurate robotic devices are very expensive. Moving the scanner around the object to be modeled is another common 3D-acquisition technique. This method has the advantage of not requiring any mechanical aids such as a turntable or robot. This is an attractive option because it reduces the complexity of the 3D imaging equipment. However, solving the registration problem in this situation is more difficult, since nothing is known a-priori about the different positions of the 3D sensor, which is often the case if the sensor is moved using a robotic device. This Thesis makes no assumptions of the initial pose of the laser range finder.

3.2.2 Active Sensor Data

There are a number of ways to format the 3D data points acquired by an active sensor. The raw 3D data points can be processed and ordered according to readability, to minimize space constraints, or stored algebraically.

The 3D data points acquired by an active sensor are normally considered *dense* since many data points are acquired with a very small spacing between these points on the surface of the object. The type of active sensor used to acquire the data will dictate the type of raw data that is produced. There are three main forms of 3D data produced by active sensors:

1. Point clouds
2. Profiles, and
3. Range Images

Point clouds are acquired in an unordered manner using a point sensor. If we placed the point sensor on a pan-tilt controller, and scanned along an axis in a regular manner, we would acquire a profile or a line of the 3D points. The only assumption we make on the profile is that each line was acquired in a consistent manner. We can also acquire a profile directly by using a laser which is spread into a profile using a lens. Such an active sensor is called a profile scanner. If we have the ability to place a profile scanner on a pan-tilt unit and scan along an axis, we would acquire a range image. A range image

consists of an ordered set of 3D data points. Raw 3D range images are commonly referred to as 2.5D images since the scanning of the image scene is done in a single pass from one vantage point (Figure 3.2-1). If the intensity or colour is present in the 2.5D image, it is quite natural to view the scene in 2D.

Figure 3.2-1 A laser range finder generating a 2.5D image.

The advantage of a 2.5D image is that for every 2D data point there exists a corresponding 3D data point. It is possible to use 2D feature detectors as well as invariant 3D descriptors on the image data.

The colour laser range images used for this Thesis were scanned by the National research Council of Canada's Institute of Information Technology group [Rio94]. A high resolution colour laser scanner based on the auto-synchronized and triangulation principle, was used to scan the images. The spatial resolution of the scanner is 50 microns and the depth resolution is 10 microns [TBG02].

3.2.3 Representation of 3D Transformations

The basic language used in 3D modeling involves the composition of 3D transformations. 3D transformations are represented by means of 4x4 matrices that are composed of primitive transformations such as the translation in the X, Y, and Z directions, and the rotation about the X, Y, and Z axis.⁴ These six primitive transformations constitute *six degrees of freedom*. Therefore, if we include the notion of a *world* coordinate system, each registered image consists of registration data (geometric, intensity, and possibly, colour data), as well as the position of the image in space represented by the 4x4 coordinate matrix.

⁴ Please refer to Appendix A for a detailed description of the basic Transformation Matrices.

3D image transformations have six degrees of freedom: the translation in X, Y, and Z directions and the rotation about X,Y, and Z axes. Solving the problem of determining the camera position with a brute force algorithm would require us to exhaustively search a six dimensional space. Since this is not practical, a lot of work has been done to reduce the time-complexity of finding a solution to the registration problem.

3.2.4 Image Views

Figure 3.2-2 defines several views V^i where $i = 1, \dots, N$, are the number of scans, and S^j defines the number of overlaps $j = 1, \dots, k$.⁵

Figure 3.2-2 Sample view (V), and the overlap (S) configurations.

It is clear that for two images P and Q to be registered, $P \cap Q \neq \emptyset$, in other words the intersection cannot be null in order for a meaningful pairwise registration to occur.

To create a 3D model of an object it is necessary to capture 3D data sets covering the entire surface of the object from a variety of viewpoints (Figure 3.2-2). A network of adjacent images is generated. Adjacency refers to the topological relationship between

⁵ The image overlaps are also referred to as correspondence sets.

the acquired images. Figure 3.2-3 (a) shows five duck images as nodes in a graph and the lines that are joining the nodes indicate that there is an overlap between the image pairs.

Figure 3.2-3 (a) A set of duck images, the edges imply image overlap (b) the graph representation of the duck image views V_1 to V_5 showing the Transformation matrices T that align the image pairs.

Thus, given a set of 2.5D duck images V_1 to V_5 , our aim is to compute the transformation matrices $T_{(i,j)}$ that align each image pair, as shown in Figure 3.2-3 (b).

3.2.5 Range Image Errors

In any practical setting there are errors that will affect the registration algorithm. Errors may be due to:

- the scanning resolution and accuracy of the range sensor
- camera defects
- environmental factors
- limitations of the range sensor

Additionally the result of such errors gives rise to issues such as:

- static noise
- outliers
- specularities
- perspective irregularities
- colour and intensity errors
- defocusing.

All image errors can be classified as:

1. Static or Dynamic
2. Internal or External
3. Geometric or Photometric

Static errors can be corrected in the same manner for all images, and often occur due to calibration errors. If the camera lens has a small defect, or dust/dirt is present, the image may have a static error. The errors can be identified as spots or areas of static noise, that are present in all the images. Dynamic errors can be caused by environmental factors, including shadows, cloud coverage, object movements, deformations, growth, or atmospheric changes between images.

Internal errors refer to the hardware limitations of the camera. Internal errors include mis-calibration errors, and errors due to spectral characteristics, focal length, defocusing, distance to the scene and aberrations. When a camera attempts to scan a point outside its design specifications, it results in outliers or sensor noise. External errors can be caused by environmental factors that cause disturbances to a scene, or by continuously changing sensor operations.

Photometric errors are partially due to internal errors that affect the intensity values of the image. Photometric errors include specularities due to highly reflective and mirror-like surfaces. Geometric errors may occur due to the sensor scanning outside its calibration range. This can give rise to perspective changes from the viewpoint of the sensor.

A registration method that uses range images must be robust in order to handle many types of data errors.

3.2.6 Data Conversion

The 2.5D data images were taken by a colour laser range finder from the National Research Council of Canada (NRCC) and stored in a proprietary format called the EXT format (Figure 3.2.4).

Figure 3.2-4 Separating the 2.5D laser range data into 2D and 3D data files.

Accompanying the EXT images were C libraries that could read and write the NRC format. The first step was to separate the 2.5D EXT format to the popular PPM and VRML data formats⁶. The EXT images were read using the C function `export_range` and were then converted.

⁶ The PPM 2D format was chosen instead of the JPG format since the PPM files are non-lossy.

3.3 2D Point Features

The point feature problem takes as input a 2.5D image. The 2.5D image is composed of a 3D VRML file and a co-located 2D PPM file. The output is a list of point features F , that were identified in the input image (Figure 3.3-1).

Figure 3.3-1 The input and output point feature procedure.

We used the 2D component of the 2.5D image in order to automatically determine the feature points. Since it is possible to use well-established 2D feature detectors we do not have to use ‘expensive’ 3D point feature methods.

3.3.1 Feature Detection

Feature detection is dependent on which geometric invariants are used by the detection algorithm. An invariant of that image is a function of the 2D image whose value is unchanged by a particular transformation. For example, a corner is a good feature since it remains a corner in Euclidean space after a transformation. Corners are described in 2D using the intensity information of the image. The idea is to find corners as features in the 2D data, and then to compute a 2D and a 3D descriptor of the region around that corner.

3.3.2 Harris', Susan's, and Lowe's Keypoint Generators

Keypoint generators [ZiT98] [HLS02] [MiS03] are a significant part of an automatic registration system. If the automatic registration system uses correspondences between keypoints then the keypoint generator must generate accurate keypoints. The idea is to generate keypoints in image 1 and determine the corresponding keypoints in image 2. Assuming that the image has enough overlap so that some of the points between image 1 and image 2 match, then it is possible to determine the feature correspondences. Three keypoint generators used were developed by:

- Harris [HaS88],
- SUSAN [SmB95], and
- Lowe [BrL02] [Low03]

which are explained below. All three keypoint generators operate on 2D images.

The Harris keypoint generator [HaS88] uses a combination of an edge and corner detector. Harris uses the idea of a junction: an edge meeting at a corner to define his keypoint generator. The method is similar to Moravec's corner detector [Mor80] that looks at the average intensity values of a local window in an image after shifting the image in various directions.

The SUSAN principle [SmB95] can also be used as a keypoint generator for detecting edges and corners in 2D images. The concept of having each feature point associated with a local area of similar brightness is the basis for the SUSAN principle.

Lowe's scale invariant keypoint generator [BrL02] [Low03] uses a sequential filtering approach to reduce the number of 'expensive' operations that are applied.

The stages in the keypoint detection step are:

- Scale-space peak selection
- Keypoint Localization
- Orientation Assignment
- Determination and utilization of Keypoint Descriptors.

The first stage of computation determines points that are invariant to scale and orientation by using a difference-of-Gaussian (DoG) function to identify potential interest points. Keypoints are then selected on the measure of their stability. The Orientation assignment is based on local image properties, and all future operations are performed relative to this assignment. Finally, the keypoint descriptor uses image gradients around each keypoint that allows for local shape distortion and change in illumination. The Lowe keypoint generator defines large number of features that densely cover the image.

3.3.3 Accuracy of Feature Points

The quality of the feature points is crucial in the matching process. If the feature detector cannot choose points that match between the images, then the matching process has no hope in succeeding. The feature detectors will behave in a different manner for different images.

Figure 3.3-2 The Harris, Susan, and Lowe feature detectors set to detect 40 feature points shown as (i), (ii), and (iii) respectively, where the arrows indicate potential matching points, and the green lines indicate automatically matched correspondences.

Figure 3.3-2 shows the three sets of the totem pole images using 40 feature points per image. The white arrows represent feature points that are a possible match. These were chosen manually to highlight how few possible matches there are between the images. Figure 3.3-2 (i) used the Harris feature point detector, where there are 7 possible correspondences out of a total of 40 feature points. Out of those 7 correspondences only one pair was accurate enough for registration. Figure 3.3-2 (ii) shows the results of the Susan feature point detector which only had four sets of points (four arrows per image)

that could generate correspondences, and again only one set was accurate enough for the registration. The last Figure, 3.3-2 (iii), shows the Lowe feature detector, which had a cluster of feature points around the totem pole eye. The accuracy of the registration decreases for such a cluster of points. If three points in the cluster are used to determine the registration, even a small amount of error can generate a large error in the pair-wise registration.

The totem pole itself has a high-sheen surface and light reflects differently all along its surface. When changing the view, the intensity values change due to the sheen of the surface. Since the feature detectors depend on intensity, the totem pole images were a particularly difficult choice for generating quality feature points.

3.4 2D Point Descriptors

The point descriptor function takes as its input a list of feature points. It then computes the descriptors of each feature point and outputs the list of descriptors (Figure 3.4-1).

Figure 3.4-1 The Point Feature descriptor Process.

The following section will describe how colour correlation windows are used to generate 2D point descriptors.

3.4.1 Colour Correlation Window Method

When colour is present in the images, it can be used to determine correspondences between 2D keypoints by comparing a small window around each keypoint using image correlation. Figure 3.4-1 shows a square 5x5-colour correlation window around a keypoint on the duck surface.

Figure 3.4-2 A square 5x5 Colour Correlation Window taken from a duck image.

The key elements required for colour correlation matching are:

- the Correlation Window Method,
- the Colour Spaces, and
- the Window size.

Colour Correlation windows are typically square or circular. The choice of the colour space, and the correlation method are parameters that affect the correlation.

The Colour Correlation Window Matching using colour spaces match feature points between two 2D images I_1 and I_2 using different colour spaces. The correlation function returns a probabilistic value representing the probability that feature point of I_1 matches a particular feature point of I_2 . The motivation is that after the feature points between the

images are matched, the images can be accurately registered using only the image data [Rot99-2].

3.4.2 Correlation Methods

Assume that we have two images I_1 and I_2 , and that $I_1(i), I_2(i) \in [0, k]$, where k is a small integer, and the index i ranges over all N pixels. The correlation function is defined as $\phi(I_1, I_2) \in [0, 1]$, where 0 is a perfect match and 1 a perfect mismatch.

The differences of the sum of the squares, equation (3.4-1), is called the *least squares correlation method* and is given by:

$$\phi = \frac{\sum_{i=1}^N (I_1(i) - I_2(i))^2}{k^2 N} \quad (3.4-1)$$

The *histogram method* places all pixels of the images I_1 and I_2 into j equally distributed histogram bins H . Each bin $H(i)$, has size k/j and the bins are compared to each other using the sum of the differences per bin:

$$\phi = \frac{\sum_{i=1}^j |H_1(i) - H_2(i)|}{2N} \quad (3.4-2)$$

The *ordinal measures correlation method* is based on the relative ordering of the intensity values in an image window [BhN98]. A rank permutation is used to describe the ordering, and a distance metric is used to define the ordinal measures.

The *normalized cross-correlation* of image pairs I_1 and I_2 is given by:

$$\varphi = \frac{\sum_{i=1}^N (I_1(i) - \bar{I}_1)(I_2(i) - \bar{I}_2)}{2N\sqrt{\sigma(I_1)\sigma(I_2)}}, \quad (3.4-3)$$

where

$$\sigma(I_1) = \sqrt{\sum_{i=1}^N (I_1(i))^2 - (\bar{I}_1)^2}, \quad (3.4-4)$$

and $\sigma(I_2)$ is defined similarly as the standard deviation of the pixel images. $\bar{I}_1$ and $\bar{I}_2$ are defined as the average pixel values of I_1 and I_2 respectively.

These different correlation methods are used to compare the local image regions around each local feature. The result of the correlation is the measure of similarity between keypoints in the two images.

3.4.3 Correlation Windows

Correlation windows are typically square or circular. Window sizes can be as small as 3x3 pixels, and they typically vary in size by an odd number of pixels so that there is a well-defined window center.

The process of converting a square correlation window to a circular correlation window is given in Figure 3.4-2.

Figure 3.4-3 Converting a square Colour Correlation window to a circular Colour Correlation Window.

3.4.4 Colour Spaces

There are different ways to represent colours in a digital image. The RGB colour space is a widely used model that generates various colour using its three components: red, green, and blue. In practice, the values of each component fall within the range $[0,255]$, which results in a possible 16.78 million different colours.

The HSV or Hue-Saturation-Value colour space was designed to be user-oriented since it deals with colours intuitively using tint, shade, and tone. The translation between RGB to HSV, and HSV to RGB is well known⁷.

⁷ Please refer to Appendix F for sample C code that converts RGB to HSV.

Some of the other colour spaces include HIS (Hue-Saturation-Intensity), CIE (Commission Internationale de l'Eclairage's) which is a linear combination of the RGB colour space, TSL (Tint-Saturation-Luminance), CMY (Cyan-Magenta-Yellow), and YIQ which is used by the U.S. for TV broadcasting [MiR01].

3.5 3D Point Descriptors

In this section we describe the task of determining invariant descriptors for 3D objects using 3D point descriptors. The data input and output is the same as in Section 3.4, that is, the input is a list of feature points and the output is a list of point descriptors.

Spin images define an oriented point from which a 3D-point descriptor is defined. An invariant descriptor is a shape descriptor unaffected by object pose, by perspective projections, and by intrinsic camera parameters [FMZ91]. Once the 3D-point descriptor is defined, the spin images can be compared, for example, using a normalized linear correlation coefficient.

3.5.1 Spin Images

We now describe a spin image that is used as a 3D descriptor of a feature point. The key component of a spin image is an oriented point. An oriented point has a direction perpendicular to the surface of the 3D image $\bar{n}$, which is pointed towards the camera. An oriented point is generated by calculating the plane of best fit⁸, π (drawn as an ellipse in figure 3.5-1 (b)), of the surrounding data of the oriented point, and then generating the normal to the plane π passing through the point V_0 .

⁸ To determine the plane of best fit please refer to Appendix B.

Figure 3.5-1 (a) Circular window around an oriented point (b) definition of an oriented point V_0 on a plane π showing α and β distances to a point P_0 .

The α and β distance values are generated from the coordinates of the point $P_0 = (x_0, y_0, z_0)$ relative to the oriented point, V_0 . In this case α is the perpendicular distance from point P_0 to the normal of the oriented point V_0 and β is the perpendicular distance from P_0 to the plane π .

Let us assume that the plane π is defined by the equation $Ax + By + Cz + D = 0$. Consider the oriented point V_0 on the plane, where $\bar{n}$ is the direction of the oriented point, V_0 , and a point P_0 close to V_0 . Then the perpendicular distance from the plane π to the point V_0 , is the distance β which given by:

$$d(P_0, \pi) = \beta = \frac{\bar{n} \cdot (V_0 - P_0)}{|\bar{n}|} = \frac{Ax_0 + By_0 + Cz_0 + D}{\sqrt{A^2 + B^2 + C^2}} . \quad (3.5-1)$$

The distance α , is the perpendicular distance from the normal $\bar{n}$ passing through P_0 , and is given by:

$$d(P_0, \bar{n}) = \alpha = \frac{-\bar{n} \cdot (P_0 - V_0)}{\bar{n} \cdot \bar{n}} = \frac{-d(P_0, \pi)}{|\bar{n}|} \quad (3.5-2)$$

The spin image is created by generating the α and β values of the surrounding data points in a small window around the oriented point (Figure 3.5-1). There is no rule to determine how many data points are to be sampled around the oriented point, but the more points are used, the more accurate the correlation will be.

Figure 3.5-2 Spin Image Generation

The Spin Image generation is outlined in Figure 3.5-2. Once the data points around the oriented points are chosen, the α and β values (Figure 3.5-1) are generated. From these α and β values, a 2D histogram of the data is created. Spin images derive their name from

the 2D histograms that can be used to render 2D grayscale images. The bin-size is 2 to 8 times the mesh resolution [JoH96]. The mesh resolution can be determined by calculating the average of the median edge lengths of all the 2.5D images. The average median edge length can be determined *a priori*.

Since the purpose of spin images is to correlate point pairs between images, it is necessary to generate spin images for all points in the correspondence set. There may be many spin images generated, and therefore they must be stored efficiently. Each spin image has a header that contains the following information:

1. The Spin point coordinate
2. The Average median edge length
3. The Histogram rows and columns
4. The Max and min α and β values

The 2D histogram can be converted into a 1D array and only non-zero bins are written to disk. With this format, an ASCII spin image file that has 2,500 data points has a size of less than 10 kb.

3.5.2 Comparing Spin Images

Two spin images from different data sets P and Q will not be exactly the same and therefore we need to have a method to correlate them. We expect that corresponding spin images will have similar spin image bins. The normalized linear correlation coefficient is used to compare spin image bins (equation 3.5-3) [JoH96].

$$R(P, Q) = \frac{N \sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N \sum p_i^2 - (\sum p_i)^2)(N \sum q_i^2 - (\sum q_i)^2)}} \quad (3.5-3)$$

In the above equation, N is the total number of bins, and p_i and q_i are the image bins from data sets P and Q respectively. The function R returns a value between zero and one, with one implying that the histograms are identical and zero signifying they are completely different.

3.6 Matching

Matching refers to the process of comparing point feature descriptors between all feature points of image V_1 to all the feature points of image V_2 . The input to the matching process are two descriptor lists D_1 and D_2 of the image views V_1 and V_2 respectively (Figure 3.6-1). The output is a probability matrix $\mathbf{M}$, that matches all combinations of D_1 and D_2 .

Figure 3.6-1 The matching procedures input and output.

The columns of the matrix $\mathbf{M}$ correspond to the descriptor list D_1 , and similarly, the rows of the matching matrix $\mathbf{M}$ corresponds to the descriptor list D_2 . An entry $\mathbf{M}(i,j)$ of $\mathbf{M}$ corresponds to the probability that the feature point i of image V_1 matches with that of feature point j of image V_2 .

3.6.1 The Matching Problem

The matching problem is solved for a pair of 3D images as follows:

1. Match all 2D point descriptors and generate the matching matrix $\mathbf{M}_1$
2. Match all 3D point descriptors and generate the matching matrix $\mathbf{M}_2$
3. Combine the results of $\mathbf{M}_1$ and $\mathbf{M}_2$

The 2D point descriptors generated by the colour correlation windows can be compared using one of the correlation methods, such as the least squares method. Although Section 3.4 outlines many different correlation methods, the least squares method, which is a simple and powerful way to compare the colour correlation windows [MiR01], was implemented. The 3D descriptors generated by the spin images can be compared using the normalized correlation coefficient as outlined in Section 3.5.2.

Descriptors that use geometry become less important when the surfaces are very planar, smooth, and homogeneous. The power of the 3D descriptor is due to the unique signature of the image surface that exists when the surface is heterogeneous. Similarly, colour correlation windows cannot be used to correlate features if the image surface is monochromatic. The power of the colour correlation window becomes evident when the image intensity values vary heterogeneously across the image surface. Our approach uses the best available 2D feature point generators in

conjunction with the best available 3D descriptor. By using both 2D and 3D features, the probability that the two feature points match, is increased.

3.6.2 Combining 2D and 3D descriptors

When comparing the colour correlation windows around a feature point in image V_1 to that of the colour correlation window of feature point in image V_2 , the result is a value between $[0,1]$. We say that $\mathbf{M}_1(i,j)$ is the probability that the keypoint i in image V_1 matches that of keypoint j of image V_2 . Similarly when we compare the spin image of keypoint i of V_1 with the keypoint j in V_2 the result is the matrix $\mathbf{M}_2(i,j)$. Therefore, $\mathbf{M}_1$ and $\mathbf{M}_2$ are the set of image *weights* generated using the 2D and 3D descriptors respectively. We combine the 2D and 3D descriptors by multiplying each entry of the matrix M as follows,

$$M = \begin{bmatrix} M_1(0,0)M_2(0,0) & . & . & . & M_1(m,0)M_2(m,0) \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ M_1(0,n)M_2(0,n) & . & . & . & M_1(m,n)M_2(m,n) \end{bmatrix} \quad (3.6-1)$$

A threshold μ can be used to eliminate poorly matched features, and so, if $\mathbf{M}(i,j) < \mu$ then set $\mathbf{M}(i,j) = 0$.

3.7 The Correspondence Problem

The correspondence problem attempts to identify matching feature points between image pairs. The input to the correspondence problem is the summary matrix $\mathbf{M}$ that was calculated by matching feature point descriptors. The correspondence method then determines the set of correspondences $\mathbf{C}$ (Figure 3.7-1).

Figure 3.7-1 The Correspondence methods input and output.

The correspondence problem can be recast as a complete weighted bipartite graph problem. Given a weighted bipartite graph $G = (V, E)$, determine the largest-size set of edges S from E such that each vertex in V is incident to at most one edge of S . The problem is bipartite since we have two sets of feature points, the graph is complete and weighted since we have results of the matched values, and it is matching because correspondences can only match one to one.

We now describe a greedy method that will generate the maximal result of the complete weighted bipartite graph problem. The function is called SORT_BOX and its input is a matrix $\mathbf{M}$ containing the weights of the graph. In the previous section we have combined the results of the colour correlation windows and the spin images. These results will act as the weights to the bipartite graph. There are m feature points of Image V_1 and there are

n feature points of image V_2 . To describe the algorithm we say that the weights of the complete weighted bipartite graph can be stored in a matrix $\mathbf{M}$ of size $m \times n$. An entry $\mathbf{M}(i,j)$ represents the probability that feature point i in Image V_1 matches that of feature point j in Image V_2 . The output is an $m \times 2$ matrix $\mathbf{C}$ representing the one-to-one correspondences of the feature points. For example, if $\mathbf{C}(k)=(i,j)$ then image feature i of V_1 matches with image feature j of V_2 . The details of the algorithm are given in Algorithm 3.5-1.

Algorithm 3.5-1 Recursive Matching algorithm that takes an $m \times n$ matrix $\mathbf{M}$ and an empty matrix $\mathbf{C}$ as input, the result is an $m \times 2$ matrix $\mathbf{C}$.

SORT_BOX($\mathbf{M}, \mathbf{C}$)

```

1      (m,n) = sizeOfMatrix(M);
2      if((m>0)&&(n>0))
3          # Recursive function that returns the index of the largest value in the matrix
3          (i,j) = indexOfMaxMinMatrixValue(M,"max");
4          C = push (i,j) onto C;
5          if((m>1)&&(n>1))
6              M = subtractRowAndColumn(M,i,j);
7              # Now recurse
8              C = SORT_BOX(M, C);
9          end if
10     end if
11     return C;
```

Algorithm 3.5-1 operates as follows. The **SORT_BOX** function is recursive and has two input parameters: (i) the weights of the graph $\mathbf{C}$, and (ii) the solution matrix $\mathbf{D}$. Initially $\mathbf{D}$

is empty. The function `indexOfMaxMinMatrixValue(C,"max")` identifies the largest value in the matrix **C** and returns its matrix index (i,j) ⁹. This matrix index (i,j) is the first and ‘best’ solution to the problem, and, is added to the solution matrix **D** in row 0. The algorithm then removes row *i* and column *j* from **C** using function `subtractRowAndColumn(C,i,j)`, and runs recursively on the new sub-matrix of **C**. The solutions are pushed as rows to the end of the solution matrix **D**. For a detailed example, of this please refer to Appendix E.

Note that algorithm 3.5-1 is a greedy algorithm [CLR94] since it chooses the largest value at every step. We do not obtain a *maximum* matching, but rather a *maximal* matching.

⁹ The pseudo code for the function is described in Appendix D.

3.8 Correspondence Filtering

Correspondence filtering refers to the process of determining the best three correspondences given the correspondence set C (Figure 3.8-1).

Figure 3.8-1 The correspondence list is filtered and reduced to the best three correspondences.

Once the best three correspondences have been determined the transformation that will register the two images together must be calculated. It is possible to register both images using only three correspondences of the 3D feature points. Let a_i be the set of correspondences in image A and b_i the set of correspondences in image B.

Figure 3.8-2 A set of Correspondences between image A and B.

If there are n correspondences of the 3D feature descriptors, using probability theory there are $\binom{n}{3}$ possible triangle combinations within each image¹⁰. The *triangle alignment algorithm* compares each triangle of image V_1 with the corresponding triangle of image V_2 . It is important to note that if we have true 3D data points then we know that distance and area are 3D invariants. In other words, each edge of the triangle (a_i, a_j, a_k) should be equal to each edge of the triangle (b_i, b_j, b_k) in length. Similarly the areas of the two matching triangles should be equal. The triangle comparison uses a combination of triangle area, length of the triangle edges, and the triangle centroid. The triangle pair that is most similar is used to register the two data sets.

The triangle alignment algorithm is given as follows:

1. Score each pair of $\binom{n}{3}$ possible 3D matching triangles according to their 3D geometry
2. Determine the triangle pair with the largest score and return the three correspondences of that triangle.

¹⁰ The combinations of choosing 3 items from a group of n , $\binom{n}{3}$, reduces to $\frac{n!}{(n-3)!3!} = \frac{1}{6} n(n-1)(n-2)$.

3.9 Triangle Alignment

Once the best three correspondences are determined, the triangle alignment algorithm uses those correspondences to calculate the transformation matrix that aligns image V_1 and V_2 (Figure 3.9-1).

Figure 3.9-1 The triangle alignment algorithm calculates the transformation matrix T computed from three correspondences.

Assume that we are given a set of three correspondences (c_1, c_2, c_3) where each $c_i = (a_i, b_i)$, and $(a_i, b_i) \in \mathbb{R}^3$. Note that a_i is a point in image V_1 and b_i is a point in image V_2 . Define triangle $A = (a_i, a_j, a_k)$ and triangle $B = (b_i, b_j, b_k)$. The steps of the algorithm are shown in Figure 3.9-2.

Figure 3.9-2 Triangle Alignment algorithm steps (i) to (iv).

The triangle alignment algorithm transforms triangle B to triangle A as follows:

1. Choose three points a_i , a_j , and a_k corresponding to b_i , b_j , and b_k
2. Translate b_i to a_i
3. Rotate b_j to a_j about the point a_i
4. Rotate b_k to a_k about the line $\overline{a_i a_j}$

Note that the step 2 uses translational transformation matrix, followed by a rotational transformation matrix (step 3), and followed by another rotational transformation matrix in the last step. The three transformation matrices are multiplied together to generate the resulting transformation matrix¹¹.

¹¹ For a detailed description of the basic transformation matrices refer to Appendix A.

3.10 Verifying the Transformation

The goal of this step is to verify the 3D transformation computed by the triangle alignment algorithm. The idea is to apply this transformation to all the 3D data points, which should align them properly. When the alignment is correct there will be a significant number of overlapping data points of the 2.5D images that are very close to each other. This is reasonable because we assume that each of these 2.5D range images have significant overlap because this is how they were acquired. Therefore, the verification algorithm simply counts the number of overlapping data points. We say that a pair of 3D data points, one from image A and the second from image B, overlaps if the distance between them is less than a given tolerance t . It should be noted that the 3D data does not have to be in any specific format and the point densities can vary. The inputs to the algorithm are two arrays A and B and a tolerance value t (Algorithm 3.10-1). The algorithm returns the number of overlapping data points. Note that the magnitude of the data sets A and B are denoted as $\|A\|$ and $\|B\|$ respectively.

Algorithm 3.10-1 **OVERLAP**(A, B, t) counts all the distances between elements of A, with the elements of B, that are greater than a tolerance value t.

OVERLAP(A, B, t)

```

1  count = 0;
2  for i = 1 to  $\|A\|$ 
3      for j = 1 to  $\|B\|$ 
4           $(a_x, a_y, a_z) = A(i)$  ;
5           $(b_x, b_y, b_z) = B(j)$  ;
6          if (  $\sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2} < t$  )
7              count = count + 1;
8          end if
9      end for j
10 end for i
11 Return( count / (( $\|A\|$ )( $\|B\|$ )) )

```

A reasonable value for the tolerance t is the average median edge length of neighboring data points of A and B multiplied by a user specified constant c. The constant c represents the accuracy of the registration. To make this verification step more efficient, it is possible to vary the size of the 3D data sets or to sample only part of the data.

4 Experiments and Results

The purpose of this Chapter is to present the implementation and results of our automatic 3D registration process. To achieve this goal the first section outlines the implementation and the latter sections describe the experiments. Three different registration methods are used to test the components of the registration procedure, they are:

- Manual Registration
- Semi-automatic Registration
- Automatic Registration

The goal of the exercise is to test:

- Matching methods
- 2D point features and descriptors
- 3D point descriptors
- Registration and Verification algorithms

The manual method is used to register the data sets for the semi-automatic and automatic registration procedure. The results of the manual registration are used as a base line ground-truth for comparison with the results of the semi-automatic and automatic registration procedures.

Manual registration refers to the process of choosing the feature points manually and also choosing the correspondences manually. Semi-automatic registration refers to the process of manually selecting feature points and automatically determining correspondences. Automatic registration refers to the process of automatically choosing feature points, determining the correspondences, and generating and applying a transformation matrix on a pair of images with the purpose of aligning the data sets.

4.1 Implementation Details

In this Thesis we implemented a large number of programs and libraries written in C and Perl/Tk that coded the automatic matching and correlation process. The first section explains the generation of descriptors, the second section explains the matching and correlation process, and the final section describes the implementation of the graphical user interface.

4.1.1 Generating Descriptors

Figure 4.1-1 shows the process of generating descriptors from the proprietary NRC EXT images. The first stage converted the EXT images to the popular PPM and VRML data formats storing the 2D colour and 3D data respectively. The features are then automatically generated returning an array of feature points. Descriptors are then generated from the list of feature points.

Figure 4.1-1 The process of generating descriptors from EXT images.

The main correlator function is able to call the data manager (datamngr) and the keypoint-chooser (keypointchooser) as shown in Figure 4.1-2. The data conversion process uses a data manager to convert EXT to PPM and VRML images, which explicitly calls the ext2img C program to do the conversion.

Figure 4.1-2 Programs and libraries used for the data conversion and feature point creation process. Note that programs denoted by (*) are existing programs and libraries written by other researchers.

The Thesis used a number of different EXT images such as the duck, totem pole and toupe. The filesAndDirs library saves and stores each set of images in a different directory.

The ext2img C-program uses two libraries: the PIL C-library¹² that can convert between different 2D file formats, and the NRC's library¹³ that can read the EXT data files. The libraries are denoted with (*) since they are not standard function calls, and they are existing programs written by other researchers. All other programs and libraries described were explicitly written by us for this Thesis.

Feature points are created using the keypointchooser function. The keypointchooser calls the keypoint generator that generates the Lowe features¹⁴. The findcorners program was used to generate the Harris and Susan feature points¹⁵. The appropriate output is then formatted and sent to the correlator program as an array of points.

Figure 4.1-3 shows the libraries used to generate feature point descriptors.

Figure 4.1-3 Libraries implemented to generate feature point descriptors.

¹² The PIL library was written by Anthony Whitehead and was used by permission [Whi04].

¹³ The NRC libraries to read the laser range images were used by the permission of the National Research Council.

¹⁴ The keypointchooser application was written by David Lowe [Low04] and was used by permission.

¹⁵ The findcorners application was used by permission and written by Anthony Whitehead [Whi04].

A list of feature points is sent to the spinImage library and a number of spin images are created and cached to disk. The spinImage library uses the simpleHistogram library as well as the simplePlane library to generate spin images. The colour correlation windows are generated using the colourCorrelation library, which calls the simpleColour library.

4.1.2 Matching and Correlation Implementation

Once the descriptors are generated the process matches each feature point. The feature point can be matched using spin images (correlation library), colour correlation windows (colourCorrelate library), or using both the spin images and the colour correlation windows (Figure 4.1.4).

Figure 4.1-4 Implementation of the Matching process.

To verify the spin images, we implemented the simpleVRML library. The colourCorrelate library was used in order to compare the colour correlation windows.

Once the matching is complete the correspondences are determined (sortBox library) and the transformation matrix computed using the registration library (Figure 4.1-5)

Figure 4.1-5 Libraries implemented to determine the correspondences and to generate the transformation matrix.

The correspondences are determined using the sortBox library which is an implementation of a greedy matching algorithm. The correspondences are pruned by the registration library, and the best three correspondences are used to compute the transformation matrix, T , with the use of the triangle library. This transformation matrix, T , is the solution to the automatic matching and correlation problem. The correlator program uses the simpleTransformation and simpleMatrix libraries to compute, store, and verify the transformation matrices.

4.1.3 The Graphical User Interface

The correlator program launches the graphical user interface (GUI) that can run the various routines contained in the libraries outlined in Section 4.1.1 and 4.1.2. Furthermore, the GUI can run in manual, semi-automatic, and automatic mode in order to provide flexibility in the testing phase¹⁶. The correlator GUI can vary the following parameters:

1. The data image pairs,
2. The type of feature point detector,
3. The number of feature points,
4. The correlation methods,
5. The correlation window size,
6. The matching threshold,
7. The manual, semi-automatic and automatic mode.

The GUI was written in Perl/Tk and its functions reside in the correlator library.

¹⁶ Please refer to Appendix H for a screen-capture of the correlator GUI.

4.2 Manual Registration

The manual registration technique requires a method to display two images side-by-side. The user can then manually choose correspondences between the two images. Given, a minimum of three pairs of correspondences, a transformation matrix can be calculated that will align the three selected points. Since the process of selecting the data points is prone to errors, it may be worth selecting additional correspondences (Figure 4.2-1), and then choosing the best transformation that will align the images.

Figure 4.2-1 Manual selection of six pairs of correspondences of the duck image.

Registering three points can be accomplished directly as outlined in Section 3.8. Using the triangle correlation method, we determine the best triangle match for all the possible sets of triangles generated by the correspondence pairs.

The data points are manually chosen so that every trial presents a possible unique set of correspondences. In practice, each set of correspondences can generate slightly different results. For this reason ten different experiments are performed to determine the actual overlap of each of the two data sets. The results are assumed to be the ground truth against which the semi-automatic and automatic experiments are evaluated. For these experiments, the assumption is that the computed percentage of 3D point overlap between the two images is correct and is the actual overlap percentage. The result of the manual registration experiment of the duck (Figure 4.2-1) is shown in Table 4.2-1.

Table 4.2-1 Image data specifications and registration results of the duck data set

Data Names	duck1 and duck2
Data Dimension	256 x 128
Number of Data Points	32,768
2D Data Size (ppm)	98 kb
3D Data Size (wrl)	1.35 Mb
Number of Correspondences used	6
True Overlap	29.00%
Samples used to determine overlap	2,048

By inspection it is possible to verify the registration and the overlap algorithm. Figure 4.2-2 shows the two duck images and their areas of overlap.

Figure 4.2-2 Manual identification of overlap of the duck image set

The areas of overlap identified by visual inspection are approximately 30%, and this agrees with the result calculated by the algorithm, namely 29.00%.

Figure 4.2-3 Two duck images are registered and displayed.

Figure 4.2-3 shows the result of the registration of the two duck images in a VRML browser. Both images seamlessly fit together even though the overlapping points have not been removed.

The summary of the manual registration process is as follows:

1. The user manually selects a number of correspondences,
2. The application automatically chooses the best three correspondences and calculates the transformation matrix,
3. The transformation matrix is applied and the data sets are compared to generate the percentage overlap,
4. Steps 1. to 3. are repeated with different correspondences and the largest overlap value is chosen as the base ground truth overlap,
5. The registered images are visually inspected to ensure correctness.

4.3 Semi-Automatic Registration

Semi-Automatic Registration attempts to automate the manual registration process. In the manual registration procedure a minimum of three points are selected, and the images are registered. Semi-automatic registration refers to the process of manually selecting feature points and automatically determining correspondences.

The semi-automatic registration experiments divide the automatic registration procedure into the following distinct parts:

- Feature Selection (Manual)
- Feature Correspondences (Automatic)

The following section will describe the quality of the automatic feature detectors and experimental results using colour correlation. The first section describes the view correspondence relationship of the range image data used in these experiments. The following sections test the automatic feature point selection and the colour correlation experiments.

4.3.1 View correspondences and overlap definitions

The experiments to test the automation of the registration process used a set of 2.5D images of a totem pole as shown in Figure 4.3-1.¹⁷

Figure 4.3-1 A set of five overlapping totem pole images.

The view correspondence mapping function of the five images of Figure 4.3-1 (a) to (e) are labeled views V^1 to V^5 respectively. The amount of overlap is calculated *a priori*, manually, as described in the previous section. Both the view correspondence mapping function and the amount of overlap are shown in Table 4.3-1.

Table 4.3-1 Amount of overlap between the five totem pole views V^1 to V^5 .

μ	1	2	3	4
$\alpha(\mu)$	V^1	V^2	V^3	V^4
$\beta(\mu)$	V^2	V^3	V^4	V^5
Overlap (%)	60.36	39.96	35.89	55.92

¹⁷ The images were generated by a laser range finder from the National Research Council of Canada's Visual Information Technology group.

4.3.2 Colour correlation experiments

The Colour Correlation experiment uses the intensity component of the 2.5D images, in combination with spin images, to identify correspondences. The testing application implemented for this experiment is able to read the 2D overlapping PPM images and display them side-by-side to the user. In this set of experiments the features were selected manually, while the correspondences were selected automatically. For testing purposes, it is possible to select a window size and the feature points manually. A circular colour correlation window is used to histogram and compare the colour information.

This experiment combines the colour correlation method with spin images. The first experiment will demonstrate how colour can be used to support the identification of a correspondence pair.

Figure 4.3-2 Spin Image matching (a), colour correlation (b), spin and colour (c).

Figure 4.3-3 shows four keypoints in image (a) circled in red and four keypoints circled in white. In this case each keypoint has a unique colour. The correspondences drawn in blue were determined using spin images. Note that the correspondences are not correct.

Image pair (b) shows the use of colour correlation matching with the same keypoints as in (a). Since the colours are very distinct, the colour matching was able to identify the correspondences correctly.

The third matching experiment combines the spin image and the colour correlation windows. The results of the matching are drawn using green lines in Figure 4.3-3 (c). From the figure we see that combining the spin image and colour correlation identifies the correspondences correctly.

Figure 4.3-3 Totem views V_1 and V_2 showing spin match (blue), colour match (red), and spin and colour results (green).

Figure 4.3-4 shows a spin image and colour correlation experiment using six keypoints in each image. The center of the circle indicates the keypoint location and the circle's area represents a correlation window of radius nine pixels. The blue line indicates the spin image keypoint match; notice none of them correlated properly. The red line indicates the colour correlation; four keypoints matched correctly. The green line shows the results of the combination of the spin image and colour correlation matching. The combined result gave five out of six possible correct matches. The 3D results of the pairwise registration are shown in Figure 4.3-5.

Figure 4.3-4 Registered VRML totem pole views V_1 and V_2 showing the side view (a) top view (b) and bottom view (c).

How is it possible, that there are zero correct spin image matches, four correct colour correlation matches, and the combination of the two resulted in five correct matches? The answer is that by summing the weights of the spin image and the colour correlation

results we are able to change the results of the complete weighted bipartite graph matching (see Section 3.7.1).

The strength of the colour correlation becomes evident when there are several unique colours present in the image. Figure 4.3-4 shows spin images that were clearly mismatched, but the similarity of the colour of the window around each keypoint strengthened the probability that two keypoints are valid correspondences.

4.4 Automatic Registration

The manual registration technique describes how to register two images and verify the success of the registration. This manual step is used as the ground-truth for the percentage of overlap of the two views. The semi-automatic step showed how to automate the feature selection and matching using colour correlation. Feature points and correspondences can be generated automatically, the appropriate transformation matrix calculated, and the data is registered and verified. Automatic registration will combine these methods, in the sense that both the feature points and their correspondences are found automatically.

The testing application implemented for the automatic registration experiment can dynamically vary the following parameters:

1. Number of feature points
2. Type of descriptors
3. Type of feature detectors
4. Correlation window size
5. Image data
6. Correspondence thresholds.

Any one of these parameters can affect the outcome of the automatic registration process.

4.4.1 View correspondences and overlap definitions

The automatic registration experiments use a set of toupe images as shown in Figure 4.4-1.

Figure 4.4-1 Four 2D views of the toupe image V_1 to V_4 shown as (a) to (d) respectively.

The amount of overlap between the views was determined manually as described in Section 4.2. The image views V_1 to V_4 correspond to the images in Figure 4.4-1 (a) to (d) respectively. The amount of overlap of the toupe images are given in Table 4.4-1.

Table 4.4-1 Amount of overlap between the four ‘toupe’ views V^1 to V^4 .

μ	1	2	3	4	5	6
$\alpha(\mu)$	V^1	V^2	V^3	V^1	V^2	V^1
$\beta(\mu)$	V^2	V^3	V^4	V^3	V^4	V^4
Overlap (%)	52.63	50.90	48.68	32.73	33.14	25.66

4.4.2 Correspondences and thresholds

The application implemented a solution to the complete weighted bipartite graph problem as outlined in Section 3.8. Each solution is ordered by the probability that the pair of data points is a valid match. It is up to the user to choose a threshold that will be high enough to guarantee accurate results, yet low enough to generate enough correspondences.

Figure 4.4-2 Toupe views V_3 and V_4 using a low threshold and 70 feature points.

The number of correspondences can be controlled using the number of feature points and thresholds. The thresholds can be applied to the probabilities of each correspondence pair. Given two types of images, 70 feature points per image, and a low threshold many correspondence pairs are generated as seen in Figure 4.4-2.

Figure 4.4-3 Correspondences generated from Lowe's feature points using the toupe views V_1 and V_2 and 40 feature points. (i) shows the feature points, the arrows indicate points that do not have a correspondence, and (ii) shows the correspondences.

Figure 4.4-3 shows five correspondence pairs generated from 40 feature points on the toupe views V_1 and V_2 after the threshold was applied. The threshold reduced the number of correspondence pairs from a possible 40 pairs to the 5 'best' pairs¹⁸. Note that the white arrows in Figure 4.4-3 (i) were added manually to indicate feature points that do not have a corresponding point in the image pair. These non-corresponding points increase the probability of an incorrect match. The correspondences in 4.4-3 (ii) were determined automatically.

¹⁸ For a complete and tabulated list of experiments please refer to Appendix G.

Figure 4.4-4 Three registered views (a) to (c) of a pair of toupe images.

Three of the five correlating pairs in Figure 4.4-3 are correct. The triangle correlation method was then able to automatically choose the best three sets of keypoints and the data was registered correctly as seen in Figure 4.4-4.

4.4.3 Automatic registration using spin images

The application also implemented the ability to automatically determine feature points and then determine correspondences using spin images. A 2D circular window around the 2.5D feature point is used to extract the geometric information for the spin image method.

The Spin Image algorithm follows the approach as described in Section 3.5. The 3D plane-fitting algorithm in Appendix B is used to calculate α and β values and the results are then compiled in a histogram. The Spin Image Object is able to process one or more feature points and extract the Spin Image Information. This information is stored in a text file and cached so that the results can be re-used. The Spin Image Object has functions to read and write Spin Images from and to file respectively.

The following set of experiments uses spin images to attempt to register the toupe data pair V_2 and V_3 . The experimental data is shown in Table 4.4.3.

Table 4.4.3 Data used for the automatic spin image experiment.

Toupe Data Pair	(V_2, V_3)
True Overlap	51.73%
Spin Image Window size	15
Colour Window size	N/A
Threshold	0.05 (normalization on)

Figure 4.4.5 plots the number of feature points for the three feature detectors against the resulting overlap percentage. The Lowe and Harris feature detectors generated overlaps

of close to 100% but the actual amount of overlap was 51.73%. This makes sense because of the symmetric nature of the toupe object.

Figure 4.4-5 Image view V_2 and V_3 of the toupe showing the feature points vs. the overlap of three feature detectors using only the spin image descriptor.

Figure 4.4.6 shows the accuracy given the actual overlap amount. The Lowe feature detector faired well in the 40-60 feature point range, although the registration was not correct.

Figure 4.4-6 Image view V_2 and V_3 of the toupe showing the feature points vs. the accuracy of three feature detectors using only the spin image descriptor.

This experiment showed that using only spin images it was not possible to accurately register the pair of images.

4.4.4 Comparing feature detectors

The following experiments use three different feature detectors on the toupe images: Harris, Susan, and Lowe. Each feature detector has its own strengths and weaknesses, and is designed to identify certain features in an image. The testing application is able to switch between feature detectors while holding the other parameters constant. In this manner, the performance of the feature detectors can be compared. The experimental data is shown in Table 4.4.4.

Table 4.4.4 Data used for the feature point comparison experiment.

Toupe Data Pair	(V_1, V_2)
True Overlap	52.63%
Spin Image Window size	15
Colour Window size	9
Threshold	0.7 (normalization off)

Figure 4.4-7 Image view V_1 and V_2 of the toupe showing the feature points vs. the overlap of three feature detectors.

Figure 4.4-7 shows the registration results¹⁹ of three feature detectors. The graph plots the number of feature points used against the computed percentage of overlap for views V_1 and V_2 of the toupe. The actual overlap result of the toupe views V_1 and V_2 (as shown in Table 4.4-4) is 52.63%. Why is it that the Harris feature detector gave an overlap amount close to 100% rather than the actual amount of 52.63%? This is because the toupe images are symmetrical in nature, and the registration scoring algorithm only verifies the geometry component. Therefore the geometry of the two images overlap seamlessly, but the intensity values do not.

¹⁹ The actual data values are tabulated in Appendix G.

Figure 4.4-8 Results of the registration of views V_1 and V_2 of the toupe image using three feature detectors.

Figure 4.4-8 shows the results of the registration of image views V_1 and V_2 graphed by its accuracy given the actual overlap amount. Note that the Harris feature detector generated results with approximately 100% overlap but the actual accuracy was just over 50%. The Lowe feature detector was able to register the data pair correctly between the range of 30-50 feature points. When the Harris feature point detector was set to 30 feature points, the registration was correct. The data pair was not correctly registered by using the Susan feature points. Given the above results, the Lowe feature detector was able to generate the most accurate and consistent results.

4.4.5 Varying the window size

This experiment varied the circular window size of the spin and colour correlation windows. The data used in the experiment is shown in Table 4.4.5.

Table 4.4.5 Data used for the window size experiment.

Toupe Data Pair	(V_1, V_2)
True Overlap	52.63%
Number of feature points	40
Colour & Spin used	Yes
Threshold	0.7 (normalization off)

Figures 4.4-9 and 4.4-10 show the results of varying the correlation window size of the colour, spin, and both colour and spin.

Figure 4.4-9 Toupe image V_1 and V_2 showing window size vs. the percentage overlap.

Figure 4.4-10 Toupe image V_1 and V_2 showing window size vs. percent accuracy.

After compiling the spin and colour results, the circular colour correlation window was fixed at size 9 pixels. Varying the Spin image window size did not affect the accuracy very much. We already stated that the toupe image is symmetrical, and therefore it can be expected that the spin images will not affect the registration as much as the colour descriptor. Recall that the spin image descriptors are dependent on the geometry and the colour on intensity. When varying the size of the colour correlation window, the spin image window size was fixed at 15. The accuracy approached 100% when the colour correlation window size was nine²⁰. The registration accuracy did not increase with the size of the correlation window, in fact the only accurate registration occurred when the correlation windows were small.

²⁰ Refer to the appendix G, Tables G31-G33, for a complete list of the windows size results.

5 Conclusion

The major contribution of the Thesis was the design and implementation of a novel automatic 3D registration system. The implementation allows the user to test many different aspects of the automatic registration process. The first part of the Thesis presented a literature survey of many 3D registration methods. The rest of the Thesis was devoted to solving the problem of automatic registration. Once the overall process was described in detail, the implementation issues of automatic correlation and matching, and their related experimentation was defined.

5.1 Summary

The Thesis began with a literature survey of 3D registration methods. The basic problem of 3D registration was defined and we explained in detail how the goal of the registration process affects the implementation strategy. Different registration, classification, and classical geometry problems were identified. The intensity and colour, direct, spin image, and automatic registration survey outlined many registration methods found in the literature.

Chapter 3 described our basic approach of Automatic Correlation and Matching. Automatic Correlation and Matching refers to the process of automatically matching feature points between an overlapping pair of colour 2.5D images. This correlation method is unique in the fact that it combined both 2D and 3D descriptors. The idea is that

the 3D descriptor can capture the 3D geometry of the object and the 2D descriptor can compute the local intensity.

Chapter 3 began by describing the format of the image data. The colour 2.5D images were acquired by the NRC using a synchronized colour laser range scanner. The output of the active sensor data is a dense set of 3D data points and co-located intensity data. The original EXT images were converted into its two component pieces, the 2D colour data was converted to a PPM image, and the 3D data was converted to the popular VRML format.

The automatic matching and correlation method can automatically generate 2D point features using the method of Harris, Susan, or Lowe. The 2D feature point detectors use invariant image features such as corners to determine keypoints within the images. The methods are important since without accurate feature points, there is no chance of determining accurate correspondences. The 2D and 3D feature point descriptor methods were presented, and the focus was on spin images and colour correlation windows respectively. The matching process compared all possible 2D and 3D descriptors of image V_1 , with all possible 2D and 3D descriptors of image V_2 . The matching results were stored in a matrix $\mathbf{M}$. The final correspondences were determined by solving the complete weighted bipartite graph matching problem using a greedy maximal algorithm on the matrix $\mathbf{M}$. It is possible to register a pair of 3D images using only three correspondences. The list of correspondences was pruned by comparing all possible triangle combinations of the correspondence set. Once the 'best' three correspondences

were found, they were used to generate the transformation matrix T , that aligned the two image views. The alignment was verified using an algorithm outlined at the end of Chapter 3.

Chapter 4 explained the implementation, experiments, and the results of the manual, semi-automatic, and automatic registration methods. Section 4.1 presented details of the software implementation. The code was written in a combination of C and Perl/Tk under the Linux operating system. The graphical user interface had many dynamic variables that enabled the testing of all major libraries of the system.

The experimental data sets were taken from a number of overlapping 2.5D range images of a duck, totem pole, and a toupe. The manual registration experiments were used as a baseline for the other experiments.

The semi-automatic experiments tested the colour correlation windows and found that colour can be used to increase the probability that a correspondence pair matches.

The automatic registration experiments demonstrated that it is possible to accurately register pairwise 2.5D images using the matching for correlation process. The experiments varied a number of parameters, such as the image data, correlation window sizes, descriptors, the number of feature points, and the correlation thresholds. Each set of experiments was tabulated and graphed. Each table and graph combination listed the results of 21 independent experiments.

It is important to set a correct threshold to limit the number of possible correspondences when running the automatic registration experiment. The more correspondences, the more likely it is that incorrect correspondences are determined. The number of correspondences can affect the speed of the matching and the pruning of correspondences. The quality of the correspondences can affect the accuracy of the final image alignment.

The automatic spin image results showed that it was not possible to accurately register certain 2.5D range images such as the toupe images using only the 3D spin image features. The Lowe feature detector was the most accurate of the 2D feature detectors, but it did not perform well enough to accurately align the 2.5D range images correctly. It was necessary to use a combination of spin images (3D features) as well as the corners (2D features) found in the intensity images. This proved to be the most successful approach.

The experiments tested the Harris, Susan, and Lowe feature detectors and found that the Lowe feature detector along with the spin images and the colour correlation windows was able to align the toupe images correctly by using a range of 30-50 feature points. The Harris feature detector, along with spin images and colour correlation windows, was able to align the toupe image accurately only when using 40 feature points. Finally, the Susan feature point detector, along with the spin images and colour correlation windows, was not able to align the toupe image accurately using a feature point range of 10-70 points.

The next set of experiments attempted to test how the size of the correlating window affected the overall alignment. A correct alignment was chosen and the colour and spin windows were varied. The result indicated that varying the window did not affect the quality of the registration much. The circular window size was varied in a range of 5 to 41 pixels in diameter.

5.2 Future Work

This Thesis showed that it is possible to accurately register 2.5D images by matching feature points. One possible improvement is to make the entire process more robust. Various camera and data errors should be identified and software measures implemented to limit these errors.

There is a lot of potential to improve on the time complexity of various phases of the process. Spin images are created per feature point, and their generation can be implemented in parallel. Since the method only pertains to a pair of images, a Multiple View Registration or a parallel- Multiple View Registration method can be implemented.

The overlaps created by different camera views can be very complex. Overlap configurations can be complex due to:

1. Extrusions
2. Occlusions
3. Complex camera transformations
4. Varying degrees of resolution

An interesting area of research would attempt to identify and classify different image overlaps. Identifying overlaps can potentially reduce the complexity of the problem from six-degrees of freedom to five or less degrees of freedom.

The registration verification algorithm can be improved to check how well the geometry and the intensity align. Some images are symmetrical, and so simply aligning geometry does not imply that the intensity data is aligned properly.

Feature detectors are used to determine specific features in an image. One improvement in the feature detection process is to automatically scan an image and score it for compatibility for a specific feature detector. Scanning an image for a specific feature descriptor would also be beneficial. For example, an image that is smooth and has few irregular surface features would not be a good candidate for spin images. If the same image has varying degrees of colour, it may be appropriate for a colour correlation method. Determining the compatibility of a specific descriptor for an image *a priori*, could then be used to determine the weight of the descriptor.

The triangle correlation method uses the best three correlation points to register the images. What happens if the best and the second best set of three correspondences are weighted equally or differ by a very small amount? An avenue for further research is that the triangle correlation method can be used as the basis of an ICP algorithm, iterating the result until an accurate registration is determined.

The “Bipartite Weighted Maximal Matching” problem and is a well-studied problem that has been solved in the literature. Different solutions to the Bipartite Weighted Maximal Matching problem will generate different correspondences. Different correspondences

will generate a different transformation matrix. A summary of the effect of different graph algorithms on the automatic registration problem would benefit the research area.

References

- [ABC03] M. Andreetto, N. Brusco, G. M. Cortelazzo, “Automatic 3D modeling of archaeological objects”, IEEE Workshop on Applications of Computer Vision in Archaeology, June 17, 2003.

- [AFR98] A. Ashbrook, R.B. Fisher, C. Robertson, N. Werghi, “Finding Surface Correspondences for Object recognition and Registration using Pairwise Geometric Histograms”, Proc. 5th Eur. Conf. on Computer Vision 1998, Freiburg, Germany, 1998.

- [AHB87] K. S. Arun and T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets", IEEE Transactions on Pattern Analysis and Machine Intelligence, number 5, pp. 698-700, Sept. 1987.

- [BeM91] P.J. Besl , N.D. McKay, “A Method for Registration of 3-D Shapes”, GM Research Laboratories, Computer Science Department, October 1, 1991.

- [BeR00] Bernardini, F., Rushmeier, H, “The 3D Model Acquisition Pipeline”, Eurographics Association, 2000.

- [BeS97] R. Benjemaa, F. Schmitt, "Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique", International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, May 12-15, 1997.
- [BhN98] D. N. Bhat, S. K. Nayar, "Ordinal Measures for Image Correspondence", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, number 4, pp. 415-423, 1998.
- [BLP95] R. Bergevin, D. Laurendeau, and D. Poussart, "Registering Range Views of Multipart Objects", Computer Vision and Image Understanding, Vol. 61, No.1, pp. 1-16, Jan. 1995.
- [BMR01] Bernardini, F., Martin, I.M., Rushmeier, H., "High-Quality Texture Reconstruction from Multiple Scans", IEEE Visualization and Computer Graphics, Vol. 7, No.4, pp. 318-332, Oct.-Dec. 2001.
- [BrL02] M. Brown, D. G. Lowe, "Invariant Features from Interest Point Groups", British Machine Vision Conference, BMVC 2002, Cardiff, Wales, pp. 656-665, Sept. 2002.
- [Bro92] L. G. Brown, "A Survey of Image Registration Techniques", ACM Computing Surveys, 24(4), 325 – 376, 1992.

- [CaC99] Daniel Camara, Mario Fernando Montenegro Campos, "A Methodology for 3D Registration of Range Images For Object Visualization", IEEE, 0-7695-0481-7/99, 1999.
- [CaH98] O. Carmichael, M. Hebert, "Unconstrained Registration of Large 3D Point Sets for Complex Model Building", Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Victoria B.C., Canada, pp. 360-367, Oct. 1998.
- [CHC99] C. Chen, Y. Hung, J. Cheng, "RANSAC-based DARCES: A New Approach for Fast Automatic Registration of Partially Overlapping Range Images", IEEE Transaction on Pattern Analysis and Machine Vision, Vol. 21, Number 11, Nov. 1999.
- [ChM92] Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Images", Image and Vision Computing, 10(3):145-155, Apr. 1992.
- [ChM95] Y. Chen and G. Medioni, "Description of Complex Objects from Multiple Range Images Using an Inflating Balloon Model", Computer Vision and Image Understanding, V. 61, no. 3, May 1995.
- [CLR94] T. H. Cormen, C. E. Leiserson, R.L. Rivest, "Introduction to Algorithms", MIT Press, pp. 329-355, 1994.

- [CWC96] Y. Cheng, V. Wu, R. Collins, A. Hanson, and E. Riseman, "Maximum-Weight Bipartite Matching Technique and Its Application in Image Feature Matching," Proc. SPIE Visual Comm. and Image Processing 1996, Orlando, FL, 1996.
- [DeT95] F. W. DePiero, M. M. Trivedi, "3-D computer vision using structured light: Design, calibration and implementation issues," Advances in Computers, vol. 43, pp. 243-278, 1995.
- [DWJ94] C. Dorai, J. Weng, A. K. Jain, "Optimal Registration of Multiple Range Views", 12th Proceedings IAPR International Conference on Pattern Recognition, Jerusalem, Israel, pp. 569 – 571, Oct. 1994.
- [DWJ97] C. Dorai, J. Weng, and A. K. Jain, "Optimal registration of object views using range data", IEEE Trans. Pattern Anal. Machine Intel., Vol. 19, No. 10, pp. 1131-1138, 1997.
- [ELB97] S. Elgazzar, R. Liscano, F. Blais, A. Miles, "3D Data Acquisition for Indoor Environment Modeling Using a Compact Active Range Sensor", IEEE Instrumentation and Measurement Technology Conference, Ottawa, Canada, May 19-21, 1997.

- [ELF97] D.W. Eggert, A. Lorusso, R.B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms", *Machine Vision and Applications*, 1997.
- [FaH86] O. D. Faugeras, M. Hebert, "The Representation, Recognition, and Positioning of 3D Shapes From Range Data", *Techniques for 3D Machine Perception*, INRIA North-Holland 1986, pages 13-51.
- [FMZ91] D. Forsyth, J.L. Mundy, A. Zisserman, "Invariant Descriptors for 3-D Object Recognition and Pose", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 10, Oct. 1991.
- [GoK93] A. Goldberg and R. Kennedy, "An efficient cost scaling algorithm for the assignment problem", Technical Report 93-1481, Department of Computer Science, Stanford University, 1993.
- [GRB94] G. Godin, M. Rioux, R. Baribeau, "Three-dimensional registration using range and intensity information", *SPIE Vol. 2350 Videometrics III*, pp. 279-290, Nov. 1994.

- [GSB94] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau, "Registration of multiple range views for automatic 3-d model building", Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 581-586, 1994.
- [Gue01] J. Guehring, "Reliable 3D Surface Acquisition, Registration and Validation using Statistical", Institute for Photogrammetry, university of Stuttgart, <http://citeseer.nj.nec.com/guehring01reliable.html>, Nov. 1994.
- [HaS88] C. Harris and M. Stephens, "A combined corner and edge detector", Proceedings of the 4th ALVEY vision conference, pp. 147-151, University of Manchester, England, Sep. 1988.
- [HDD92] H. Hoppe, T. DeRose, T. Duchamp, J. A. McDonald, W. Stuetzle, "Surface Reconstruction from Unorganized Points", Journal of Computer Graphics 1992, Vol. 26, number 2, pp. 71-78, 1992.
- [HLS02] D. Hall, B. Leibe, and B. Schiele, "Saliency of Interest Points under Scale Changes", Proceedings of British Machine Vision Conference 2002, Cardiff, GB, Sep. 2-5, 2002.

- [Hop96] H. Hoppe, "Progressive Meshes", Journal of Computer Graphics 1996, Vol. 30, pp. 99-108, 1996.
- [JoH96] A.E. Johnson, M. Hebert, "Recognizing Objects by Matching Oriented Points", Proc. CVPR97, pp. 684-689, 1997.
- [JoH97] A.E. Johnson, M. Hebert, "Surface Registration by Matching Oriented Points", International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, May 12-15, 1997.
- [JoH99] A.E. Johnson, M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 5, pp. 433 – 449, May, 1999.
- [JoK97] A.E. Johnson, S.B. Kang, "Registration and Integration of Textured 3-D Data", International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, May 12-15, 1997.

- [Jok00] Olli Jokinen, "Matching and Modeling of Multiple 3-D Disparity and Profile Maps", ACTA Polytechnica Scandinavica, Mathematics and Computing Series, No. 104, Mar. 2000.
- [JoO00] M. Jones, J.P. Oakley, "Registration of Image Sets using Silhouette Consistency", VISP(147), No. 1, page 1, Feb. 2000.
- [Low03] D. Lowe, "Distinctive image features from scale-invariant keypoints", submitted to International Journal of Computer Vision, Jun. 2003.
- [Low04] D. Lowe, Research Homepage, <http://www.cs.ubc.ca/spider/lowe/>, University of British Columbia, Computer Science, Apr. 2004.
- [LUC00] J. Luck, C. Little, and W. Hoff, "Registration of Range Data Using a Hybrid Simulated Annealing and Iterative Closest Point Algorithm", Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, Apr. 2000.
- [MaY95] T. Masuda, N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images", Computer Vision and Image Understanding, Vol. 61, No. 3, pp. 295-307, May 1995.

- [MiR01] A. Miles, G. Roth, "Colour Correlation Window Matching using Colour Spaces", Carleton Journal of Computer Science No. 5, Apr. 2001.
- [MiS03] K. Mikolajczyk, C. Schmid, "A performance evaluation of local descriptors", Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference, pp. 257- 263, 18-20 Jun. 2003.
- [Mor80] H. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover", Tech Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute, Sept. 1980.
- [MSY96] T. Masuda, K. Sakaue, N. Yokoya, "Proceedings of the 13th International Conference on Pattern Recognition, vol. 1 , pp. 879 –883, 1996.
- [NGU99] V. Nguyen, V. Nzomigni, Stewart, C.V., "Fast and Robust Registration of 3D Surfaces Using Low Curvature Patches", Second International Conference on 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, Oct. 4-8, 1999.
- [Pul97] K. Pulli, "Surface Reconstruction and Display from Range and Colour Data", Ph.D. Thesis, University of Washington, 1997.

- [Pul99] K. Pulli, "Multiview Registration for Large Data Sets", Second International Conference on 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, Oct. 4-8, 1999.
- [Rio94] M. Rioux, "Digital 3-D imaging: theory and applications", SPIE Proceedings, Videometrics III, International Symposium on Photonic and Sensors and Controls for Commercial Applications, Boston, MA, Vol. 2350, pp. 2-15, NRC 38343, Oct. 31 – Nov. 4, 1994
- [RoL92] G. Roth, M.D. Levine, "Minimal subset random sampling for pose determination and refinement", Advances in Machine Vision: Strategies and Applications, World Scientific, Singapore, pp. 1-21, 1992.
- [Rot99] G. Roth, "Building models from sensor data, an application shared by the vision and graphics community", in Nato Meeting: Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia, pp. 87-98, Aug. 1999.
- [Rot99-2] G. Roth, "Registering two overlapping range images", In Second International Conference on Recent Advances in 3D Imaging and Modeling, Ottawa, Canada, pp. 191-200, Oct. 1999.

- [RoW95] G. Roth, E. Wibowo, "A Fast Algorithm for making mesh-models from multiple-view range data", In Proceedings of the 1995 Robotics and Knowledge Based Systems Workshop, Canadian Space Agency, St. Hubert, P.Q., pp. 349-355, Oct. 1995.
- [RoW97] G. Roth, E. Wibowo, "An Efficient volumetric method for building closed triangular meshes from 3-D image and point data", In Graphics Interface 97, Kelowna, B.C., Canada, pp. 173-180, May 1997.
- [SDA00] Y. Sun, C. Dumont, M. A. Abidi, "Mesh-based integration of range and color images," in Proc. SPIE AeroSense, The International Society for Optical Engineering, vol.4051, pp.110-117, 2000.
- [SiB97] Gilles Simone, Marie-Odile Berger, "A two-stage robust statistical method for temporal registration from features of various type", INREA research report No 3235, Aug. 1997.
- [SLH98] A.J. Stoddard, S. Lemke, A. Hilton, T. Renn, "Estimating pose uncertainty for surface registration", Image Vision Computing 16, pp. 111-120, 1998.
- [SmB95] S. M. Smith, J. M. Brady, "SUSAN – A New Approach to Low Level Image Processing", Technical Report TR95SMS1c, Chertsey, Surrey, UK, 1995.

- [SzC94] Richard Szeliski, James Coughlan, "Spline-Based Image Registration", Digital Equipment Corporation Technical Report Series, Apr. 1994.
- [TBD02] J. Taylor, J-A Beraldin, G. Godin, L. Cournoyer, M. Rioux, J. Domey, "NRC 3D Imaging Technology for Museums & Heritage", Proceedings of the First International Workshop on 3D Virtual Heritage, Geneva, Switzerland, pp. 70-75, Oct. 2-3, 2002.
- [Wei97] S. Weik, "Registration of 3-D Partial Surface Models Using Luminance and Depth Information", International Conference on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Ontario, Canada, IEEE Computer Society Press, May 12-15, 1997.
- [Whi04] Anthony Whitehead, Research Homepage, <http://vision.scs.carleton.ca/~awhitehe/>, Carleton University, School of Computer Science, Apr. 2004.
- [WiB00] J. Williams, M. Bennamoun, "Simultaneous Registration of Multiple Corresponding Point Sets", Space Center for Satellite Navigation, Brisbane Australia, Jan. 31, 2000.

- [ZHA94] Z. Zhang, "Iterative Point Matching for Registration of Free-form Curves and Surfaces", *International Journal of Computer Vision*, 13(2), pp. 119-152, 1994.
- [ZiT98] D. Ziou and S. Tabbone, "Edge Detection Techniques - An Overview", *International Journal of Pattern Recognition and Image Analysis*, Vol. 8, pp. 537-559, 1998.

Appendices

Appendix A: 3D Transformation Matrices

The 3D translation transformation matrix translating by an amount (a,b,c) in the directions (x,y,z) is given by:

$$\text{Trans}(x,y,z)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{bmatrix} \quad (\text{A1})$$

The 3D rotation transformations about the x, y, and z axes by an angle θ , β , ϕ respectively are given by:

$$\text{Rot}(x,\theta)=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A2})$$

$$\text{Rot}(y,\beta)=\begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A3})$$

$$\text{Rot}(z,\phi)=\begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A4})$$

The general rotation matrix T is given by multiplying A1 to A4 (equation A5):

$$T = \begin{bmatrix} \cos(\beta) \cos(\phi) & -\cos(\beta) \sin(\phi) & \sin(\beta) & 0 \\ \sin(\theta) \sin(\beta) \cos(\phi) + \cos(\theta) \sin(\phi) & -\sin(\theta) \sin(\beta) \sin(\phi) + \cos(\theta) \cos(\phi) & -\sin(\theta) \cos(\beta) & 0 \\ -\cos(\theta) \sin(\beta) \cos(\phi) + \sin(\theta) \sin(\phi) & \cos(\theta) \sin(\beta) \sin(\phi) + \sin(\theta) \cos(\phi) & \cos(\theta) \cos(\beta) & 0 \\ a & b & c & 1 \end{bmatrix}$$

In this manner *six degrees of freedom* can be stored in a single 4x4-transformation matrix. Note that it is possible to multiply A1 and A4 in a different order and it will generate a different general transformation matrix T .

Appendix B: 3D Plane Fitting

This section will solve the problem of fitting a number of arbitrary 3D points to a 3D plane.

Given a set of n 3D data points $P = \{(x_1, y_1, z_1), \{(x_2, y_2, z_2) \dots \{(x_n, y_n, z_n) \}$, let $\bar{v}_i$ be the normal to the plane of best fit $Ax + By + Cz + D = 0$ that passes through the point (x_i, y_i, z_i) . We are trying to minimize the sum of the perpendicular distances from $Ax + By + Cz + D = 0$ to (x_i, y_i, z_i) .

We can plug in each (x_i, y_i, z_i) to the equation $Ax + By + Cz + D = 0$ to form n equations.

Re arranging,

$$Ax + By + Cz + D = 0 \quad (B1)$$

we get:

$$Ax + By + D = -Cz \quad (B2)$$

$$\frac{A}{-C}x + \frac{B}{-C}y + \frac{D}{-C} = z \quad (B3)$$

Which can be re-written by substituting new constants A' , B' , and D' as follows:

$$A'x + B'y + D' = z \quad (B4)$$

When we substitute the values of the n data points (x_i, y_i, z_i) we get,

$$A'x_1 + B'y_1 + D' = z_1$$

$$A'x_2 + B'y_2 + D' = z_2$$

...

$$A'x_n + B'y_n + D' = z_n$$

The above can be written in the form:

$$M\bar{x} = \bar{b}, \quad (\text{B5})$$

$$\text{where, } M = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \dots & & \\ x_n & y_n & 1 \end{bmatrix}, \quad (\text{B6})$$

$$\bar{b} = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_n \end{bmatrix}, \quad (\text{B7})$$

$$\text{and } \bar{x} = \begin{bmatrix} A' \\ B' \\ D' \end{bmatrix}. \quad (\text{B8})$$

Working with equation (B5), we get

$$M^T M\bar{x} = M^T \bar{b}. \quad (\text{B9})$$

If the kernel of M is trivial, then the kernel of $M^T M$ is trivial which means that $M^T M$ can be inverted. Then,

$$(M^T M)^{-1}(M^T M)\bar{x} = (M^T M)^{-1} M^T \bar{b} \quad (\text{B10})$$

$$\bar{x} = (M^T M)^{-1} M^T \bar{b} \quad (\text{B11})$$

From Equation (B8) and equation (B11) we can solve for A', B' , and D' and from equation (B4), $C' = I$ to give rise to,

$$A'x + B'y + z + D' = 0 \quad (\text{B12})$$

Note that Equation (B9) is the normal equation for Equation (B5), and therefore the least squares solution of (B5) is (B11) such that (B12) is closest to $\bar{b}$ from all other $M\bar{x}$.

Appendix C: Combining descriptors example

The results of the colour correlation and spin images are tabulated in Table C-1 for three keypoints in each image. The value in the cell (i,j) represents the probability p that keypoint i in image 1 matches that of keypoint j of image 2 with²¹.

Table C-1 Colour correlation results (left), spin image results (middle), and the combined results (right).

	0	1	2
0	0.32	0.10	0.25
1	0.65	0.15	0.20
2	0.03	0.75	0.22

	0	1	2
0	0.74	0.05	0.21
1	0.05	0.25	0.70
2	0.21	0.70	0.09

	0	1	2
0	0.23	0.01	0.05
1	0.03	0.04	0.14
2	0.01	0.53	0.02

The combined results of the colour correlation and the spin images are the product of their respective cells. A threshold and normalization help identify minimum and maximum values (Table C-2).

²¹ Matching the colour correlation windows and spin images generated the probabilities p .

Table C-2 Results after applying the threshold (left), normalizing (middle), and the threshold being applied to the normalized results (right).

$t=0.1$	0	1	2
0	0.23	0	0
1	0	0	0.14
2	0	0.53	0

norm	0	1	2
0	0.79	0.03	0.18
1	0.14	0.19	0.67
2	0.01	0.95	0.04

$t=0.7$	0	1	2
0	0.79	0	0
1	0	0	0
2	0	0.95	0

Table C-2 left shows the implementation of a minimum threshold on the combined result. Values lower than the threshold 0.1 are set to zero. Table C-2 middle normalizes the combined result of C-1 (right). This time a maximum threshold of 0.7 is used to generate the results of C-2 right. Note that the value of 0.67 in cell (2,1) of C-2 (middle) is regarded as being too low to be used as an accurate result.

Appendix D: Maximum and Minimum values of a matrix

Algorithm D We present a recursive method that determines the maximum or minimum value of a matrix and returns its index. The input is an $m \times n$ matrix **M** and a variable *sort* that is either set to max or min. The output is and an index pair (i,j) referencing the maximal value in **M**.

indexOfMaxMinMatrixValue(M,sort);

```

1      (m,n) = sizeOfMatrix(M);
2      if(m == 1)
3          i = 1;
4          j = indexOfLargestValue(M);
5      else if(n == 1)
6          i = indexOfLargestValue(M);
7          j = 1;
8      else
9          A = convertToArray(M);
10         # S holds the array values and I the index into the array
11         if(sort == "max")
12             (S,I) = sortArrayByMaxValue(A);
13         else if (sort == "min")
14             (S,I) = sortArrayByMinValue(A);
15         end if
16         # R is an array of row indices that have the maximum value, add the max row index
17         R[0] = I[0];
18         # Check if there are ties with the largest value and add their row indices
19         for( a=1 to sizeOf(A) )
20             # Check if max values and column numbers are equal

```

```

21         if((S[a-1] == S[a]) && (I[0]%n == I[a]%n))
22             R[a]=I[a];
23         else
24             a=sizeOf(A)
25         end if
26     end for
27     # If there is a tie then there will be multiple row indices present
28     if(sizeof(R) > 1)
29         # Determine the sub matrix; keep the row of R[i] and remove value R[i].
30         M=subMatrixUsingArrayIndex(M,R);
31         # Recursion Step
32         (i,j) = indexOfMaxMinMatrixValue(M,"min");
33     end if
34 end if
35 return (i,j);

```

Please refer to Appendix E for an example that uses the above algorithm.

Appendix E: Weighted Bipartite Graph Matching Example

We will demonstrate the maximal weighted bipartite graph matching solution with the use of an example. Consider a complete 5x4 bipartite graph as shown in Figure E-1.

Figure E-1 Complete Bipartite Graph.

We will define the weights as a 5x4 matrix, the rows index feature points I, and the columns index the feature points J. The left hand side (L.H.S.) are the weights and the right hand side (R.H.S.) is the index into the L.H.S.

0.91	0.38	0.96	0.71	0	1	2	3
0.87	0.91	0.96	0.36	0	1	2	3
0.36	0.91	0.12	0.36	0	1	2	3
0.36	0.91	0.38	0.40	0	1	2	3
0.87	0.12	0.08	0.01	0	1	2	3

We first sort the rows by the largest number. Note that 0.96 is the largest number and it is a tie between rows 0 and 1. The algorithm must decide whether to choose 0.96 from row 0 or from row 1.

0.96	0.91	0.71	0.38	2	0	3	1
0.96	0.91	0.87	0.36	2	1	0	3
0.91	0.36	0.36	0.12	1	0	3	2
0.91	0.40	0.38	0.36	1	3	2	0
0.87	0.12	0.08	0.01	0	1	2	3

In case of a tie we sort each row of the sub-matrix and look for the largest number of each row. The sub-matrix is defined by the tied numbers, as shown by the dotted lines above.

0.91	0.71	0.38	0	1	2
0.91	0.87	0.36	0	1	2

The largest numbers of each row are tied again. We recursively repeat this until there are no more ties.

0.71	0.38	0	1
0.87	0.36	0	1

We then look at the largest numbers of each row and choose the smallest number, that is 0.71. Therefore we return column 0 from the recursion process. This identifies row 0 and column 2.

0.91	0.38	0.96	0.71
0.87	0.91	0.96	0.36
0.36	0.91	0.12	0.36
0.36	0.91	0.38	0.40
0.87	0.12	0.08	0.01

0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3
0	1	2	3

The algorithm returns (0,2) as being the maximal index to the value 0.96 in the matrix.

We then remove row 0 and column 2 and generate the sub-matrix as shown below:

0.87	0.91	0.36
0.36	0.91	0.36
0.36	0.91	0.40
0.87	0.12	0.01

0	1	2
0	1	2
0	1	2
0	1	2

This is repeated recursively until all the solutions are generated.

The sorted set of largest values are on the L.H.S. and the original matrix indices are on the R.H.S.

0.96	2	0
0.91	1	2
0.87	0	4
0.40	3	3

This results in the solution shown in Figure E-2.

Figure E-2 Solution to the bipartite graph problem.

Appendix F: RGB to HSV Conversion

C Code that converts RGB (Red, Green, Blue) values to HSV (Hue, Saturation, Value).

RGB values are in the range [0,1], H values are in the range [0,360] , S and V values are in the range [0,1] except when S equals 0 then H is undefined.

```
void rgb_hsv(double r, double g, double b, double *h, double *s, double *v)
{
    double maxmin3(int,double,double,double);
    double max,min,delta;

    max = maxmin3(1,r,g,b);
    min = maxmin3(0,r,g,b);
    delta = max - min;
    /* v value is max of rgb */
    *v = max;
    /* s saturation is calculated */
    *s = (max != 0.0) ? (delta / max) : 0.0;
    /* h hue is calculated */
    if(*s == 0.0)
        *s = UNDEFINED;
    else
    {
        if(r == max)
            *h = (g - b) / delta;    /* some yellow magenta combination */
        else if(g == max)
            *h = 2.0 + (b - r) / delta;    /* some cyan and yellow combination */
        else if(b == max)
            *h = 4.0 + (r - g) / delta;    /* some magenta cyan combination */
        *h *= 60.0;    /* convert hue to degrees */
        if(*h < 0.0)    /* non-negative check */
            *h += 360.0;
    }
}
```


Appendix G: Automatic Registration Results

The first set of 30 experiments using spin images and colour correlation windows varied the number of feature points, the feature detectors, and calculated the overlap of the toupe images V_1 and V_2 . The experimental parameters are given in Table G1 and the results are listed in Table G2-G4.

Table G1 – Registration parameters for the results of Tables G2-G4.

Toupe Data Pair	(V_1, V_2)
Actual overlap	52.63%
Spin Image Window size	15
Colour Window size	15
Threshold	0.75 (normalization off)

Table G2,G3,G4 - Spin Image and Colour Correlation results for the Harris (left), Susan (middle), and Lowe (right) feature detectors which are Tables G2, G3, and G4 respectively, showing number of feature points, amount of overlap and Accuracy.

Pts	Overl.	Acc.
10	0	0.00
20	0	0.00
30	0	0.00
40	56.58	93.02
50	55.35	95.09
60	99.34	52.98
70	99.34	52.98
80	99.34	52.98
90	99.34	52.98
100	99.34	52.98

Pts	Overl.	Acc.
10	0	0.00
20	0	0.00
30	0	0.00
40	0	0.00
50	78.54	67.01
60	39.47	75.00
70	20.73	39.39
80	66.12	79.60
90	0	0.00
100	0	0.00

Pts	Overl.	Acc.
10	0	0.00
20	0	0.00
30	98.68	53.33
40	98.68	53.33
50	98.68	53.33
60	98.68	53.33
70	98.68	53.33
80	98.68	53.33
90	98.68	53.33
100	98.68	53.33

The next set of 30 experiments using spin images and colour correlation windows varied the number of feature points, the feature detectors, and calculated the overlap of the toupe images V_2 and V_3 . The experimental parameters are given in Table G5 and the results are listed in G6-8.

Table G5 – Registration parameters for the results of Tables G6-G8.

Toupe Data Pair	(V_2, V_3)
Actual overlap	50.9%
Spin Image Window size	9
Colour Window size	9
Threshold	0.75 (normalization off)

Table G6,G7,G8 - Spin Image and Colour Correlation results for the Harris (left), Susan (middle), and Lowe (right) feature detectors which are Tables G6, G7, and G8 respectively, showing number of feature points, amount of overlap and Accuracy.

Pts	Overl.	Acc.
10	0.00	0.00
20	0.00	0.00
30	17.60	34.58
40	97.37	52.27
50	97.37	52.27
60	89.39	56.94
70	98.44	51.71
80	98.44	51.71
90	98.44	51.71
100	98.44	51.71

Pts	Overl.	Acc.
10	0.00	0.00
20	9.38	18.43
30	38.90	76.42
40	9.38	18.43
50	9.38	18.43
60	25.58	50.26
70	10.86	21.34
80	94.24	54.01
90	26.40	51.87
100	62.50	81.44

Pts	Overl.	Acc.
10	0	0.00
20	0	0.00
30	0	0.00
40	0	0.00
50	92.19	55.21
60	92.19	55.21
70	92.19	55.21
80	92.19	55.21
90	35.94	70.61
100	25.99	51.06

The following set of 30 experiments tested the next pair of toupe images V_3 and V_4 .

The experimental parameters are given in Table G9 and the results are listed in G10-G12.

Table G9 – Registration parameters for the results of Tables G10-G12 of toupe views V_3 and V_4 .

Toupe Data Pair	(V_3, V_4)
Actual overlap	48.68%
Spin Image Window size	9
Colour Window size	9
Threshold	0.05 (normalization on)

Table G10,G11,G12 - Spin Image and Colour Correlation results for the Harris (left), Susan (middle), and Lowe (right) feature detectors which are Tables G10, G11, and G12 respectively, showing number of feature points, amount of overlap and Accuracy.

Pts	Overl.	Acc.
10	22.12	45.44
20	41.86	85.99
30	54.28	89.68
40	62.83	77.48
50	62.83	77.48
60	62.83	77.48
70	22.94	47.12
80	22.94	47.12
90	22.94	47.12
100	22.94	47.12

Pts	Overl.	Acc.
10	20.81	42.75
20	66.78	72.90
30	33.63	69.08
40	97.53	49.91
50	21.22	43.59
60	24.34	50.00
70	63.90	76.18
80	90.05	54.06
90	66.94	72.72
100	57.57	84.56

Pts	Overl.	Acc.
10	9.95	20.44
20	9.95	20.44
30	40.54	83.28
40	40.54	83.28
50	40.54	83.28
60	93.59	52.01
70	93.59	52.01
80	0	0.00
90	0	0.00
100	0	0.00

The next 63 experiments tested the toupe images using a spin image window of 15 pixels and a colour correlation window of 15 pixels and are tabulated in Tables G13-G21. The accuracy was determined by dividing the overlap amount by the actual overlap amount as shown in Table 4.3-1.

Table G13, G14, G15 – Toupe image views V_1 and V_2 using a threshold of 0.7 and the normalization turned off showing the number of feature points vs. overlap and accuracy, Table G13, G14, and G15 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	31.83	60.48
20	10.53	20.01
30	10.53	20.01
40	52.47	99.70
50	99.59	52.85
60	99.84	52.71
70	99.84	52.71

Pts	Overl	Acc
10	0	0.00
20	0	0.00
30	73.77	71.34
40	35.94	68.29
50	29.36	55.79
60	23.36	44.39
70	42.35	80.47

Pts	Overl	Acc
10	0	0.00
20	0	0.00
30	51.73	98.29
40	52.63	100.00
50	52.63	100.00
60	49.84	94.70
70	41.94	79.69

Table G16, G17, G18 – Toupe image views V_2 and V_3 using a threshold of 0.05 and the normalization turned on showing the number of feature points vs. overlap and accuracy, Table G16, G17, and G18 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	23.68	45.78
20	65.54	78.93
30	22.94	44.35
40	96.55	53.58
50	92.19	56.11
60	94.08	54.99
70	73.11	70.76

Pts	Overl.	Acc.
10	31.66	61.20
20	26.73	51.67
30	52.80	97.97
40	27.88	53.90
50	25.33	48.97
60	69.33	74.61
70	69.57	74.36

Pts	Overl.	Acc.
10	9.46	18.29
20	30.26	58.50
30	54.69	94.59
40	15.46	29.89
50	15.46	29.89
60	95.97	53.90
70	95.97	53.90

Table G19, G20, G21 – Toupe image views V_3 and V_4 using a threshold of 0.02 and the normalization turned on showing the number of feature points vs. overlap and accuracy, Table G19, G20, and G21 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	21.55	44.27
20	75.83	64.20
30	21.30	43.76
40	24.67	50.68
50	26.32	54.07
60	23.60	48.48
70	65.13	74.74

Pts	Overl.	Acc.
10	30.18	62.00
20	41.28	84.80
30	59.13	82.33
40	41.20	84.63
50	29.85	61.32
60	25.74	52.88
70	23.19	47.64

Pts	Overl.	Acc.
10	11.84	24.32
20	9.95	20.44
30	9.95	20.44
40	55.67	87.44
50	49.75	97.85
60	40.54	83.28
70	97.53	49.91

The next 63 experiments tested the toupe images by setting the spin image window size equal to the colour correlation window size at 15 pixels. The results of the experiments are tabulated in Tables G22-G30. As in the previous examples the accuracy was determined by dividing the overlap amount by the actual overlap amount as shown in Table 4.3-1.

Table G22, G23, G24 – Toupe image views V_1 and V_2 using a threshold of 0.7 and the normalization turned off showing the number of feature points vs. overlap and accuracy, Table G22, G23, and G24 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	23.85	45.32
20	25.08	47.65
30	23.60	44.84
40	98.52	53.42
50	98.85	53.24
60	99.84	52.71
70	99.84	52.71

Pts	Overl.	Acc.
10	0.00	0.00
20	60.77	86.61
30	44.08	83.75
40	38.90	73.91
50	34.21	65.00
60	31.50	59.85
70	38.90	73.91

Pts	Overl.	Acc
10	43.17	82.03
20	56.58	93.02
30	99.67	52.80
40	98.68	53.33
50	99.84	52.71
60	99.89	52.69
70	14.23	27.04

Table G25, G26, G27 – Toupe image views V_2 and V_3 using a threshold of 0.05 and the normalization turned on showing the number of feature points vs. overlap and accuracy, Table G25, G26, and G27 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	22.94	44.35
20	23.68	45.78
30	35.03	67.72
40	25.99	50.24
50	99.92	51.77
60	18.83	36.40
70	26.56	51.34

Pts	Overl.	Acc.
10	27.63	53.41
20	18.67	36.09
30	46.05	89.02
40	25.41	49.12
50	29.36	56.76
60	14.31	27.66
70	83.47	61.97

Pts	Overl.	Acc.
10	20.72	40.05
20	99.59	51.94
30	9.54	18.44
40	47.86	92.52
50	47.86	92.52
60	47.86	92.52
70	17.60	34.02

Table G28, G29, G30 – Toupe image views V_3 and V_4 using a threshold of 0.02 and the normalization turned on showing the number of feature points vs. overlap and accuracy, Table G28, G29, and G30 use the Harris, Susan, and Lowe feature detectors respectively shown below from left to right.

Pts	Overl.	Acc.
10	19.41	39.87
20	27.71	56.92
30	79.93	60.90
40	21.63	44.43
50	15.87	32.60
60	27.96	57.44
70	64.14	75.90

Pts	Overl.	Acc.
10	28.54	58.63
20	21.46	44.08
30	94.74	51.38
40	18.01	37.00
50	65.38	74.46
60	27.38	56.24
70	28.29	58.11

Pts	Overl.	Acc.
10	7.07	14.52
20	19.74	40.55
30	97.86	49.74
40	91.53	53.18
50	12.75	26.19
60	9.05	18.59
70	97.53	49.91

Tables G31-G33 show the results of the window size experiments using the toupe images V_1 and V_2 . The following 33 experiments used a threshold of 0.7 and 40 feature points since those values generated accurate results in the previous examples. The idea was to test varying window sizes for the spin images and colour correlation windows. Table G31 shows the results when the spin image window equals the colour correlation window and was set at 15 pixels, Table G32 shows the results when the spin image windows was fixed at 15 pixels, and Table G33 shows the results of fixing the colour correlation window at 9 pixels.

Table G31, G32, G33 – Window size experiment of the toupe image views V_1 and V_2 showing three window size experiment results, G31-G33, from left to right respectively.

<i>Spin = colour = 15</i>			<i>Spin fixed at 15</i>			<i>Colour fixed at 9</i>		
Win.	Overl.	Acc.	Win.	Overl.	Acc.	Win.	Overl.	Acc.
5	99.84	52.71	5	99.84	52.71	5	52.63	100.00
7	99.84	52.71	7	99.84	52.71	7	52.63	100.00
9	51.73	98.29	9	52.63	100.00	9	51.73	98.29
11	98.68	53.33	11	98.68	53.33	11	52.63	100.00
13	98.68	53.33	13	98.68	53.33	13	52.63	100.00
15	98.68	53.33	15	98.68	53.33	15	52.63	100.00
17	98.68	53.33	17	98.68	53.33	17	52.63	100.00
19	98.52	53.42	19	98.68	53.33	19	51.73	98.29
21	98.68	53.33	21	98.68	53.33	21	51.73	98.29
31	99.84	52.71	31	98.68	53.33	31	51.73	98.29
41	99.84	52.71	41	98.68	53.33	41	67.19	78.33

Appendix H: The Correlator GUI

The correlator GUI was developed for this Thesis and was able to run manual, semi-automatic and automatic registration methods. The GUI was written in Perl/Tk and the underlying functions were written in C and Perl under the Linux operating system. The project generated about 15,000 lines of code.

Figure H – The main correlator GUI and description of its functionality.