uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Philip Curtis
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Frequency-Domain Range Data Registration for 3-D Space Modeling in Robotic Applications

TITRE DE LA THÈSE / TITLE OF THESIS

P. Payeur
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. Cuhadar

E. Dubois

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Frequency-Domain Range Data Registration for 3-D Space Modeling in Robotic Applications

By:

Phillip Curtis

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Information Technology and Engineering

Faculty of Engineering

University of Ottawa

Your file Votre référence
ISBN: 0-494-14893-4
Our file Notre référence
ISBN: 0-494-14893-4

Compressed Abstract

The registration method presented in this thesis does not require an initial estimation of the transformation, and avoids problems of the classical iterative techniques by employing the multi-dimensional Fourier transform, which decouples the estimation of rotational parameters from the estimation of the translational parameters. Using the magnitude of the Fourier transform, an axis of rotation is estimated by determining the line that contains the minimal energy differential between two rotated 3-D images. By using a coarse to fine approach, the angle of rotation is determined from the minimal sum squared difference between the two rotated image. As the Fourier transform introduces hermitical symmetry in the rotation, the proper solution is identified through the use of a phase-correlation technique, and the estimate of translation is simultaneously obtained. Experimental results illustrate the accuracy that can be achieved by the proposed registration technique and performance is compared with that of the classical iterative closest point method.

Abstract

Autonomous robotic applications require automatic registration techniques. Most classical registration techniques are iterative in nature, or require user intervention in order to specify matching feature points. Techniques requiring user intervention are unsuitable for autonomous robotic applications, while iterative-based techniques do not scale well as the dataset size increases, and additionally tend towards locally minimal solutions. To avoid the problem of local minima solutions, an accurate initial estimation of the transform is required for iterative algorithms to perform properly.

The method presented in this thesis does not require an initial estimation of the transformation, and avoids problems of the classical iterative techniques by employing the multi-dimensional Fourier transform, which decouples the estimation of rotational parameters from the estimation of the translational parameters. Using the magnitude of the Fourier transform, it is quite simple to extract an axis of rotation by determining the line that contains the minimal energy differential between two rotated images. By using a coarse to fine approach, it is then possible to determine the angle of rotation that corresponds to the minimal sum squared difference between the two rotated image. As the Fourier transform introduces Hermitian symmetry in the rotation, a method is needed to discern the proper solution. Through the use of a cross-correlation technique, this ambiguity is solved, and the estimate of translation is obtained.

Robotic applications also have varying needs in terms of registration precision, for example the performance of collision avoidance requires less precision than object

manipulation. The proposed algorithm can take advantage of these scenarios by using a coarser representation of the occupied space when performing the registration estimation for collision avoidance, and then a finer representation of the objects during a manipulation task, with the trade off being a higher computation cost. Given the wide availability of fast implementations (in both hardware and software) of the discrete Fourier transform on many different platforms, the proposed algorithm is highly scalable according to the hardware available, and registration precision desired. Experimental results are depicted for various spatial resolutions of several simulated and real scenes, which illustrate the benefits of the proposed approach for range data registration.

Acknowledgements

I would like to thank my supervisor, Dr. Pierre Payeur, who patiently provided his guidance and understanding during my research. Without him, this thesis would not have been possible. Secondly, I would like to thank the University of Ottawa and NSERC, who provided me with the necessary funding over the past few years. Finally, I would also like to give my thanks to my family and friends, who provided me with the support and encouragement that I needed while I completed this thesis.

Table of Contents

Table of Figures

Table of Tables

Acronyms

DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
ICP	Iterative Closest Point
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
LSE	Least Squared Error
MSE	Mean Squared Error
OEM	Original Equipment Manufacturer
SNR	Signal to Noise Ratio

Chapter 1. Introduction

The role of robots, autonomous, and semi-autonomous systems in today's society is growing amid a plethora of conditions such as: increased awareness of global security, advancing technology, increased worker productivity, hostile environment exploration, and improving quality of life. These applications all require some intelligence that is built into the robotic agent in order to interpret data obtained by sensors of the surrounding environment. The role of the intelligence is to create a coherent model of the environment so that the robotic agent may successfully avoid potential obstacles, while at the same time complete the desired task. To create a coherent model of objects and of the surrounding environment requires the data acquired from numerous viewpoints to be registered to a common frame of reference.

This thesis presents a registration estimation algorithm that is automatic, and requires no *a priori* knowledge of the object that is being registered, with the only assumption being that there is sufficient overlap for the algorithm to produce an accurate estimation. The degree of overlap required for the algorithm to produce accurate results is investigated empirically in the experimentation chapter. The proposed algorithm takes advantage of the fact that the Fourier transform decouples the estimation of the rotational parameters from the estimation of the translational parameter. This is accomplished by separating the frequency information into a magnitude and phase component. The axis of rotation is determined by finding the zero crossing line between the differences of two frequency domain magnitude images. The angle of rotation is determined by finding the

minimal squared error, in a coarse to fine manner, of a subset of frequency domain location between the first frequency domain image and a progressively rotated second frequency domain image. Due to the nature of the Fourier transform, there are two possible solutions for the rotational parameters, separated by 180°. To determine the correct rotational parameters, as well as estimate the translation parameters, the second image is derotated in the space domain by both possible solutions, and then a phase correlation utilising the frequency domain is applied. The solution that produces the most impulsive phase correlation is selected as the correct solution, with the location of the impulse corresponding to the translational parameters.

The proposed algorithm, although implemented only for 3-D data sets, provides the framework for a multiple dimension extension, and as a result the theory, as well as the explanations, are generalised where possible to the multiple dimensional case. The proposed algorithm also recognizes that the majority of range sensors also collect other data types, such as reflectance and colour, and can take advantage of this additional information in the estimation of the registration, although it has only been applied for strictly range data in this thesis.

1.1. Motivation

The proliferance of low cost, and high quality range sensing systems has lead to their use for many different purposes, such as creating virtual objects for virtual museums, space exploration, and games. The prevalent methods for registering range images involve relying on the positional sensors of the data acquisition apparatus, relying

on complex feature detection and matching algorithms, or iterative algorithms that may converge to an incorrect solution. These prevalent methods may be unnecessarily precise for a particular application, such as object avoidance, thus a registration algorithm that is scalable according to the precision needed is highly desired. Finally, one of the major drawbacks of these methods is that the execution time increases dramatically with an increase in the number of points in the datasets to be registered.

Lucchese *et al.* describe a 3-D range image registration algorithm operating in the frequency domain in [1], which is complex and is inefficient in both computational and memory resources. On the other hand, the algorithm described has potential for scalability with dataset size, while avoiding the complex task of feature detection as well as the problems inherent in iterative approaches. This thesis uses the proposed theory as a starting point, and makes frequency domain registration less complex, and more computationally and memory efficient, while generalizing the algorithm into a multidimensional framework for data containing many different values and preserving the accuracy of the registration estimate.

1.2. Contributions

This thesis makes contributions to the field of registration estimation by:

1) Proposing an automatic registration estimation algorithm that is non-iterative, does not require human intervention, does not require an initial estimation, and is independent of the geometry of the object.

2) Proposing a scalable and resource (computational and memory) efficient method using the frequency domain for registration estimation.

3) Introducing a generalized framework for frequency domain range registration that can handle multiple data types in multiple dimensions.

1.3. Organization

This thesis is divided into six chapters, the first of which is this introductory chapter. The second chapter provides a review of several different registration methods. The next chapter derives the basic theory of frequency domain registration. Chapter 4 details the proposed frequency domain registration algorithm, while stressing the similarities and differences with the algorithm proposed by Lucchese *et al.* in [1]. The experimental setup, as well as the results from the application of the proposed algorithm, is contained in chapter 5. Chapter 6 provides the conclusion as well as future work to be performed. The three appendices located at the end of this thesis provide background material in rigid linear transformation, Hermitian symmetry, and the Fourier transform.

Chapter 2. Literature Review

The registration of range images is a complex task, which has been explored for numerous years. As a result there are many methods, which endeavour to solve this problem. The various algorithms that exist can be broken down into several categories: iterative, direct calculation, automatic, semi-automatic, space-time domain, transformation domain, stochastic modeling, and so on. The fact that all of these categories are not mutually exclusive makes categorization much more difficult. In this literature review, the methods are split up into three categories: space domain algorithms, frequency domain approaches, and other transform domain approaches.

2.1. Space Domain Methods

Space domain methods are the most classical of registration techniques, which includes the widely used iterative closest point algorithm and its derivatives, the direct three-point problem, and feature matching. In this section an in depth look into the Iterative Closest Point (ICP) method is made, followed by a discussion of some of the modifications that other researchers have proposed to improve this method. Finally, alternate space domain methods are discussed.

2.1.1. Iterative Closest Point

The most widely explored approach to solve the registration estimation problem seen in literature is the iterative approach, and the most widely adopted of these approaches is the ICP algorithm and its various reincarnations and modifications. The

ICP algorithm, initially proposed by P.J. Besl and N.D. McKay [2], describes a method for registering a set of 3D data (P) with a reference or model dataset (X). The method operates by calculating the closest points in set P, with those in set X. From this matching of closest points, an estimate for the registration parameters is made. P is transformed by this estimate, and the mean squared error (MSE) between the transformed P and X is made. If the MSE is not beneath a predefined threshold, the estimation process is repeated using the newly transformed dataset P, otherwise the current estimate of the registration is the solution.

The ICP algorithm, as with most iterative convergence algorithms, tends towards the closest local minima when using discrete datasets. The solution obtained may or may not correspond to the global minimizing solution, and hence the ICP algorithm requires an initial estimate that is close to the actual registration parameters, or a particularly transformed dataset, to obtain a precise solution. The ICP algorithm is flexible as to what form the datasets take (e.g. points, curves, surfaces, etc). The ICP algorithm, as described by P.J. Besl and N.D. McKay in [2], uses quaternion notation internally to obtain the rotation and translation estimates. The estimation of the registration parameters is performed in two stages, by first calculating the rotational parameters and then using the rotational parameters to calculate the translational parameters; hence if the rotation is incorrect, the translation will be incorrect. The advantages of the ICP algorithm are its precision, flexibility, and ease of use, and the main disadvantage of the ICP algorithm is its tendency to converge to local minima solutions instead of the proper global minimum solution without a close initial estimate.

The ICP algorithm provides very accurate and precise solutions under ideal situations involving minimal rotation and translation, but as soon as it is applied to situations with large rotations and translations, then ICP algorithm results in an incorrect local minima solution. Due to the precision of the ICP algorithm in ideal circumstances, a lot of research has been directed towards improving the ICP algorithm to make it more applicable for more common situations where there are large differences in the registration. There are several papers available in literature, which discuss modifications to the ICP algorithm in order to provide better convergence toward the global minimum. This is achieved by either transforming the data, altering the matching criterion, providing a close initial estimate of the registration parameters, or by acceleration of the algorithm at various steps.

2.1.2. Alterations and Variations of the ICP Algorithm

To minimize the probability of approaching a local minimum in the ICP algorithm, M.A. Rodrigues and Y. Liu [3] have introduced an error histogram. This error histogram aids in the point correspondence to minimize the correspondence error, thereby minimizing the registration error.

T. Masuda and N. Yokoya [4] perform a random sampling of the dataset and then use a least median of squares estimator within the ICP framework to provide results. This reduces computational complexity since the data is sub-sampled by the random sampling, minimizes the effect of noise and outliers due to the least median of squares

estimator, and increases probability of achieving a global minimum solution depending on the strategy used to sample the image.

D. Huber, O. Carmichael, and M. Hebert [5] have added an initial estimation phase to the ICP algorithm based upon the local signature of a range image measurement. The local signature is based upon the idea of a spin-image, which describes the local area phenomenon in relation to the normal of any particular point. The spin images of each viewpoint are sub-sampled, to reduce the complexity of the following matching step, and then are matched such that these local signatures are aligned. This produces an initial estimate for the traditional ICP algorithm.

Y. Liu and Marcos A. Rodrigues [6] have modified the ICP algorithm to produce what the authors term the geometric histogram ICP method (GH-ICP). The GH-ICP method introduces a geometric histogram to produce possible solutions to the registration problem based upon the geometric properties of the proposed matches made in the ICP algorithm. From the geometric histogram, the solution that corresponds to the peak of the histogram is used as the solution for the current step of the ICP algorithm. The GH-ICP algorithm is more immune to outliers, and less susceptible to problems introduced by smaller overlapping regions than that of the classical ICP algorithm.

R. Benjemaa and F. Schmitt [7] have improved the execution efficiency of the ICP algorithm by adding a multi-Z-buffer, and the capabilities of matching more than two overlapping range images at once. Each range image is segmented into a set of optimized z-buffers by the usage of Gauss spheres. Gauss spheres describe the normal to the

surfaces. Since the data is now arranged in multi-z-buffers, simple 2-D image processing techniques can be now be utilized; this reduces the complexity of the matching step in the ICP algorithm resulting in an overall acceleration in the registration process. The registration of multiple viewpoints is accomplished through the analysis of multiple overlapping regions using statistical methods. The multi-z-buffer technique relies upon data that is sampled upon a regular grid, otherwise resampling is required.

G. Sharp, S. Lee, and D. Wehe [8] have modified the ICP algorithm's point matching criterion to use invariant features, in what they term the Iterative Closest Points using Invariant Features (ICPIF). ICPIF involves choosing the point correspondence based upon a distance that is proportional to the sum of the Euclidian position and feature descriptor. An invariant feature is a feature that does not change upon the motion of the camera/object.

D. Simon, M. Hebert, and T. Kanade [9] have introduced methods to speed up the ICP algorithm through the use of kd-trees, closest point caching, efficient distance calculations, and algorithm acceleration. The use of kd-trees in the closest point computation reduces the complexity from $O(N_{Data\_Set}N_{Model\_Set})$ to $O(N_{Data\_Set}logN_{Model\_Set})$. The use of closest point caching reduces the amount of searching for a closest point at each stage, since in the ICP algorithm, the resulting registration parameters do not change rapidly, so the closest point at stage k is likely to be the closest point at stage $k+1$, especially if n closest points are cached for each point on the model set. Through the implementation of a closest surface point computation using a triangle mesh that represents the model set stored in both 2D and 3D representations, it is possible to

calculate the distances using simple 2D geometric equations, thereby reducing the complexity of the distance calculations. Finally the algorithm can be accelerated through the use of extrapolation techniques if the registration differences between iteration cycles are similar. The primary drawback of the work is the requirement that the model set is stored as a triangular mesh in kd-trees, but this work can be segregated and applied part and parcel in other algorithms. The algorithm presented allows real-time registration of range images.

2.1.3. Other Space-Domain Methods

H-Y. Lin and M. Subbarao [10] introduce a new technique for fast 3D model reconstruction. The authors describe a complete modeling system using range data acquired from a stereo vision system, where the vision system is rotated by a known amount about the object being modeled. The registration process uses the known angular displacement about the object in the initial estimation of the registration parameters. Performing a least squares matching process in the overlapping regions refines this estimate. The registration technique is quick, but relies on many known parameters such as axis of rotation and translation.

C-S. Chen, Y-P. Hung, and J-B. Cheng [11] introduce a new technique, which is termed the Data-Aligned Rigidity-Constrained Exhaustive Search (DARCES) method. This method relies on the extraction of control features from the data to be registered, which are then matched with control features on the model dataset. A minimum of three control points are required to fully estimate the registration in a 3D environment. This

method is quick, but the accuracy is limited by the data. For refinement of the results, the authors suggest using the estimated registration as the initial estimate for the ICP algorithm.

E. Gagnon, J-F. Rivest, M. Greenspan, and N. Burtnyk [12] propose an operator assisted registration technique to be used when high reliability is required. In this method, the operator selects matching features between sets of range intensity data. This estimate of the matching of the features is improved upon by a localized feature correlation algorithm. The method only requires 3 feature matches, and takes into account the operator selecting the easiest to match features first, and the hardest to map features last, by applying a progressively decreasing weight to each feature match. The translation is estimated based upon the first feature matched, then using this as the point of origin, the second feature match is then rotated to be aligned. The alignment of the third feature match then occurs, completing the registration estimate. The authors suggest that if further refinement is required, the resulting registration estimate could be used as the initial estimate to the ICP algorithm. The primary drawback of this algorithm is the requirement of feature matching selected via a human operator, and as such cannot be used in autonomous systems without modification.

D. Tubić, P. Hébert, and D. Laurendeau [13] propose a method of pose refinement and 3D surface modeling using range curves. Their motivation for this arises from the fact that many laser range sensors acquire data along a predetermined pattern, usually a line, or a sinusoidal curve, and by exploiting this information, surface normals and other parameters can be accurately estimated. The pose refinement technique relies on

scanning patterns involving many intersections, and then estimating the surface normal at the points of intersection. Once the surface normal is estimated, a refinement on the pose of range curve is made, and a model is created. As more range curves are acquired, the respective pose refinements improve, and the model becomes more accurate and complete. The advantage of this technique is that the model is built as the data is acquired, and each line of acquisitions is registered as it is being merged, but it requires an estimate of the pose of the range curves in order to determine the intersections, as well as complex scanning patterns.

C.M. Cyr, A.F. Kamal, T.B. Sebastian, and B.B. Kimia [14] propose a method of 2-D to 3-D registration based upon shape matching. The objects to be registered are a 3-D model of the object, acquired via computed tomography scan (CT-Scan) or magnetic resonance imaging (MRI), with a 2-D image, acquired via X-Ray or fluoroscopy. The algorithm first uses a coarse to fine approach about a sphere centered on the 3-D object to find the viewing axis (parameterized by the azimuth and altitude angles), analogous to the 3-D axis of rotation. During this coarse to fine descent, the 2-D image is matched against the 3-D model, for each current viewing axis. The viewing axis, which has the minimal error, is zoomed upon for the next iteration in the coarse to fine algorithm. Once the desired angular resolution is reached for the viewing axis, the angle of rotation is determined. The images are matched by first transforming the images into what is termed 'shock' graphs, and then a simulated annealing algorithm is used to determine the extent of the match. The algorithm does not describe how to obtain translation.

2.2. Frequency Domain Methods

L. Lucchese and G.M. Cortelazzo [15] demonstrate a technique using frequency domain analysis for estimating 2-D rigid transformations. The algorithm is based upon several properties of the Fourier transform, including the separation of the rotational parameters from the translational parameters, Hermitian symmetry (see appendix 2) due to real input data, and the difference between the magnitudes of a rotated image and a mirrored version of the original in the frequency domain that produces a pair of orthogonal zero crossing lines at angles corresponding to half of the rotation angle. The presence of these lines inspires the use of a Hough transform to determine the angle of these lines, and as a result the authors use a histogram involving the polar coordinates of the magnitude frequency spectrum. The peak of this histogram provides an estimate to the angle of these lines. A coarse to fine approach is used when creating these histograms to reduce computational complexity, by continuously recreating the histogram at higher resolutions in small regions about the peaks. To increase the frequency domain resolution, they zero-pad the input images. The estimation of the rotation angle provides two possible solutions with a 180° separation due to the Hermitian symmetry of the Fourier transform. This ambiguity in the solution to the angle of rotation is solved in the translation estimation. The translation estimate employs a phase correlation function between the two images, for each of the rotation estimates. The proper estimate will produce an impulsive function, while the other estimate will not produce an impulsive function. The location of the impulse provides the translation estimate. This method has a high degree of immunity to noise, since noise tends to have a flat spectrum, and as a

result, the zero crossing lines will still be present. Additionally, if a low-pass filter is applied, the noise power is weakened, and the coarse to fine algorithm improves in performance since many peaks that are very close together become one larger peak. The disadvantages of this algorithm are the time that it takes to perform the fast Fourier transform (FFT), and the histograms, but the algorithm only relies on the raw data – no features need to be extracted. The authors have obtained results with precision to one one-hundredth of a degree for the rotational parameter without the presence of noise, and have good performance (less than 1 degree error) in up to a –9 dB SNR. The translation estimates maintain precision until –3dB SNR.

L. Lucchese, G. Doretto, and G.M. Cortelazzo [1] extend their previous work with frequency domain registration estimation in 2-D [15] to the 3-D case. As previously stated, the Fourier transform decouples the rotational parameter estimation from the translation estimation. In order to prevent the impulsive nature of the effect of singularities on the frequency spectrum, the data set is convolved with a spherical Gaussian kernel with standard deviation of 0.05 and diameter of 7 voxels. This creates a spherical solid region about each point with density decaying with distance from the point.

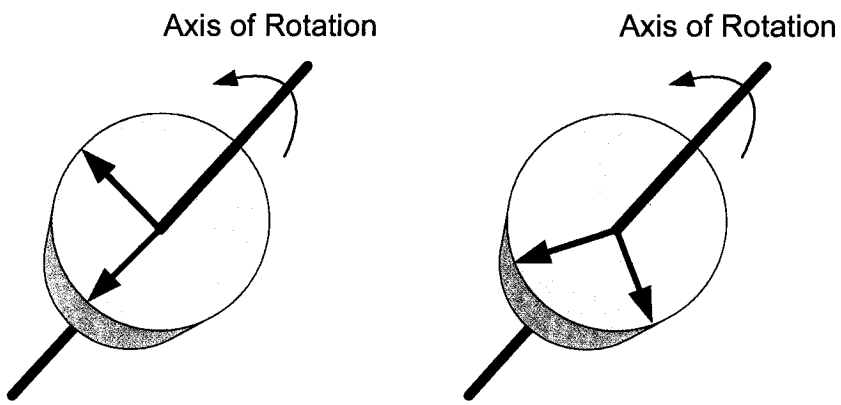

Fig. 1. Rotation About an Axis

It is well known that all rotations in 3-D space can be represented as a rotation about an axis of rotation (see Fig. 1), a fact of which the frequency domain technique takes advantage. By rotating an object in space about an axis, the position of all points change – except those belonging to the axis of rotation. This, when described mathematically, corresponds to multiplying a scaled version of an eigenvector of a matrix with the matrix itself (note that the axis of rotation corresponds to the eigenvector with eigenvalue equalling 1 for the rotation matrix). This holds true in the frequency domain, since rotation is not affected by a Fourier transform, hence by determining the zero-line in the difference function of the magnitude of the Fourier transforms of the images to be registered, the axis of rotation can be determined.

The authors perform the axis of rotation estimation step by determining the radial projections of the difference image through the DC (0,0,0) point. By performing this step, the determination of the axis of rotation is reduced to determining the minimum of a 2-D function. The estimate is further refined by resampling the function in a small thin

cylinder around the estimated axis of rotation, then determining angular histograms of the projection of the cylinder onto the three orthogonal elementary planes, and then finally determining the angles corresponding to the maxima for each histogram. These maxima are used to determine the optimal axis of rotation. With the axis of rotation now determined, the coordinate system is transformed such that the problem becomes a 2-D rotation problem about the *w*-axis, as illustrated in Fig. 2.

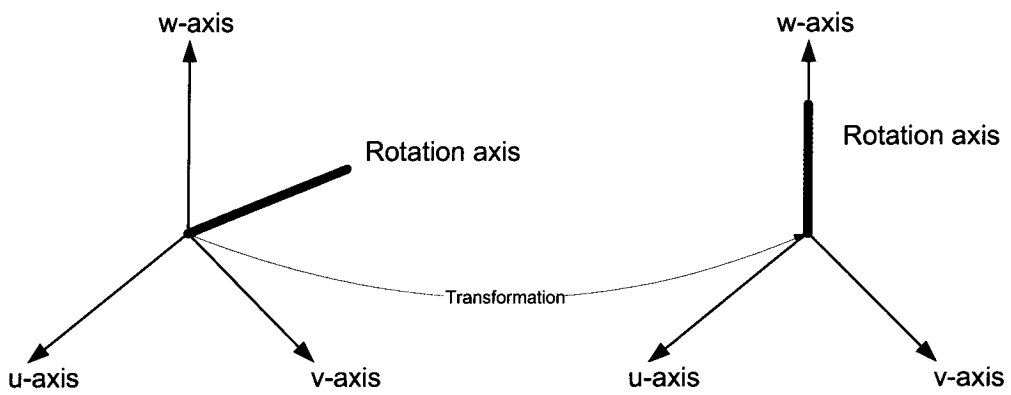

Fig. 2. Co-ordinate System Transformation

Now that the problem is reduced to a 2-D problem of estimating the rotation about *w*, the authors create a 2-D image from the 3-D image by integrating along the *w*-axis. The complexity of finding the angle of rotation is further reduced to 1-D by changing to polar co-ordinates and integrating along the distance parameter. Finally, the angle of rotation is determined by the peak of the correlation between the corresponding 1-D functions of each range image.

Due to the Hermitian symmetry of the Fourier transform, there are two complementary solutions, separated by 180°. The ambiguity between solutions is

resolved in the estimation of translation. To estimate the translation, the original data is rotated by each solution, and transformed into the frequency domain. A phase correlation between images is performed. The phase correlation function corresponding to the correct solution will be impulsive in nature, and the location of the impulse corresponds to the translation. The phase correlation function corresponding to the incorrect solution will not be impulsive in nature.

In order to avoid the computational penalty of performing a 3-D phase correlation, the authors perform three 1-D phase correlation functions based upon the projections onto the primary axis. Additionally to minimize the numerical errors involved in the computation of differences between small numbers, a logarithmic difference function is used for the estimation of the rotation axis. Finally, to reduce the effects of discrimination in estimating the angle of rotation, a minimal search based windowing function is used along each plane, and these minima are added together to form the 2-D image. The method also states for further refinement of the results, apply the ICP algorithm using the estimated transformation determined by the algorithm above as the initial value. The disadvantages of this algorithm is the computational cost of applying the FFT several times (1 time for each image for estimation of the axis of rotation, 1 more time for one image in estimation of the angle of rotation, and finally 2 more times on one image for the estimation of translation), as well as the need for computing several histograms. The main advantage of this algorithm is that it avoids the need to extract and match features, which may be more computationally intensive than applying the multiple

FFTs, as well as avoiding local minima solutions that may occur with traditional iterative algorithms such as ICP.

2.3. Other Transformation Domain Methods

A.P. Ashbrook, R.B. Fisher, C. Robertson, and N. Werghi [16] have developed a system to segment articulated objects into rigid subsets using surface patches. The system operates by segmenting the range data into surface patches, and then estimating the rigid transformation that aligns most of the surface patches. A comparison between the sets of patches are made, and those which are in agreement are considered to be part of the same segments, and those which are not in agreement are considered to be in a different segment of the articulated object. This sequence of registration estimation and segmentation continues until there are no more segments to register. Their registration technique involves finding corresponding surface patches between range images. These correspondences are then mapped through the use of a Hough transform, and those surface patches contributing the highest peak in the Hough transform are considered to be part of the rigid sub component, and are registered according to where the peak is in the Hough transform. The advantages of this method of registration is that matches which are incorrect are filtered out via the Hough transform, thus increasing the reliability of the registration estimation, and this method can deal with articulated objects, which standard registration methods cannot achieve without heavy pre-processing. The drawback of this method is its complexity in fitting the surface patches to the raw data and finding the correspondence between images.

P. Payeur and C. Chen [17] propose a registration method by applying planar patches to the raw range data, then transforming the planar patches to a compact Gauss sphere representation, and finally registering the Gauss spheres. The planar patches are applied by segmenting individual lines of data, acquired by a laser line range scanner, into straight-line segments. Line segments are then matched between acquired lines, and are merged into surface patches. Once the surface patches are obtained, the surface normals of the patches, along with the centre coordinates of the patches are merged into a Gauss sphere. The Gauss spheres are then registered by determining the best rotational parameters to align the Gauss spheres. The translation estimate is acquired by the best alignment of the centres of the surface patches. This technique is less affected by noise since the use of surface patches minimizes the noise present in the range data set before the transformation into a Gauss sphere, but unfortunately the assumption of flat rigid surfaces minimizes the applications to which this algorithm can be utilized (i.e. surfaces with curvature), and it assumes some organization to the data.

K. Higuchi, H. Delingette, M. Hebert, and K. Ikeuchi [18] propose a registration method using a spherical representation. The method essentially calculates how to warp a sphere to obtain the representation of the data using simplex angles. Simplex angles contain a measure of the local convexity of the surface. Using the simplex angle spherical representation separates the rotational estimate from the translational estimate similar to that used in Gauss sphere or frequency domain representations. The rotational estimate occurs by rotating the spheres about all three degrees of freedom until the rotation yielding a minimal mean squared distance is achieved. The translation is then

determined using a linear least squares approach between the de-rotated object and the second object. The disadvantage with this method is that the calculation of the simplex angles from unorganized raw data takes a long time due to its complexity, but fortunately in addition to the registration estimate, a surface model of the object is created at the same time.

2.4. Chapter Summary

In summation, there are several distinct algorithms that perform registration of range images. The most common type of registration algorithms are those based off of the ICP algorithm developed by Besl *et al.* in [2]. The ICP algorithm performs well for very small rotations and translations, but poorly for larger rotations and translations. The goal of the other papers is to improve the algorithms performance to cover more cases. Other researchers have investigated other methods for determining the registration of range images while still using the spatial domain range data. Lucchese *et al.* in [1] propose an interesting method involving the use of frequency domain techniques, which is more broadly applicable compared to the ICP algorithm. Finally, researchers have investigated various other transformations of range data in order to estimate registration.

Chapter 3. Frequency Domain Registration Estimation Derivation

Registration estimation involves determining the rotations and translations required to align one image with another. In order to understand the mathematical principles behind registration, some introductory theory of rigid linear transformations (see appendix 1) is required. The estimation of registration parameters typically relies on the extraction and matching of features or points between two images. This is true of the spatial domain algorithms, and other transformation domain algorithms presented in chapter 2. Frequency domain approaches avoid the need to match features, and can deal with unorganized data clouds due to the nature of the Fourier transform (see appendix 3). By avoiding the feature detection, extraction, and matching steps in registration, the frequency domain algorithms avoid possible imprecision and poor matches inherent to this sort of process. This chapter describes the theory required for frequency domain registration techniques, expanding upon the mathematical arguments put forth by Lucchese *et al.* in [1].

In literature, there are several different conventions on how to notate discrete functions, continuous functions and vectors. The notation which is used within this thesis uses f[ ] to describe discrete functions, f( ) to describe continuous functions, and $\vec{[]}$ to describe vectors. In practice, the source data (denoted below by $\mathrm{Im}(\vec{x})$) will be a series

of dirac pulses representing the presence of a detected surface, but the mathematical framework discussed in this chapter is meant applicable to any type of range data.

Let N be the number of dimensions of the signal, and M be the diagonal matrix containing the reciprocal of the size of each dimension. Let $\mathrm{Im}(\vec{x})$ be the continuous space domain range function, with $\vec{x}$ being the vector that represents the location of the function in Cartesian space. Let $\mathrm{Im}_1[\vec{n}]$ be the discrete space domain samples of the continuous space domain range image, $\mathrm{Im}(\vec{x})$, from one viewpoint, and let $\mathrm{Im}_2[\vec{n}]$ be the discrete space domain samples of the continuous space domain range image, $\mathrm{Im}(\vec{x})$, from a different viewpoint, with the vector $\vec{n}$ indicating the sample location in discrete Cartesian coordinates, with respect to the origin of the image. Let the rigid transformation between Im_1 and Im_2 be represented by:

$$\mathrm{Im}_1[\vec{n}] = \mathrm{Im}_2[R\vec{n} + T] \tag{1}$$

where R is the NxN rotation matrix, and T is the Nx1 translation vector. Note that it is assumed that $\mathrm{Im}_2[R\vec{n} + T]$ has been rotated and translated in continuous space and then appropriately filtered and resampled onto the discrete space grid.

The discrete Fourier transform (DFT) representation of Im_1 and Im_2 are:

$$F_{\mathrm{Im}_1}[\vec{k}] = \sum_{n_N=0}^{M_N-1} \sum_{n_{N-1}=0}^{M_{N-1}-1} \cdots \sum_{n_1=0}^{M_1-1} \mathrm{Im}_1[\vec{n}] e^{-j2\pi \vec{k}^T M \vec{n}} \tag{2}$$

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_2\left[\vec{n}\right]e^{-j2\pi\vec{k}^T M\vec{n}} \tag{3}$$

$$,\vec{n}=\begin{bmatrix}n_1\\n_2\\\cdots\\n_N\end{bmatrix},\quad \vec{k}=\begin{bmatrix}k_1\\k_2\\\cdots\\k_N\end{bmatrix},\quad M=\begin{bmatrix}\frac{1}{M_1} & 0 & \cdots & 0\\ 0 & \frac{1}{M_2} & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & \frac{1}{M_N}\end{bmatrix}$$

where $\vec{k}$ is the vector representing the N-dimensional discrete frequency index, n_1, n_2, ..., n_N are the components of the vector $\vec{n}$, and M_1, M_2, ..., M_N are the discrete size of each of the respective N dimensions. Now, if the rigid transformation depicted in eq.(1) is applied to eq.(3), the relationship between F_{Im1} and F_{Im2} can be found. For the sake of simplicity of expression for the relationship between F_{Im1} and F_{Im2}, F_{Im2} is used even though a different notation would normally be used in the following contexts due to the introduction of the rigid linear transformation.

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_2\left[R\vec{n}+T\right]e^{-j2\pi\vec{k}^T M\left(R\vec{n}+T\right)}$$

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_1\left[\vec{n}\right]e^{-j2\pi\vec{k}^T M\left(R\vec{n}\right)}e^{-j2\pi\vec{k}^T MT}$$

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\left(\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_1\left[\vec{n}\right]e^{-j2\pi\vec{k}^T MR\vec{n}}\right)e^{-j2\pi\vec{k}^T MT}$$

Since the transpose of a rotation matrix is equal to its inverse (see appendix 1.),

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\left(\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_1\left[\vec{n}\right]e^{-j2\pi M\vec{k}^T\left(R^{-1}\right)^T\vec{n}}\right)e^{-j2\pi\vec{k}^TMT}$$

$$F_{\mathrm{Im}_2}\left[\vec{k}\right]=\left(\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_1\left[\vec{n}\right]e^{-j2\pi M\left(R^{-1}\vec{k}\right)^T\vec{n}}\right)e^{-j2\pi\vec{k}^TMT}$$

If a change in variables to the above equation is made ($\vec{k}\rightarrow R\vec{k}$), it then becomes:

$$F_{\mathrm{Im}_2}\left[R\vec{k}\right]=\left(\sum_{n_N=0}^{M_N-1}\sum_{n_{N-1}=0}^{M_{N-1}-1}\cdots\sum_{n_1=0}^{M_1-1}\mathrm{Im}_1\left[\vec{n}\right]e^{-j2\pi\vec{k}^TM\vec{n}}\right)e^{-j2\pi\left(R\vec{k}\right)^TMT} \tag{4}$$

Now, notice that in eq.(4) the translation component is independent of $\vec{n}$, and that the part of the equation in brackets is equal to F_{Im1} (see eq.(2)). This leads to the deduction of the relationship between F_{Im1} and F_{Im2}:

$$F_{\mathrm{Im}_2}\left[R\vec{k}\right]=F_{\mathrm{Im}_1}\left[\vec{k}\right]e^{-j2\pi\left(R\vec{k}\right)^TMT} \tag{5}$$

From eq.(5), two separate equations can be developed, one for solving for the rotation matrix (R), and one for solving for the translation vector (T) when given the rotation matrix. This is accomplished by separating the equation into amplitude and phase components.

$$\left|\mathrm{F}_{\mathrm{Im}_2}\left[R\vec{k}\right]\right| = \left|F_{\mathrm{Im}_1}\left[\vec{k}\right]\right| \tag{6}$$

$$\angle\mathrm{F}_{\mathrm{Im}_2}\left[R\vec{k}\right] = \angle F_{\mathrm{Im}_1}\left[\vec{k}\right] - 2\pi\left(R\vec{k}\right)^T MT \tag{7}$$

From eq.(6), it is possible to deduce the rotation matrix (R) using the amplitude spectrum of Im_1 and Im_2. Once the rotation matrix is known, the translation vector (T) can be solved for through the use of linear numerical methods based upon the phase spectrum of Im_1 and Im_2 (eq. (7)).

3.1. Determining the Rotation Matrix

In order to determine the rotation matrix, there are a few issues that must be addressed. The first and foremost is that the Fourier transform of real data (range data is assumed to be real), is Hermitian symmetric (see appendix 2). In other words, the frequency domain spectrum for the frequency values between 0 and π are also represented by the complex conjugate of the values contained between 2π and π, i.e.:

$$F(w) \rightarrow F^*(2\pi - w) \tag{8}$$

In the discrete mapping of the DFT, where $\vec{K}$ is the column vector containing the discrete size along each dimension, and $\vec{k}$ is a discrete frequency location, then this effect is represented by:

$$F\left[\vec{k}\right] = F^{*}\left[\vec{K} - \vec{k}\right], \quad \vec{K} = \begin{bmatrix} M_1 \\ M_2 \\ \cdots \\ M_N \end{bmatrix} \tag{9}$$

This mapping is both beneficial and detrimental to determining the rotation matrix. The benefit is that only half of the frequency data is unique, therefore only half of the DFT needs to be computed. The drawback to the Hermitian symmetric mapping is that two solutions for rotation are obtained, with a separation of 180° about the axis of rotation.

In order to determine the rotation matrix, the properties of rotation matrices will be exploited, in particular that a rotation matrix will have a single unique real eigenvalue, whose eigenvector is the axis of rotation. Hence, through eq. (6), if the locations where $\left|\mathrm{F}_{\mathrm{Im}_2}\left[R\vec{k}\right]\right| = \left|\mathrm{F}_{\mathrm{Im}_2}\left[\vec{k}\right]\right| = \left|\mathrm{F}_{\mathrm{Im}_1}\left[\vec{k}\right]\right|$ are found, then the line which passes through those locations is the axis of rotation. To determine these locations, a simple observation can be made: the values along the axis of rotation do not rotate. This means that the values are equal between the two images along the axis of rotation, hence if the difference between F_{Im_1} and F_{Im_2} is calculated, the locations which are zero in this difference should lie on the axis of rotation.

To determine the angle of rotation, rotate F_{Im_1} about the axis of rotation, until the rotated F_{Im_1} equals F_{Im_2}. This can be accomplished by minimizing the mean squared error (MSE) between transformed magnitude images, $\left|\mathrm{F}_{\mathrm{Im}_1}\right|$ and $\left|\mathrm{F}_{\mathrm{Im}_2}\right|$, for each value of the

angle of rotation. As previously stated this angle will also have a complementary solution with a 180° offset due to the Hermitian symmetry property of the Fourier transform.

For an example of the effects of complementary axes and angles of rotation in 2D, see the following:

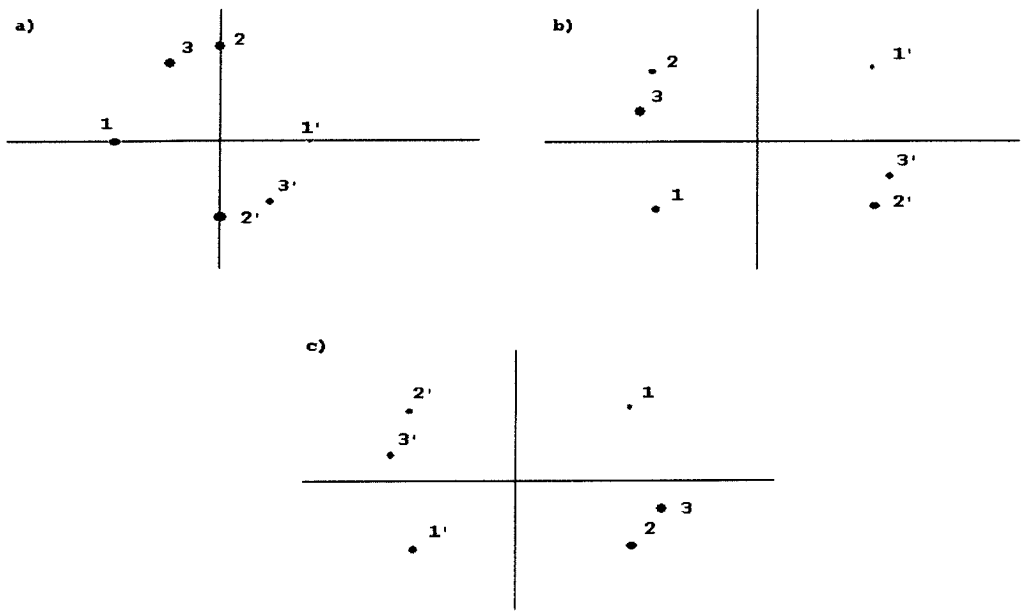

Fig. 3. Example showing a) the original DFT, b) rotation of $\pi/4$, c) rotation of $5\pi/4$

According to Fig. 3, where locations of interest are found in the top left quadrant of a) and are labelled as 1, 2, and 3, and their Hermitian symmetrical counterparts similarly labelled as 1', 2', and 3'. If the image is rotated by $\pi/4$, as is the case in image b), and the angle of rotation is to be estimated as described above, then the angle corresponding to the minimal MSE can take two different values, $\pi/4$ and $5\pi/4$. The same is true for case

c) where the image is rotated by $5\pi/4$. In the above example, the correctness of the solution is not guaranteed, since it is possible that the correct solution is the complementary solution. Fortunately, determining the translation vector provides an opportunity to override this ambiguity.

3.2. Determining the Translation Vector

Once the rotational parameters have been estimated, the translation vector can be determined. Due to the presence of two complementary solutions for the angle of rotation, a method of determining the proper solution is needed. Fortunately eq. (7) provides solutions to both the problem of solution selection, as well as estimation of the translational parameters. The application of eq. (7) is equivalent to a cross-correlation of $\mathrm{Im}_1[\vec{k}]$ with $\mathrm{Im}_2[R\vec{k}]$. The secondary solution for R (which is denoted by R'), will provide a reflection, in addition to the rotation, about the axis of rotation. The solution to the proper solution will provide an impulse, while the complementary solution will provide a non-impulsive response. This section demonstrates the proof for this, albeit for the 1-D case.

Let $x_1[n]$ be a discrete function of the scalar n, and $x_2[n]$ be a discrete function of the scalar n that differs from x_1 by a shift of n_t. This implies that $x_2[n] = x_1[n+n_t]$. To determine n_t, a cross correlation is used, which produces:

$$\sum_{n=-\infty}^{\infty} x_1[n+n_t]x_2[n] = \partial[n+n_t] \tag{10}$$

where $\partial[n+n_t]$ represents an impulsive signal. Similarly, if n is reflected in x_2 the solution is not impulsive in the general case, since:

$$\sum_{n=-\infty}^{\infty} x_1[n+n_t]x_2[-n] \neq \partial \tag{11}$$

The solution in each case will rarely be strictly impulsive, but one solution will be more impulse-like than the other solution (see Fig. 11 on p.50). This enables us to select the solution, based upon whether or not the cross-correlation between $\mathrm{Im}_1[\vec{k}]$ with $\mathrm{Im}_2[R\vec{k}]$ produces an impulse-like function, with the location of the impulse signifying the estimation of the translational parameters.

Chapter 4. Frequency Domain Registration Algorithm

The frequency domain registration problem as stated in chapter 3 provides, in ideal scenarios (infinite memory and time), very precise solutions. Unfortunately in the case of robotic applications, the memory and time resources are finite. Fortunately, in most robotic applications, the registration estimation needs not be all that precise most of the time since collision avoidance needs only to be aware of the region in which obstacles are present, and manipulators are typically compliant enough in nature to allow for small errors in rotation and position. If a high degree of precision is necessary, then the results of this algorithm can be used as the initial condition for the ICP algorithm, or one of its variants. These facts allow for the sacrifice of precision for computational time.

The algorithm as presented is implemented for 3-D data consisting of many different independent fields, which allows for the capturing of different data-types, such as intensity, probability, and colour, in addition to pure Cartesian coordinate range data to improve the estimation of the registration parameters. The development of the algorithm in this thesis has been generalised to the N-D case where applicable, where the only step that needs to be extended from the 3-D case to the N-D case is the axis-angle representation of rotation (see appendix 1). The motivation to go beyond 3-D world into the N-D world is so that additional dimensional parameters, such as time and motion, may be incorporated into the registration estimation process, thereby eliminating the various assumptions made (such as unchanging stationary objects) in typical registration.

4.1. Practical Constraints

When the acquired data is placed on a grid that is less dense than the data collected, which may occur due to non-uniform sampling or memory requirements, several problems occur. The primary problem is the fact that the data points may lie between grid positions, which introduces positional jitter. The secondary problem is that a less dense grid yields a reduced precision for estimating rotation and translation, due to the fact that each grid component occupies more space than it would in a more dense grid. Finally noise will greatly affect the estimates of the rotation and translation, in particular that of translation, as the phase of the frequency domain will not be a continuous function.

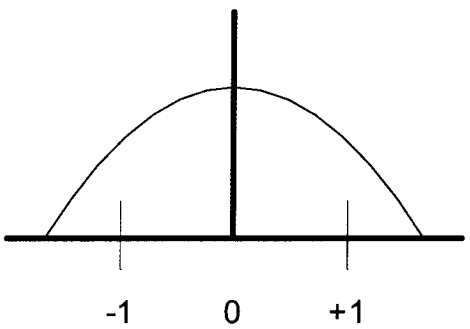

Fig. 4. Sample Pulse

The algorithm minimizes the effects of positional jitter and noise, by applying a pulse shape, similar to that shown in Fig. 4, to the data. The pulse effectively broadens the region where a particular datum is located (so that a datum located between grid points has a value at surrounding grid points), as well as acting as a low-pass filter. The only requirement as to the specific shape of the pulse is that it be low-pass and wide

enough to spread the datum to the surrounding grid points. An investigation into the best pulse shape is presented in the experimental section. The algorithm minimizes the effect of a lower resolution grid, by introducing sub-voxel precision, based upon the assumption that the magnitude of the frequency domain content will be relatively continuous in nature, and hence can be interpolated without a great loss in accuracy.

4.2. Algorithmic Description

The algorithm is briefly summarised below, with more involved description following. The notation used is similar to the notation presented in chapter 3, with the exception that the index q is used to signify which data value at a particular position is selected. For example, with Q different data-types, then $\mathrm{Im}_1(1,\vec{x})$ would select the first property, and $\mathrm{Im}_1(2,\vec{x})$ would select the second property, and $\mathrm{Im}_1(Q,\vec{x})$ would select the Q^{th} property, at the continuous space location indicated by the range image vector $\vec{x}$.

1) Put the continuous data $\mathrm{Im}_1(q,\vec{x})$ and $\mathrm{Im}_2(q,\vec{x})$ into the discrete grids $\mathrm{Im}_1[q,\vec{n}]$ and $\mathrm{Im}_2[q,\vec{n}]$ respectively.

2) Perform the DFT for $\mathrm{Im}_1[q,\vec{n}]$ and $\mathrm{Im}_2[q,\vec{n}]$ producing $F_{\mathrm{Im}_1}[q,\vec{k}]$ and $F_{\mathrm{Im}_2}[q,\vec{k}]$.

3) Calculate the squared error normalized magnitude percentage difference image $SE[\vec{k}]$ between $F_{\mathrm{Im}_1}[q,\vec{k}]$ and $F_{\mathrm{Im}_2}[q,\vec{k}]$.

4) Search for the minimal weight straight line path that goes through $\vec{0}$ in $SE[\vec{k}]$. This minimal weight path is the estimate for the axis of rotation.

5) Select several locations in F_{Im_1}, and rotate these locations about the axis of rotation, and in a coarse to fine manner, determine the angle of rotation that results in the minimal squared error normalized magnitude percentage difference.

6) Create two new images from de-rotating the continuous version of Im_2 with the 2 possible complementary solutions. Call these images Im_2^a and Im_2^b.

7) Put $Im_1(q,\vec{x})$, $Im_2^a(q,\vec{x})$, and $Im_2^b(q,\vec{x})$ into discrete 1-D grids representing the projections of the data along each dimension. Call these $Im_1[q,d,n]$, $Im_2^a[q,d,n]$, and $Im_2^b[q,d,n]$, with d representing the dimension.

8) Perform the phase-correlation for each q and d. Let $P_1^{2a}[q,d,n]$ be the phase correlation between $Im_1[q,d,n]$ and $Im_2^a[q,d,n]$, and let $P_1^{2b}[q,d,n]$ be the phase-correlation between $Im_1[q,d,n]$ and $Im_2^b[q,d,n]$. This is performed by transforming $Im_1[q,d,n]$, $Im_2^a[q,d,n]$, and $Im_2^b[q,d,n]$ using the Fourier transform into the respective frequency domain equivalents $F_{Im_1}[q,d,n]$, $F_{Im_2^a}[q,d,n]$, and $F_{Im_2^b}[q,d,n]$. $F_{Im_1}[q,d,n]$, $F_{Im_2^a}[q,d,n]$, and $F_{Im_2^b}[q,d,n]$ are then used to perform the phase correlation which is then transformed back to the space domain.

9) Create a single measure of the amplitude over q (which represents the different properties, such as colour and position, in the data set) using the Cartesian length measure. This is accomplished using $P_1^{'2a}[d,n]=\sqrt{\sum_{q=1}^{Q}\left(P_1^{2a}[q,d,n]\right)^2}$ and $P_1^{'2b}[d,n]=\sqrt{\sum_{q=1}^{Q}\left(P_1^{2b}[q,d,n]\right)^2}$.

10) Determine which function between $P_1^{'2a}[d,n]$ and $P_1^{'2b}[d,n]$ is most impulsive, by calculating the ratio of the gain of the highest peak versus the variance. The function containing the highest ratio is the proper solution.

11) Extract the location that corresponds to the highest peak from the proper solution for each dimension. This is the estimation of translation in discrete space.

12) Convert from the discrete estimation of translation to the continuous space estimation.

13) Collect transformations to provide the estimate of registration.

4.2.1. Apply the Sparse Continuous Datasets to Discrete Grids

The application of the sparse continuous datasets to discrete grids is required in order to allow the data to be processed via a FFT. To do this, certain criteria must be met, such as minimizing bandwidth and minimizing spatial jitter. Minimizing bandwidth ensures that the majority of the discrete grid is allocated to the dataset, as opposed to empty space.

To do this, a bounding box is determined for each of the datasets. Let $Q_{Im1\ BB1}$ be the transformation between $R_{Image\ 1}$, the reference frame for image 1, and $R_{Bounding\ Box\ 1}$, the reference frame for bounding box 1. Similarly let $Q_{Im2\ BB2}$ be the transformation between $R_{Image\ 2}$, the reference frame for image 2, and $R_{Bounding\ Box\ 2}$, the reference frame for bounding box 2. As a result of using the bounding boxes as a reference for the data gridding, the bounding boxes become the reference frames from which the registration estimate occurs. Since the datasets must be in an equivalent sized space to allow for a proper comparison, the minimal sized bounding box that can contain both bounding boxes is determined, and used for each dataset (see Fig. 5 for 3-D sample).

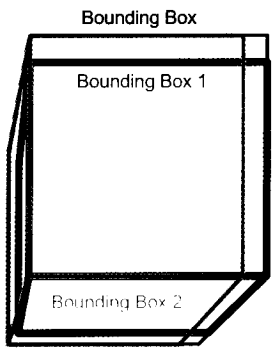

Fig. 5. Selecting a Common Bounding Box

Let B be the diagonal matrix that contains the lengths along each dimension of the common bounding box (see eq.(12)). Then as a result the registration problem can be illustrated as in Fig. 6 (shown for the 3-D case). In Fig. 6, $Q_{Im1\ Im2}$ is the problem to be solved, but it is easier and more efficient for both computation and memory reasons to solve $Q_{BB1\ BB2}$ instead, and then use the known values of QIm1 $_{BB1}$ and $Q_{Im2\ BB2}$ to solve for $Q_{Im1\ Im2}$.

$$B = \begin{bmatrix} Length\ of\ Dimension\ 1 & 0 & 0 & 0 \\ 0 & Length\ of\ Dimension\ 2 & 0 & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & Length\ of\ Dimension\ N \end{bmatrix} \quad (12)$$

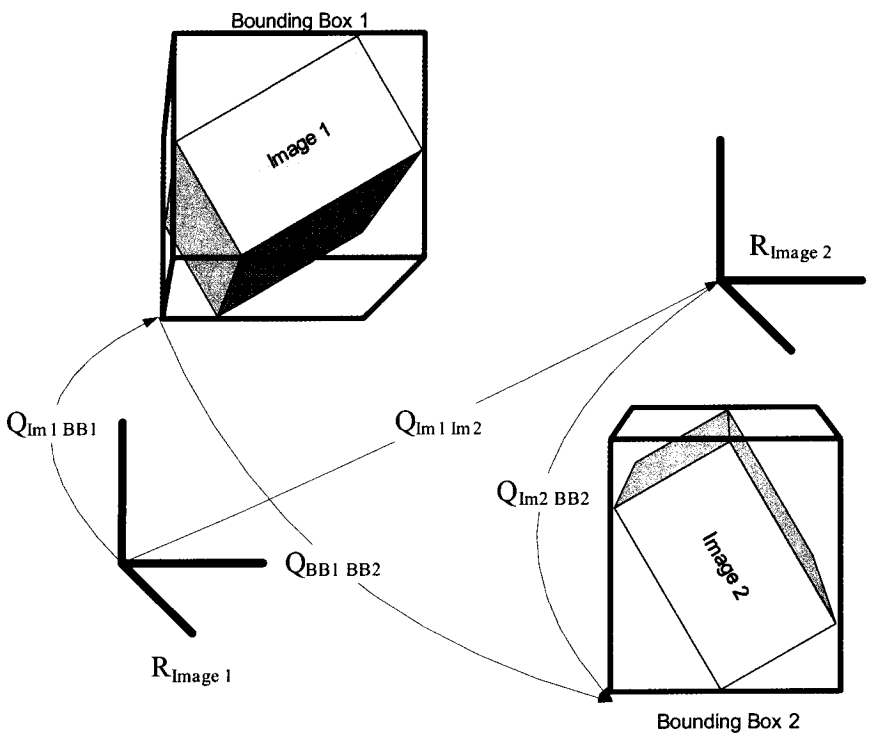

Fig. 6. Bounding Box Reference Frames

After the bounding boxes have been determined, the next step is to determine the grid points. This is done by choosing the maximum resolution along each dimension, which will correspond to the FFT size. To ensure equivalence along each dimension,

cubic grid boxes are used, hence the maximum resolution is assigned to the longest edge of the bounding box, and each other dimension is assigned the same spatial resolution based upon the ratio of the lengths of the edges of the bounding box (see Fig. 7). Some adjustments to the size of the bounding box are made to ensure that the resolution of each dimension of the grid is an integer value. Let M_d be the grid size along each dimension d, and then let the diagonal matrix M contain the reciprocals of M_d.

$$M = \begin{bmatrix} \frac{1}{M_1} & 0 & 0 & 0 \\ 0 & \frac{1}{M_2} & 0 & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \frac{1}{M_N} \end{bmatrix} \tag{13}$$

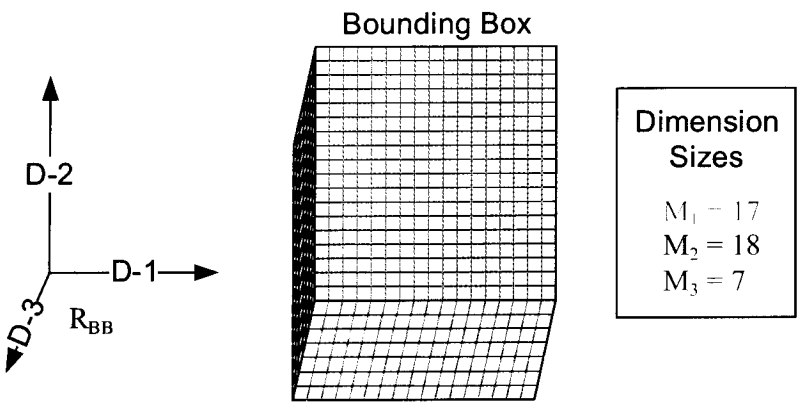

Fig. 7. Creating a Grid Within a Bounding Box

Now that the grid has been defined, the sparse datasets needs to be applied. Since the possibility exists that a data point lies between points on the grid, a method for assigning these points must be performed, while minimizing the spatial jitter effects. The method chosen involves applying a radial symmetrical pulse shape (see Fig. 4) to the data

point, and then setting the applicable grid points to the appropriate values along the pulse shape. This is equivalent to convolving the sparse dataset with a pulse, and then sampling along the grid. Each pulse in 3-D space is given a spherical region of support (hyper-spherical for the multiple dimension case) within the grid, with a standard deviation of one half of a grid length. This is presented mathematically as follows:

$$\mathrm{Im}_1[q,\vec{n}] = \delta\left(\vec{n} - M^{-1}\cdot B^{-1}\cdot Q^{-1}_{\mathrm{Im}1\,BB1}\cdot\vec{x}\right) * \phi(\vec{x}) * \mathrm{Im}_1(q,\vec{x}) \tag{14}$$

$$\mathrm{Im}_2[q,\vec{n}] = \delta\left(\vec{n} - M^{-1}\cdot B^{-1}\cdot Q^{-1}_{\mathrm{Im}2\,BB2}\cdot\vec{x}\right) * \phi(\vec{x}) * \mathrm{Im}_2(q,\vec{x}) \tag{15}$$

where $\phi(\vec{x})$ is the pulse function at $\vec{x}$, $\vec{x}$ is the continuous spatial Cartesian coordinate on the grid, * is the convolution operator, and $\delta()$ is the continuous dirac function that acts as the sampling function, converting continuous locations to discrete locations. Note the presence of M and B in eq.(14) and eq.(15) are used to achieve the proper scaling of $\vec{x}$ so that the grid blocks of the bounding box have equally sized dimensions.

A beneficial side effect of using the pulse is that in addition to spreading out the energy of each datum so that it is sampled and not dropped, it also minimizes possible effects of aliasing since a standard deviation of a half of a grid length in a low-pass pulse function corresponds to the Nyquist frequency of the gridding operation. Several pulse shapes (rectangular, triangular, Gaussian, raised cosine, inversely proportional decay, and sinc pulses) are explored in the experimental section.

4.2.2. Perform the Frequency Domain Transformation

After the datasets have been assigned to their respective grids, the next step is to transform the data on the grids to their frequency domain equivalents. This is performed by utilizing the N-D FFT as described in appendix 3:

$$F_{\mathrm{Im}_1}\left[q,\vec{k}\right] = FFT\{\mathrm{Im}_1[q,\vec{n}]\} \tag{16}$$

$$F_{\mathrm{Im}_2}\left[q,\vec{k}\right] = FFT\{\mathrm{Im}_2[q,\vec{n}]\} \tag{17}$$

4.2.3. Calculate the Difference Image

In order to determine the axis of rotation, the difference between the two magnitudes of the FFT must be calculated. The straight difference is not reliable in practice, since the magnitudes of the FFT may be small, as well as the effects caused by noise in the images may introduce incorrect minimums in the difference function. In [1], a normalized logarithmic difference is used to rectify this problem. While this does work effectively, calculating logarithmic differences are quite processor intensive compared to calculating straight differences. The difference function developed for use in the algorithm is the normalized percentage difference, which ensures that the values with a large relative difference, even when the magnitudes are small, produce a large difference, and that values with a small relative difference, even when the magnitudes are large,

produce a small difference, while having a lower processing cost compared to the normalized logarithmic difference. This difference is defined as:

$$SE\left[\vec{k}\right]=\sum_{q=1}^{Q}\left(\frac{\left|\frac{F_{Im_1}\left[q,\vec{k}\right]}{F_{Im_1}\left[q,\vec{0}\right]}\right|-\left|\frac{F_{Im2}\left[q,\vec{k}\right]}{F_{Im_2}\left[q,\vec{0}\right]}\right|}{MAX\left\{\left|\frac{F_{Im_1}\left[q,\vec{k}\right]}{F_{Im_1}\left[q,\vec{0}\right]}\right|,\left|\frac{F_{Im2}\left[q,\vec{k}\right]}{F_{Im_2}\left[q,\vec{0}\right]}\right|\right\}}\right)^2 \tag{18}$$

In this difference, the frequency domain images are normalized with respect to the zero location, as the zero location provides a good indication of the energy present in the images. The divisor is then chosen to be the maximum of the two points in the difference to ensure that the values fall between zero and one. Finally the Cartesian distance along q is taken as the final measure for each $\vec{k}$.

4.2.4. Find the Axis of Rotation

The axis of rotation can be found as previously stated, by finding the minimal weight line passing through the origin of $SE\left[\vec{k}\right]$. This is accomplished by the following process:

1) Determine the minimal weight path on the sides of a small box centred on the origin of $SE\left[\vec{k}\right]$.

2) Select the point on the side of the small box that corresponds to the minimal weight path.

3) Centre the small box on this point.

4) Calculate the minimal weight path determined based upon the sides of the new small sized box using the following restrictions

   - The point on the edge of the box that corresponds to the minimal weight path must be further from the origin of $SE[\vec{k}]$ than the previous minimal weight path point.

   - The minimal weight path is only calculated in the lower frequency regions ($\leq \frac{\pi}{2}$, or less that half the length along each dimension) to avoid the effects of positional noise that could dominate the high frequency regions.

5) Repeat steps 2) through 4) until the stopping conditions mentioned in 4) are met.

Fig. 8 provides an example (in 2-D) of the search algorithm with the box radius of 2, and the dimensions of the frequency grid being 17x17. The red circle at the centre is the origin of $SE[\vec{k}]$. The green box is the first search box, with the green circle representing the minimal weight path point, which becomes the centre of the next box in blue. The minimal weight path on the blue box, which is further away from the origin than the previous minimal weight path point, is the blue circle. The search algorithm stops at this point since the grid point occupied by the blue circle is at a point corresponding to a

frequency greater than or equal to $\frac{\pi}{2}$ (since the +4 grid point along the horizontal axis is equal to half of the highest discrete frequency value, located at the +8 horizontal grid point).

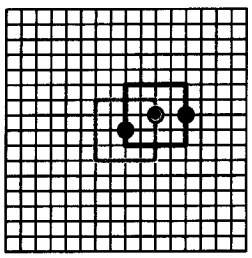

Fig. 8. Search Algorithm Example

The minimal weight path is calculated using line averaging (see eq.(19)), with linear interpolation of the $SE\left[\vec{k}\right]$ function. L is the length of $\vec{k}$ rounded up. The path starting at the origin and ending at the point $\vec{k}$ is divided into L equally sized segments. The linearly interpolated values of $SE\left[\vec{k}\right]$ at each point along the path are summed. The path weight is then normalize with respect to the length of the path (represented by the magnitude of $\vec{k}$) to ensure that paths of different lengths are compared fairly.

$$Path\ Weight\left[\vec{k}\right] = \frac{\sum_{l=0}^{L}\left(SE\left[\frac{\vec{k}}{L}l\right]\right)}{\left|\vec{k}\right|}, L = \left\lceil\left|\vec{k}\right|\right\rceil \tag{19}$$

The size of the box to use is a parameter, and when the size of the box equals half of the grid size, it is equivalent to performing an integral along the radius of the grid in

spherical co-ordinates, which is what occurs in [1]. The advantage of using a moving windowed box is that the computation is minimized to within a small segment of the difference image, while maintaining the accuracy of the integral approach.

4.2.5. Determine the Angle of Rotation

The angle of rotation is determined by first selecting several frequency domain locations (the number of locations is determined by a user parameter). These points are selected to be equally distributed between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ for each dimension. This step is performed to minimize the number of computations to be performed, and is equivalent to subsampling the grid.

The angle of rotation is then iterated coarsely between $-\pi$ and π. Due to possible numerical errors made at the various stages of calculation, the full range of $-\pi$ and π is used to determine the optimal angle, as opposed to the minimal required range of 0 and π. The frequency locations selected in F_{Im2} are rotated, followed by calculating the squared error normalized magnitude percentage difference as presented in eq.(20). The same rationale is used for the development of eq.(20) as was used for eq.(18).

$$SE[Angle] = \sum_{\forall \vec{k}} \sum_{q=1}^{Q} \left(\frac{\left| \frac{F_{Im_1}[q,\vec{k}]}{F_{Im_1}[q,\vec{0}]} \right| - \left| \frac{F_{Im2}[q, R(\vec{R}_{Axis}, Angle)\vec{k}]}{F_{Im_2}[q,\vec{0}]} \right|}{MAX\left\{ \left| \frac{F_{Im_1}[q,\vec{k}]}{F_{Im_1}[q,\vec{0}]} \right|, \left| \frac{F_{Im2}[q, R(\vec{R}_{Axis}, Angle)\vec{k}]}{F_{Im_2}[q,\vec{0}]} \right| \right\}} \right)^2 \quad (20)$$

The angle corresponding to the minimal error is selected for the new midpoint in the coarse to fine search. The selected points in F_{Im2} are now rotated more finely between (angle-$\pi/2$) and (angle+$\pi/2$), and again the error is calculated, and the minimal error angle is selected. This task continues until the desired angular precision is reached. The angular search range is cut in half and centred on the minimal error angle from the previous coarser iteration, and the range is divided up according to how many frequency divisions are desired.

Fig. 9 provides an example of the coarse to fine approach using 10 divisions for each step. The red lines indicate the angles for which the error was calculated, while the red circle indicates the minimal error angle for this iteration. The angles for the second level of the coarse to fine approach are shown in green, and for the third level are shown in blue. This, in practice, will proceed until the angular difference between angles at the current level are smaller than the desired threshold.

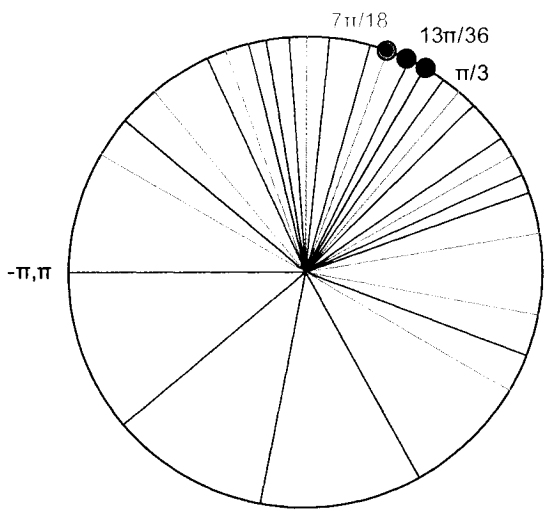

Fig. 9. Coarse to Fine Angle Determination Example

Once the angle has been determined, it should be noted that due to the Hermitian symmetrical nature of the Fourier transform (see appendix 2) there exists a second solution to the angle of rotation that differs by 180° (π radians) from the determined angle (see eq. (21)). The selection of the correct solution is performed in the subsequent sections.

$$Angle' = Angle \pm \pi \tag{21}$$

In Lucchese *et al.* [1], the angle of rotation is determined involving a 1-D correlation technique, after integrating the images along the axis of rotation, and along the radius. These steps are complex and require an additional forward and inverse Fourier transform, as well as determining a maximum of a noisy 1-D function. The coarse to fine approach eliminates the need for the complex correlation technique, and zooms in on the least squares error solution for the frequency locations selected. This ensures that accuracy is maintained, while keeping the algorithm simple and easy to understand. As the number of frequency points increases, so does the accuracy of the algorithm. Also as the number of frequency divisions increases, the precision of the algorithm increases. The computation time of the angle of rotation increases as the previously mentioned parameters increase.

4.2.6. Derotate Im_2 using the Complementary Rotation Solutions

Due to the Hermitian symmetry of the frequency domain, there exists two possible solutions for the rotation. To properly disambiguate the solutions, a phase correlation of

each solution must be performed, as suggested by Lucchese *et al.* in [1]. For this to occur, Im_2 must be derotated by each of the complementary rotational solutions, producing $Im_2^a(q,\vec{x})$, and $Im_2^b(q,\vec{x})$ (see eq.(22) and eq.(23)). This step requires going back to the space domain, due to the phase discontinuities present in the frequency domain from the sparse data sets.

$$Im_2^a(q,\vec{x}) = Im_2\left(R^{-1}\left(\vec{R}_{Axis}, Angle\right)\cdot\vec{x}\right) \tag{22}$$

$$Im_2^b(q,\vec{x}) = Im_2\left(R^{-1}\left(\vec{R}_{Axis}, Angle-\pi\right)\cdot\vec{x}\right) \tag{23}$$

Fig. 10 shows the new scenario where Im_2^a has been derotated by the correct solution, and Im_2^b has been derotated by the complementary solution. The common bounding box, determined in section 4.2.1, will be used to apply the grid to the derotated Im_2. The common bounding box is used since the derotated images should have the same dimensions as Im_1, hence $Im_2^a(q,\vec{x})$, and $Im_2^b(q,\vec{x})$ will fit.

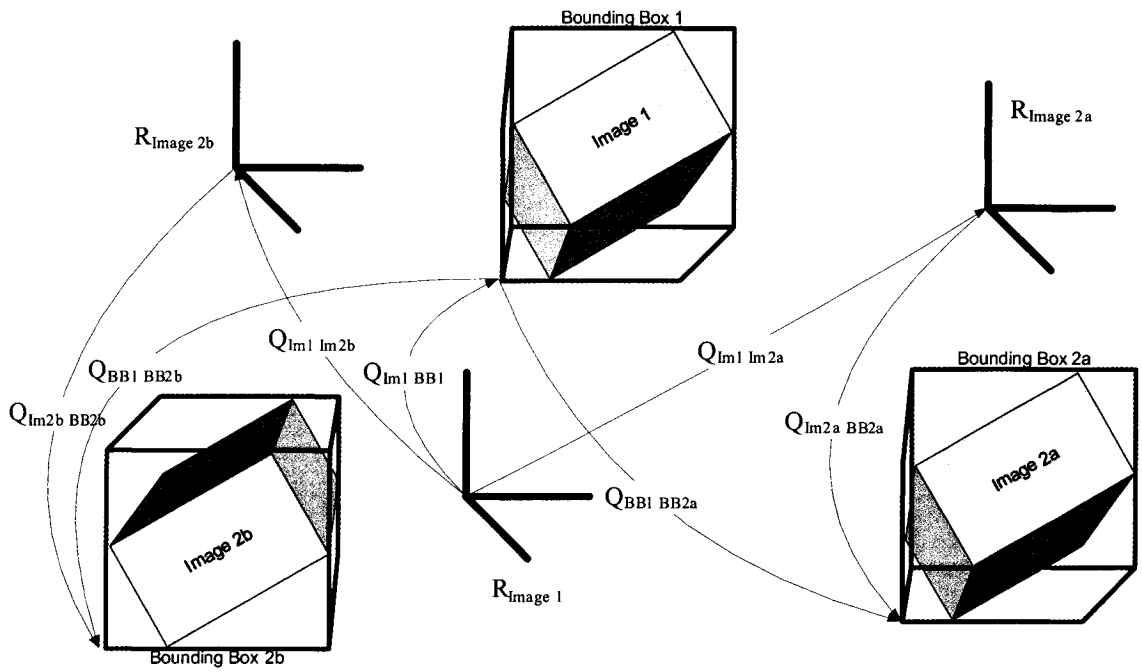

Fig. 10. Two Possible Derotated Image 2 with Reference Frames

4.2.7. Create 1-D Projections of the Datasets along each Dimension

To decrease computational time, 1-D projections are used as suggested by Lucchese *et al.* in [1]. Since these projections will be used in a phase correlation, which will be performed in the frequency domain to improve computation efficiency, the projections need to be placed on a discrete grid. For the same reasons as presented in section 4.2.1, a pulse function is used. This results in:

$$\mathrm{Im}_1[q,d,n] = \delta\left(n - M^{-1}\cdot B^{-1}\cdot Q_{\mathrm{Im1}\,BB1}^{-1}\cdot \vec{x}[d]\right) * \phi(\vec{x}[d]) * \mathrm{Im}_1(q,\vec{x}[d]) \tag{24}$$

$$\mathrm{Im}_2^a[q,d,n] = \delta\left(n - M^{-1}\cdot B^{-1}\cdot Q^{-1}_{\mathrm{Im}2a\,BB2a}\cdot \vec{x}[d]\right) * \phi(\vec{x}[d]) * \mathrm{Im}_2^a(q,\vec{x}[d]) \tag{25}$$

$$\mathrm{Im}_2^b[q,d,n] = \delta\left(n - M^{-1}\cdot B^{-1}\cdot Q^{-1}_{\mathrm{Im}2b\,BB2b}\cdot \vec{x}[d]\right) * \phi(\vec{x}[d]) * \mathrm{Im}_2^b(q,\vec{x}[d]) \tag{26}$$

where n is an index in the 1-D grid, the $[d]$ operator means to select the d^{th} dimensional value.

4.2.8. Perform the Phase Correlation

The phase correlation between Im_1 and $\mathrm{Im}_2{}^a$, and Im_1 and $\mathrm{Im}_2{}^b$ will be performed in the frequency domain, appropriately zero padded to ensure that the correlation performed is not a cyclic correlation. The phase of the Fourier transform as previously stated contains the translation component. Taking advantage of this fact, the phases between image one and the candidates for the correct solution of image two are subtracted, leaving only the phase difference, and after performing the inverse Fourier transform, the translational shift between the two images. This is formally described as follows:

$$P_1^{2a}[q,d,n] = IFFT\left\{\angle FFT\{\mathrm{Im}_1[q,d,n]\} - \angle FFT\left\{\mathrm{Im}_2^a[q,d,n]\right\}\right\} \tag{27}$$

$$P_1^{2b}[q,d,n] = IFFT\left\{\angle FFT\{\mathrm{Im}_1[q,d,n]\} - \angle FFT\left\{\mathrm{Im}_2^b[q,d,n]\right\}\right\} \tag{28}$$

4.2.9. Collapse the Multi-Data Phase Correlation

To take advantage of the extra data provided, if any, the phase correlation will now be collapsed, by using the Cartesian length function along q. This is performed to eliminate the unwanted variable q, so all that remains are the measure of translation (n) for each d dimension.

$$P_1'^{2a}[d,n] = \sqrt{\sum_{q=1}^{Q} \left(P_1^{2a}[q,d,n]\right)^2} \tag{29}$$

$$P_1'^{2b}[d,n] = \sqrt{\sum_{q=1}^{Q} \left(P_1^{2b}[q,d,n]\right)^2} \tag{30}$$

4.2.10. Select the Correct Solution

With the collapsed phase correlation functions now computed, the solution selection process may be determined. The solution corresponding to the correct rotation, will be more impulsive in nature compared to the other solution due to the nature of the correlation (see Fig. 11 for an example). This is performed by determining the summation over each dimension of the ratios of the gain corresponding to the highest peak encountered in the collapsed phase correlation function versus the variance of the phase correlation function:

$$SPGRa = \sum_{\forall d} \frac{\underset{Over\ n}{MAX}\left\{P_1^{'2a}[d,n]\right\}}{\underset{Over\ n}{\text{var}}\left\{P_1^{'2a}[d,n]\right\}} \tag{31}$$

$$SPGRb = \sum_{\forall d} \frac{\underset{Over\ n}{MAX}\left\{P_1^{'2b}[d,n]\right\}}{\underset{Over\ n}{\text{var}}\left\{P_1^{'2b}[d,n]\right\}} \tag{32}$$

The above function was chosen as a good measure of the impulsiveness of a data set, since it provides a direct measure of peak energy compared to the averaged energy of the data. The proper rotational solution corresponds to which of SPGRa or SPGRb is higher. If SPGRa is higher, the rotational solution is $R\left(\vec{R}_{Axis}, Angle\right)$, otherwise $R\left(\vec{R}_{Axis}, Angle - \pi\right)$ is the solution.

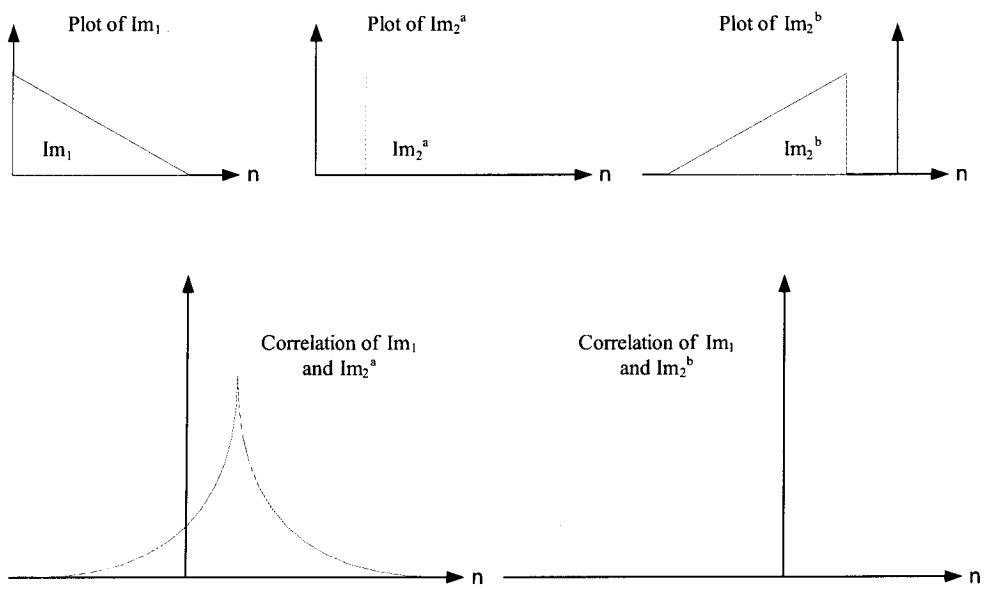

Fig. 11. Correlation to Determine Translation

4.2.11. Estimate the Translation Parameters

The translation parameters correspond to the location of the peaks in each dimension used in section 4.2.10 to choose the correct solution. If SPGRa was the correct solution, then the translation parameters correspond to the location in normal space of the peaks found in $P_1^{'2a}[d,n]$, and conversely if SPGRb was the correct solution then the translation parameters correspond to the location in normal space of the peaks found in $P_1^{'2b}[d,n]$.

4.2.12. Convert the Discrete Translation Parameters to Continuous Translation Parameters.

The conversion from the discrete translation parameters to continuous translational parameters is simply a matter of applying the scaling factor obtained by dividing the bounding box size, by the discrete grid size. This is as follows:

$$T_{Continuous} = M \cdot B \cdot T_{Discrete} \tag{33}$$

where $T_{Discrete}$ contains the discrete translation parameters as found in section 4.2.11, M is as defined by eq.(13), and B is as defined by eq. (12).

4.2.13. Collect the Transformations

To obtain the complete registration estimate, the collection of the various transformations must occur. This includes combining the translation and rotational

parameter estimates, as well as the transformations to the bounding boxes (see Fig. 12). $Q_{Im2s\ Im2}$ contains only the estimation of the rotational parameters, while $Q_{BB1\ BB2s}$ contains the estimation of the translational parameters. To obtain the complete registration between Im_1 and Im_2 ($Q_{Im1\ Im2}$), eq. (34) must be used.

$$Q_{Im1\ Im2} = Q_{Im1\ BB1} Q_{BB1\ BB2s} Q^{-1}_{Im2s\ BB2s} Q_{Im2s\ Im2} \tag{34}$$

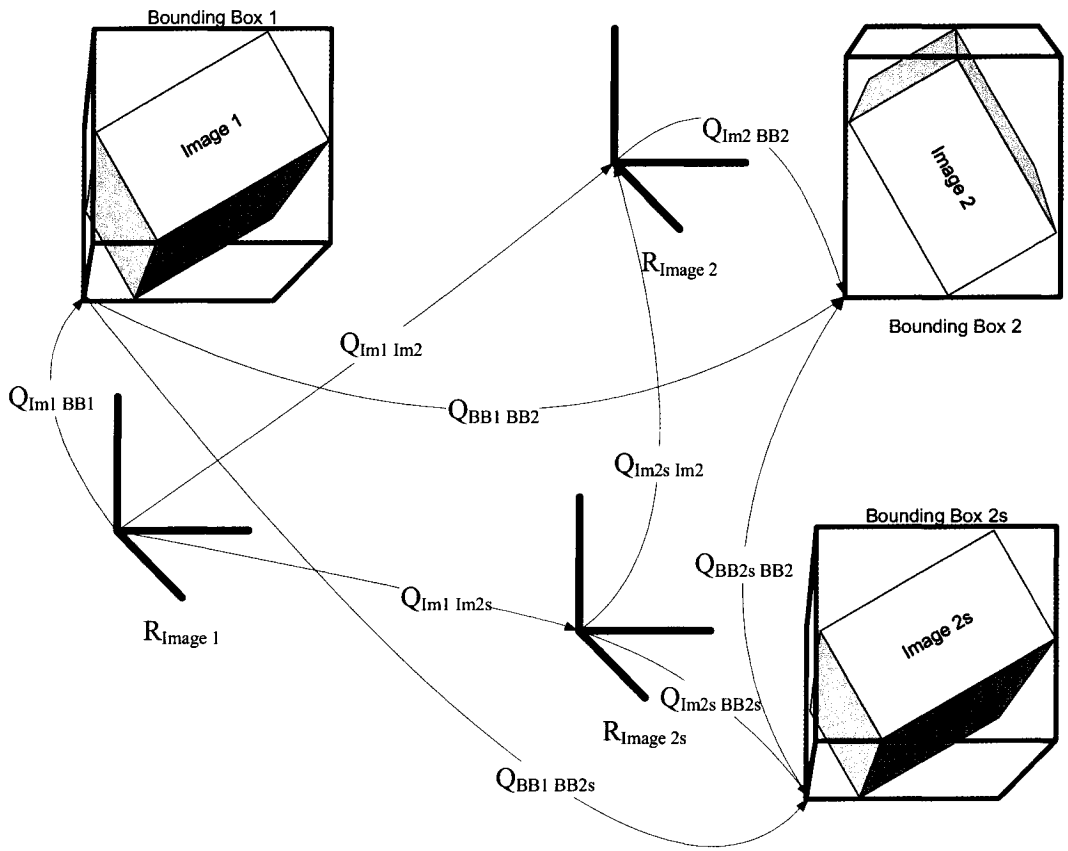

Fig. 12. Collecting Transformations

4.3. Analysis of Error

One of the most important measures of how well an algorithm performs is based upon the stability of results when errors are introduced. There are three distinct sources of error that can affect registration estimation: positional jitter, additive noise, and subtractive noise. This section details the effects of these sources of error on the proposed frequency domain registration algorithm.

4.3.1. Positional Jitter

Positional jitter occurs when data points are offset in position in relation to other data points between images, and express themselves in the frequency domain as phase changes. The application of the shaped pulse as described in sections 4.2.1 and 4.2.7 minimizes the affect of positional jitter, by spreading out the energy of the data points, thereby reducing the amplitude of the phase changes in the frequency images. The effects of positional jitter in both the space and frequency domain are illustrated in eq.(35) and eq.(36), where $J(\vec{n})$ is the amount of jitter at position $\vec{n}$.

$$\mathrm{Im}_1[q,\vec{n}] = \mathrm{Im}_2[q, R\cdot\vec{n} + J(\vec{n}) + T] \tag{35}$$

$$F_{\mathrm{Im}_2}\left[q, R\vec{k}\right] = F_{\mathrm{Im}_1}\left[q,\vec{k}\right] e^{-j2\pi\left(R\vec{k}\right)^T M(J(\vec{n})+T)} \tag{36}$$

As illustrated in eq.(36), positional jitter does not greatly affect the magnitude of the frequency domain, and as a result, does not affect the rotational estimate. However,

the solution selection and the translation estimation will be affected. If the positional jitter is small, relative to the number of points, than the translation estimate will still be accurate, as well as the solution selection.

4.3.2. Additive and Subtractive Noise

Additive and subtractive noise can take on many forms such as features that are present in one image, but not present in the second image (possibly due to non-uniform sampling and occlusion), as well as outliers. The reason additive and subtractive noise are discussed together is that the additional points present in one image, correspond mathematically to those same points being removed or subtracted from another image. These types of errors are commonly encountered when dealing with the overlap problem, and these sources of errors tend to have a greater effect on the estimation of the registration parameters compared to positional jitter. Let Im_a be the additional points added to Im_1 to obtain Im_2, and let Im_s be the points taken away from Im_1 to obtain Im_2. Then as a result the relationships between Im_1 and Im_2 become:

$$\mathrm{Im}_1[q,\vec{n}]+\mathrm{Im}_a[q,\vec{n}]=\mathrm{Im}_2[q,R\cdot\vec{n}+T]+\mathrm{Im}_s[q,R\cdot\vec{n}+T] \tag{37}$$

$$F_{\mathrm{Im}_2}[q,R\cdot\vec{k}]+F_{\mathrm{Im}_s}[q,R\cdot\vec{k}]=F_{\mathrm{Im}_1}[q,\vec{k}]e^{-j2\pi(R\vec{k})^T MT}+F_{\mathrm{Im}_a}[q,\vec{k}]e^{-j2\pi(R\vec{k})^T MT} \tag{38}$$

It is assumed that for the registration algorithm only the sums presented in eq.(37) are known. This results in the fact that Im_a and Im_s contribute to the error in the

estimation by how much their respective energy distribution varies from that of Im_1 and Im_2 (the overlapping point sets), as well as their relative energy with respect to that of Im_1 and Im_2. If Im_s has a similar energy distribution as Im_2, and Im_a has a similar energy distribution as Im_1 (for example a scenario of different sampling schemes) then the error in estimating the registration parameters will be small, however if the energy distribution of Im_s and Im_a vary greatly from that of Im_1 and Im_2 then the registration errors will be larger. This problem directly translates itself into the common registration problem of how much overlap is needed in order to assure a good registration estimate.

The frequency domain registration algorithm presented requires the estimation of the rotational parameters first, which means that this is where error will be first introduced. Using eq.(38) instead of eq.(5) to formulate eq.(18), the magnitude spectrum becomes:

$$\left|F_{\mathrm{Im}_2}\left[q, R\vec{k}\right]+ F_{\mathrm{Im}_s}\left[q, R\vec{k}\right]\right| = \left|F_{\mathrm{Im}_1}\left[q, \vec{k}\right]+ F_{\mathrm{Im}_a}\left[q, \vec{k}\right]\right| \tag{39}$$

and the normalized percentage difference image becomes:

$$SE\left[\vec{k}\right]=\sum_{q=1}^{Q}\left(\frac{\left|\frac{F_{\mathrm{Im}_1}\left[q,\vec{k}\right]+F_{\mathrm{Im}_a}\left[q,\vec{k}\right]}{F_{\mathrm{Im}_1}\left[q,\vec{0}\right]+F_{\mathrm{Im}_a}\left[q,\vec{0}\right]}\right|+\left|\frac{F_{\mathrm{Im}_2}\left[q,\vec{k}\right]+F_{\mathrm{Im}_s}\left[q,\vec{k}\right]}{F_{\mathrm{Im}_2}\left[q,\vec{0}\right]+F_{\mathrm{Im}_s}\left[q,\vec{0}\right]}\right|}{MAX\left\{\left|\frac{F_{\mathrm{Im}_1}\left[q,\vec{k}\right]+F_{\mathrm{Im}_a}\left[q,\vec{k}\right]}{F_{\mathrm{Im}_1}\left[q,\vec{0}\right]+F_{\mathrm{Im}_a}\left[q,\vec{0}\right]}\right|,\left|\frac{F_{\mathrm{Im}_2}\left[q,\vec{k}\right]+F_{\mathrm{Im}_s}\left[q,\vec{k}\right]}{F_{\mathrm{Im}_2}\left[q,\vec{0}\right]+F_{\mathrm{Im}_s}\left[q,\vec{0}\right]}\right|\right\}}\right) \tag{40}$$

From eq.(40). The following observations can be made:

1) If the number of points contained in non-overlapping parts are small (Im_a and Im_s) compared to the overlapping segments(Im_1 and Im_2), then the error introduced will be small.

2) If the number of points contained in the non-overlapping parts are not equivalent (e.g. the number of points in Im_a is much larger than the number of points in Im_s), then the error introduced will be large due to the relative difference in the overall contribution of the overlapping components.

3) If the numbers of points contained in the non-overlapping parts are significant when compared to the overlapping parts, and the point distributions are significantly different, then a large error will be introduced.

4) If the numbers of points contained in the non-overlapping parts are significant when compared to the overlapping parts, and the point distributions are not significantly different, then only a small error will be introduced.

The estimation of the translational parameters, as well as solution selection, relies on a phase correlation. The phase correlation algorithm can be thought of as aligning the centre position of two images. This is significantly effected by additive/subtractive noise, since as the number of non-overlapping points increases, it shifts the translation parameters away from the proper results. In terms of solution selection, these points will add to the base noise of the phase correlation, thereby affecting the peak magnitude to

variance ratio. If the added point adds significant noise, then the wrong solution could be chosen, yielding an incorrect registration estimate.

4.4. Chapter Summary

The goals of a registration technique directed towards robotics applications are detailed at the start of this chapter. Practical constraints of registration in the frequency domain were discussed, with emphasis on the sampling problem. The frequency domain algorithm was then described in detail, while pointing out improvements made to Lucchese *et al.* in [1]. Finally, an analysis into the effects of different sources of errors was made.

Chapter 5. Experimentation

This chapter discusses the experimental setup that was developed for the primary purpose of providing this thesis with reliable real world range data. In addition, this chapter contains the results of the execution of the registration algorithm for simulated data, and a comparison to real world data. The results of the registration algorithm are presented and evaluated against that of the original ICP algorithm proposed by Besl and McKay [2] and the frequency domain algorithm proposed by Lucchese *et al.*[1].

5.1. Experimental Setup

This section talks about the data acquisition setup involving a laser line scanner, and a stereovision camera pair mounted upon a serial robotic manipulator. The integration of the laser line scanner and the robotic arm, as well as the control software was developed by the author, while the multi-modal software component was developed by C.S. Yang [19]. The described range sensing system is used to acquire real-life raw data of wide workspaces for the evaluation of the proposed registration algorithm. This setup will be, and has been, used to acquire range data scans of objects, as well as multi-modal data acquisition. Firstly, the system components will be individually described. Secondly, the hardware and software design decisions will be discussed, with emphasis applied to the laser range finder part of the system, as this is the source sensor of the real range data being used within this thesis.

5.1.1. System Components

5.1.1.1. Robotic Arm

The robotic arm employed in the range data acquisition system is a F3 manipulator from CRS Robotics [20] that has 6 revolute joints, and is mounted upon a CRS 2 meter track [21]. This 7 degree-of-freedom (DOF) serial robotic arm is controlled via a digital programmable controller, the CRS C500C [22]. The controller has several communications methods, of which the RS-232 asynchronous link is the most desirable, since an additional communications adaptor is not required on the existing external controlling computer.

The software supplied by CRS, Robcomm3, provides a compiler for the RAPL-3 language [23], so that customized applications can be programmed for controlling the robotic arm and track. The RAPL-3 programming language, in particular interest for this application, provides a method for accepting user input over the communication link, and a method for the simultaneous control of all joints that the robot possesses.

5.1.1.2. Laser Range Finder

The laser range finder employed in the range sensing system is a Jupiter laser line scanner manufactured by Servo-Robot Inc. [24]. The Jupiter exploits the well-known synchronized triangulation technology developed at NRC [25] and is able to acquire 256 or 512 points per scan on a single line within a field of view encompassing a 1005 mm field depth, a 108 mm wide near plane, and a 579mm wide far plane, at a depth resolution

of 0.7 mm. The Servo-Robot Cami-Box [26] is used to control the Jupiter laser line scanner and also offers a RS-232 asynchronous link that connects to an external interface.

This sensor has been selected for its relatively large scan range, high resolution, and compactness. Very few commercial range finders offer a sufficient scan range combined with a resolution that is suitable for the application considered [27][28][29], while representing a light compact payload for the manipulator.

The software application developed by Servo-Robot, WinUser, has a functionality that allows the acquisition of raw range profiles, as well as the setting of the various parameters of the Jupiter scanner to suit specific applications. The acquired range data is saved to a proprietary file format (.3dx). However, previous experiments with this software revealed it to be inefficient as the operator had to intervene after each scan to manually save the data acquired. This limitation spurred the development of a new interface based on the RS-232 link to directly manipulate the sensor, acquire the range data from the controller, and then save this information to a more suitable format.

5.1.1.3. Stereovision System

In addition to the laser range finder, a chassis has been constructed that supports a pair of Sony XC-999 CCD cameras, which is a key component of the VRex CAM-3000C product [30]. The Matrox Orion video frame grabber performs the pre-processing required to de-interlace the multiplexed feed supplied by the VRex CAM-3000C into separate left and right channels by using the on-board processor.

5.1.2. System Design

5.1.2.1. Hardware

The integrated system uses a standard PC that is connected to the robotic manipulator controller and the Jupiter scanner controller via two RS-232 asynchronous links as shown in Fig. 13. A BNC coaxial cable couples the stereovision system to the video acquisition card installed in the PC. The laser range finder and the stereovision cameras are mounted on the end-effector of the robotic arm with a dedicated bracket as shown in Fig. 14. This configuration allows synchronization and tracking with high precision of the displacement of the sensor. This in turn allows the ability to request a scan of a profile when the desired pose is reached, and to automatically download, display and save the range measurements in the desired format and location for further processing by other software.

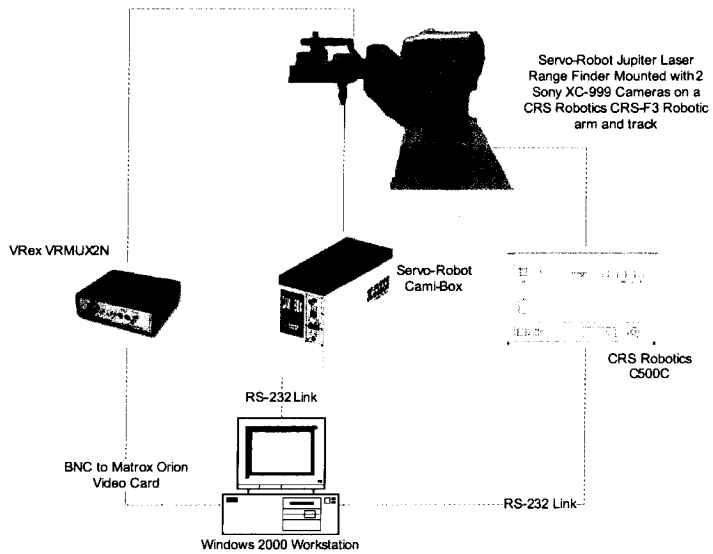

Fig. 13. Diagram of Integrated Range Sensing System

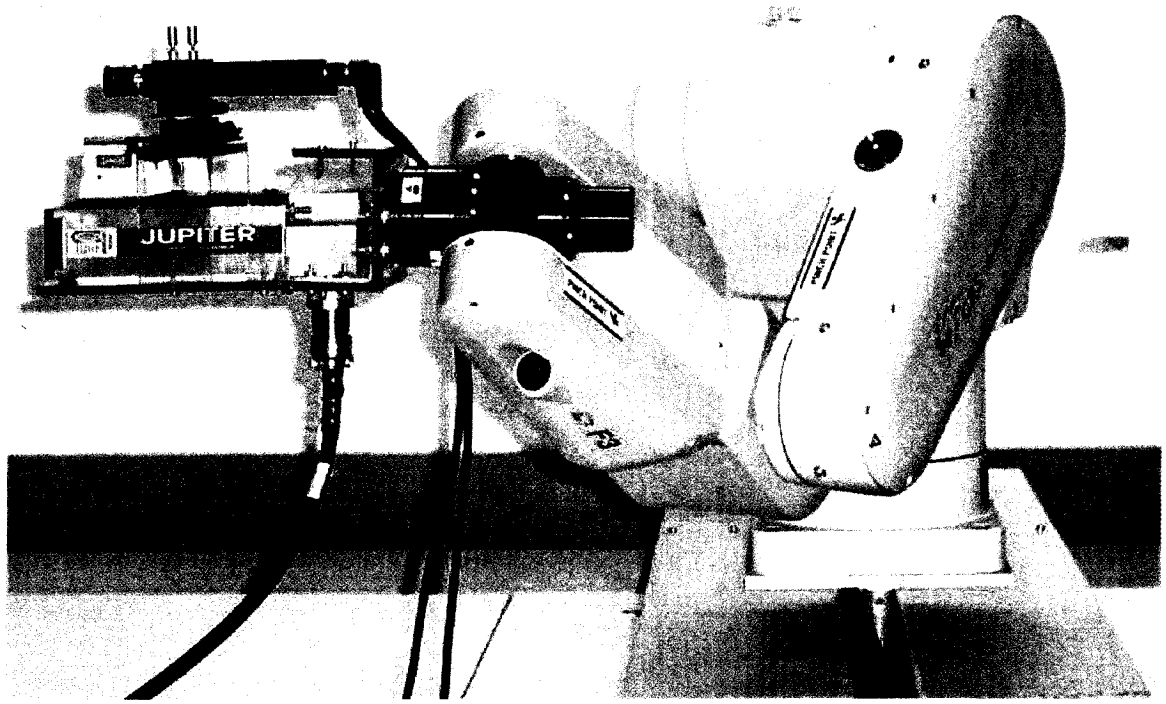

Fig. 14. The multi-modal range sensing system

5.1.2.2. Software

Using the original equipment manufacturer (OEM) supplied software, the robot had to be moved to the desired location through the use of the robot's software interface, and then using the laser range finder's software, the range data was acquired and saved to a separate file. This meant that for a scan of N lines, N different positions were manually calculated by the operator, and N different files were produced following operator's intervention. Additionally to this time consuming process, the files containing the laser line range data were in a proprietary format, unsuitable for direct fusion into a 3-D occupancy model using other software developed in the laboratory.

To overcome these limitations a custom interface was developed to integrate the functionalities of the robot arm, and the laser range finder. Using a custom developed RAPL-3 application that accepts the destination joint positions as input to control the robot arm, as well as a modified inverse kinematics solution inspired from [31] for this specific 7-DOFs manipulator, the robot controlling module of the application was developed.

The module to control the laser range finder relies on low level internal commands provided by the manufacturer to interact with the Cami-box controller [32]. Of particular interest for this application was the implementation of the commands to enable and disable the laser, to activate the acquisition of the data using the scan command and to download the information stored in the sensor's controller memory.

Prior to any use of the robotic multi-modal system for the purpose of acquisition, calibration of the range sensing sensors is required. The calibration mechanism of the stereovision system allows the customisation of the baseline of the stereovision system for different applications. A second calibration is also required between the laser range finder and the stereovision system to ensure that both systems have correlating sampling points [19]. To facilitate the required calibration processes, a graphical interface was implemented to handle both stereovision and multi-modal calibration. Coupled with robot functionality, the Matrox Image Library (MIL) [33][34], and the Open Computer Vision Library (OpenCV) [35], a calibration application was built to automate this process in an accessible pre-allocated robot workspace.

OpenCV provides the necessary intrinsic camera calibration [36] and dense stereovision system disparity algorithm [37] from the left and right images of the stereovision system. In tandem, the stereovision system and the laser range finder are calibrated using the multi-modal calibration approach proposed in [19].

Two separate user-friendly graphical interfaces (GUI) were designed. The first manages the operation of the robot arm and the laser range finder as shown in Fig. 15. This application provides an interface to acquire range data using only the laser range finder from various perspectives within the robotic workspace. The second application manages the data acquisition and calibration of the multi-modal system incorporating the laser range finder, the stereovision system, and the robotic system as shown in Fig. 16. Both GUIs use the same set of libraries developed for controlling the robotic system, the laser range finder, and the stereovision system.

Using the interface shown in Fig. 15, the operator now only has to specify the main parameters for the scan that include the starting position and orientation encoded in a homogeneous transformation matrix (S), a vector defining the magnitude of the steps of the sensor along each direction that is encoded in a homogeneous transformation matrix (D), and the number of steps for the acquisition of the data (N). Using this limited set of parameters, the operator can select the desired resolution and size of the workspace to be measured. The data collected from all positions and orientations visited along the scanning trajectory are then saved to a single file to facilitate further processing and minimize errors. The user may also select to constrict the robot to move each of its joints one at a time, to turn off the laser range finder after each acquisition, and to acquire a

scan through the rotation of the last degree of mobility only, therefore offering various scanning modes.

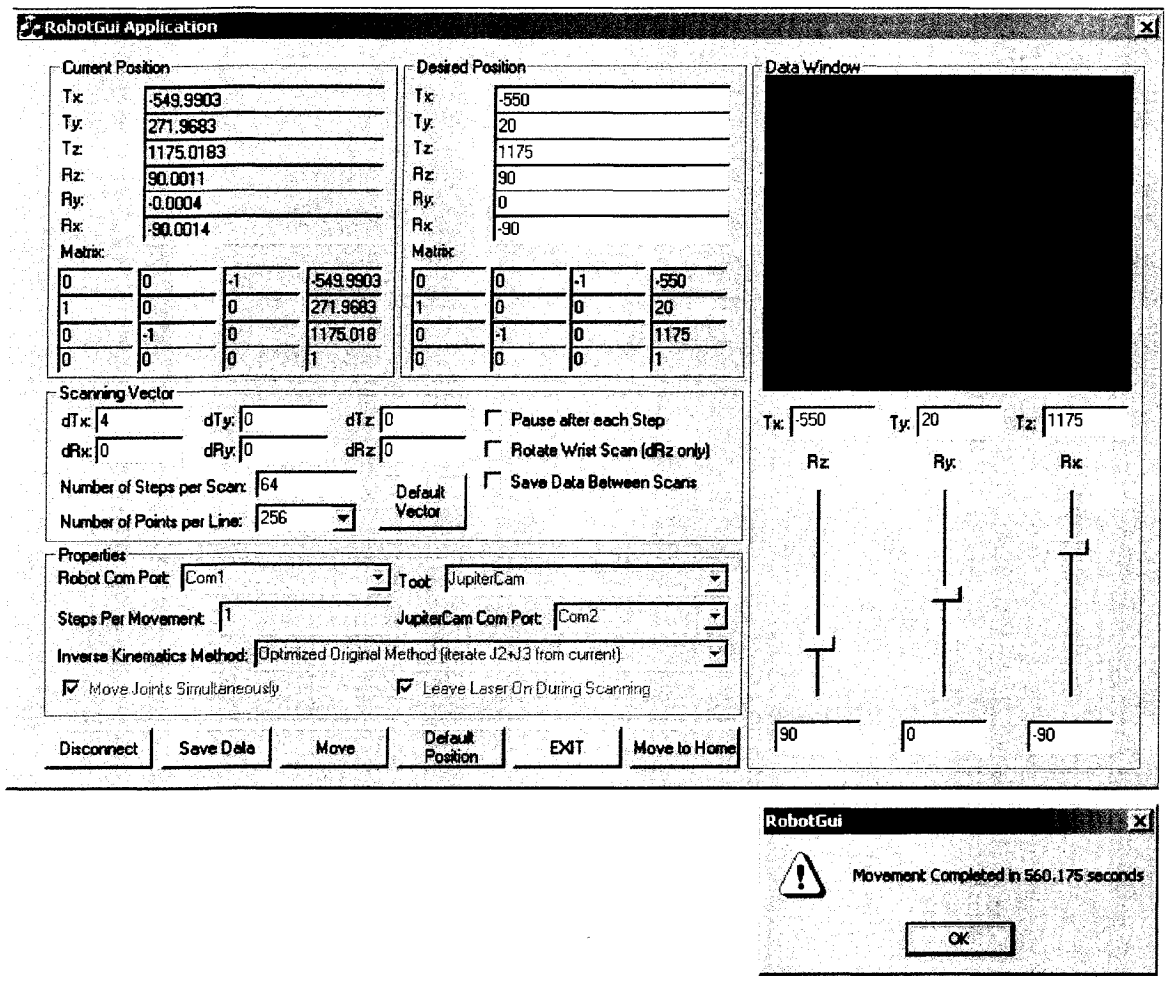

Fig. 15. Robot and laser control application GUI

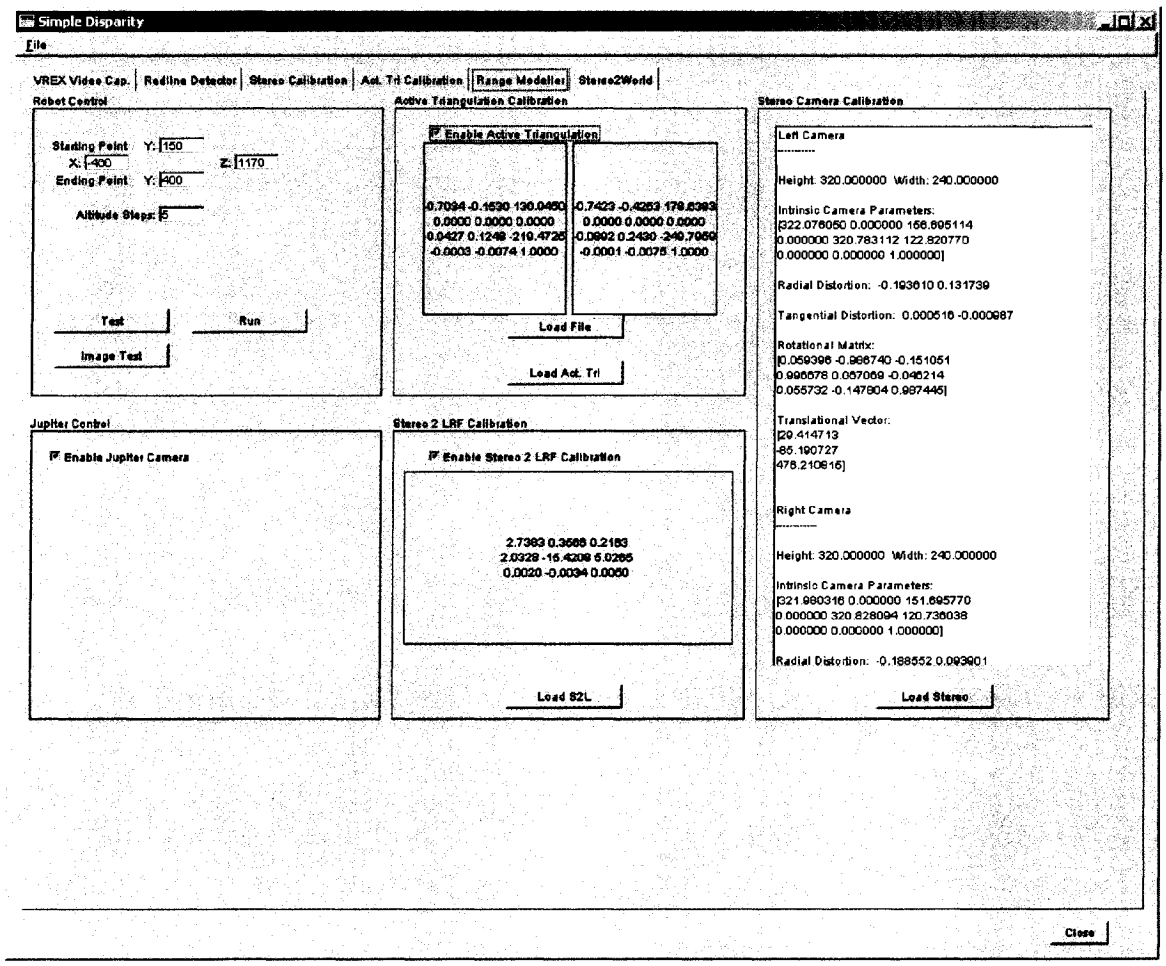

Fig. 16. Multi-modal calibration GUI

5.1.2.3. Integrated System's Operation

Assuming that the integrated robotic system and the multi-modal range sensor have been appropriately calibrated, a typical acquisition sequence using the parameters set by the operator then results in the following steps:

1) The current scan position (P) is determined by the starting transformation matrix (S), and the step transformation matrix (D) multiplied by itself n times, where n represents the current step number and is between 0 and N, such as: $P=SD^n$

2) Once the scan position is computed, the robot inverse kinematics solution is used to generate the 7 joint values.

3) The robot control module then moves each of the joints (sequentially or simultaneously) to their respective angular position.

4) Once the robot is finished moving, the line of data is acquired. The range data from the multi-modal range sensing system is kept in a temporary memory with the corresponding pose of the sensor.

5) Steps 1) to 4) are repeated until $n=N$.

6) Once the scanning trajectory is complete, the operator clicks on a button on the graphical interface to automatically save the entire acquired data to a specified file that is compatible with our 3D modeling software, or to a formatted text file.

5.1.3. Results of the Experimental Setup

Originally, the time to acquire 64 lines of data per viewing area from a series of 3 different viewing areas using the OEM provided set of tools, excluding merging the data into one viable set of data points, was on the order of a day [38]. Repeatability was low, due to the manual driving of the robotic arm by using the supplied command line interface or the pendant tool. Important loss of precision in the registration of the position and orientation parameters was occurring as a consequence of numerous manipulations.

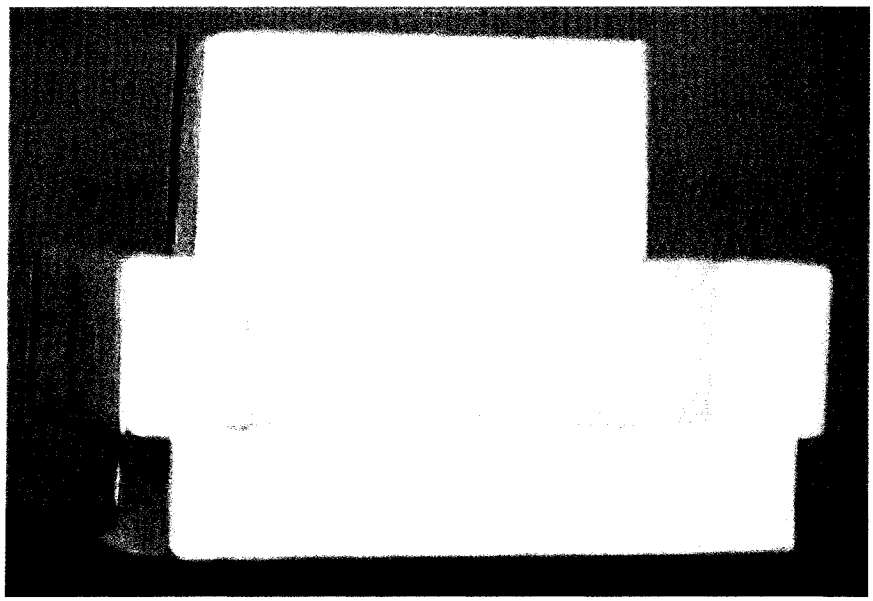

Fig. 17. Mock-up chair

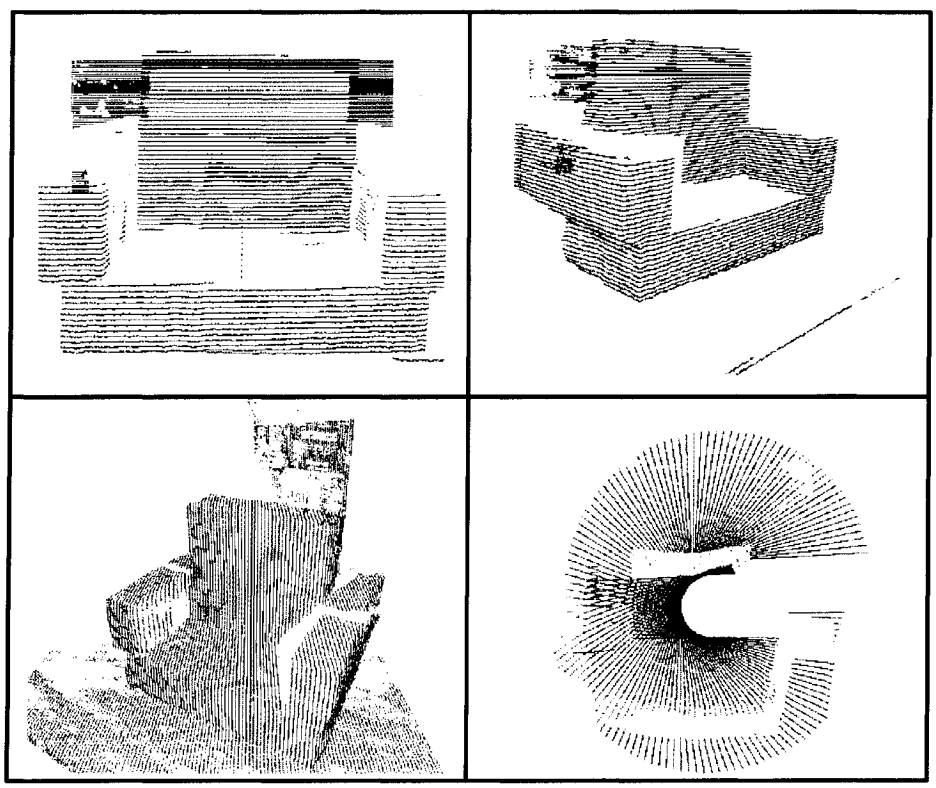

Fig. 18. Laser range finder resulting scans of mock-up chair

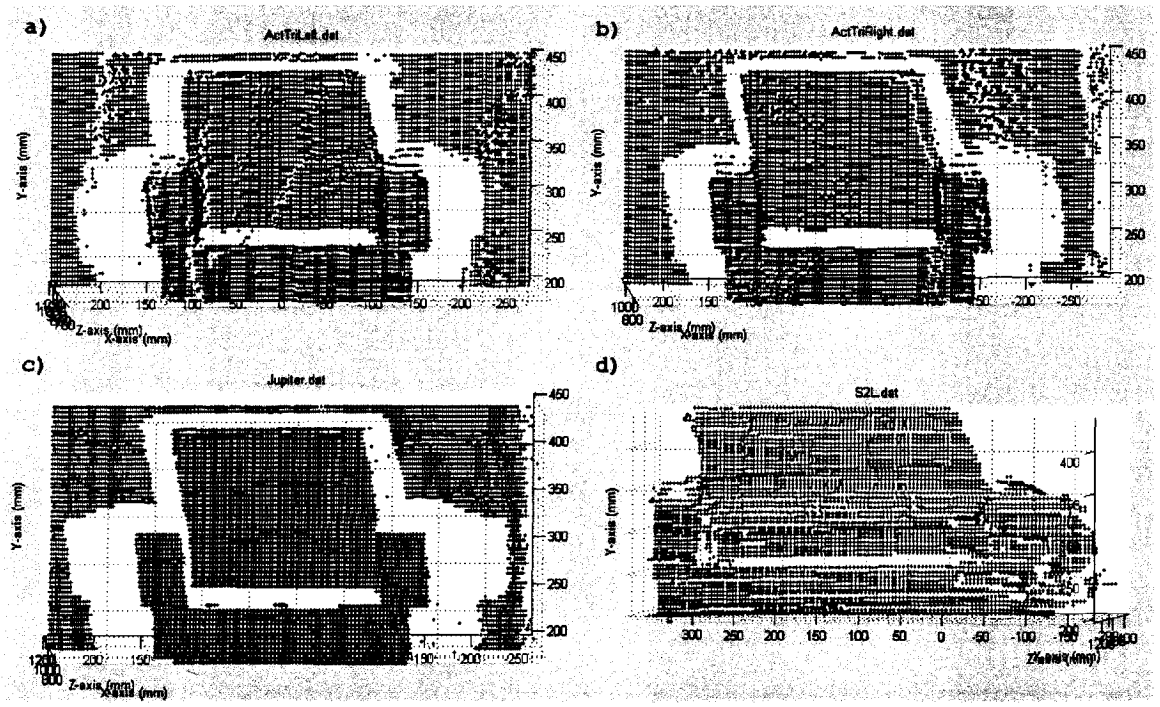

Fig. 19. Scans of the mock-up chair: a) active triangulation using left camera, b) active triangulation using right camera, c) laser range finder, and d) stereovision.

Using the latest version of the integrated multi-modal acquisition setup, the time to acquire the same 64 lines of data per viewing area from 3 different viewing areas is now reduced to about 30 minutes. Precision of the fusion of the multiple lines of range data is ensured by using maximal precision in the registration parameters of the system as provided by the robotic arm. Repeatability is also much improved due to the fact that sensor displacements are now fully programmed and take advantage of the precision of the integrated computer controlled solution. Moreover, four complementary streams of range data are now made available from a single scanning procedure.

Raw data visualization is currently accomplished through both a Windows GUI and Matlab scripts to present the 3D distribution of raw measurements. A sample of range

data points collected from a mock-up chair, shown in Fig. 17, using only the laser range finder and following different scanning patterns is presented in Fig. 18. The two top images, and the bottom left image correspond to positions and orientations of the sensor using a rectilinear scanning pattern from various viewpoints, and the bottom right image corresponds to a circular scanning pattern. The quality of the mapping and the limited number of outliers that are collected can be observed by viewing the data presented in all four images. This is critical in ensuring a proper modeling of complex surfaces on which a telemanipulator would have to interact.

Experimentation has also been conducted to evaluate the potential quality of the integrated multi-modal system. The same data visualization process is used in Fig. 19, which shows four images of the mock-up chair, presented in Fig. 17, against a wall background. Although the sources of all these datasets are from different subsystems within the multi-modal system, they have been transformed such that the views are from the same perspective using the predetermined calibration [19]. This experimentation demonstrates that a proper calibration and operation of the multi-modal acquisition system is achieved.

5.2. Experimental Results

The experiments presented in this section cover several test cases that have been used to validate the proposed registration algorithm, using several data sources. The first two data sources include different images of simulated environments: a frame of a house, and a computer system. The next data source is from a real range acquisition using the model of the house frame upon which the simulated house frame is based. The last data source is an artificially created scene of two skewed planar shapes. The house frame provides a complex and structured environment containing much symmetry for the registration algorithm, while minimizing occlusions. The simulated computer system provides a simple environment, with many occlusions, but little symmetry. The data of the skewed planar objects is used to evaluate the effectiveness of the proposed algorithm with respect to overlap.

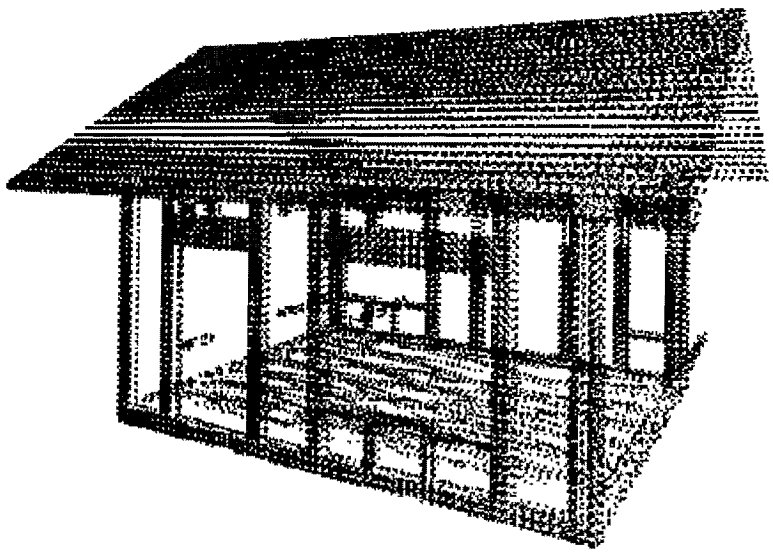

Fig. 20. Point Cloud Corresponding to the Simulated House Frame

Fig. 21. Point Cloud Corresponding to the Simulated Computer

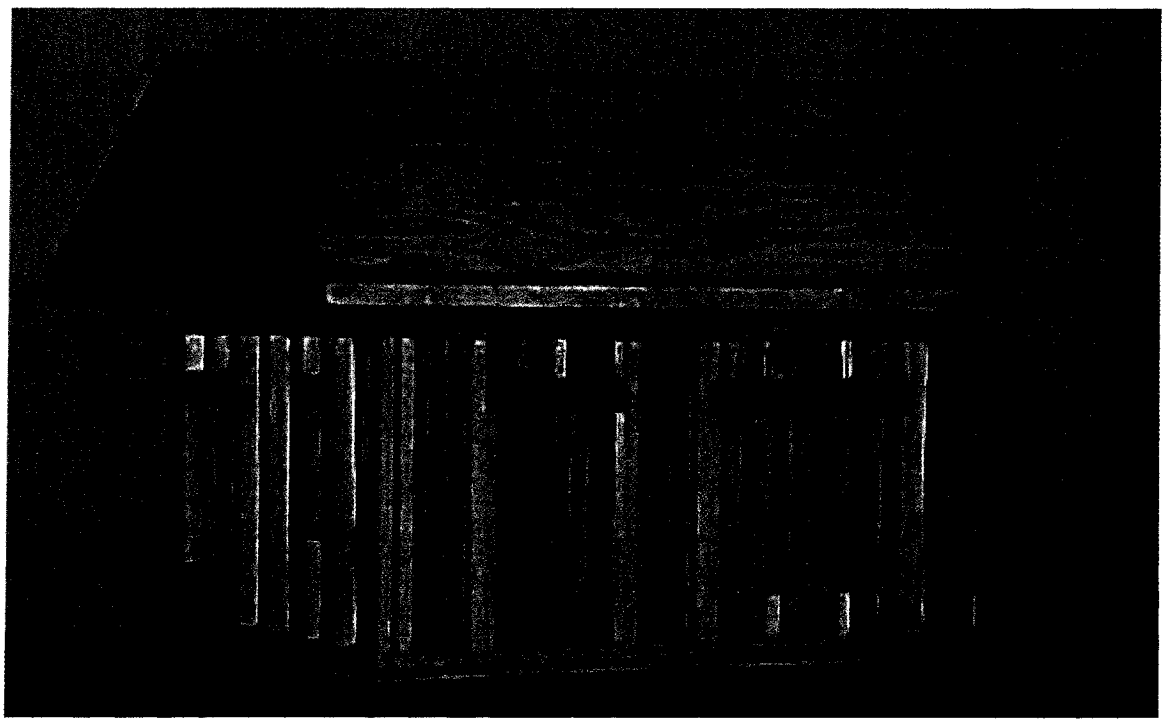

Fig. 22. Actual House Frame

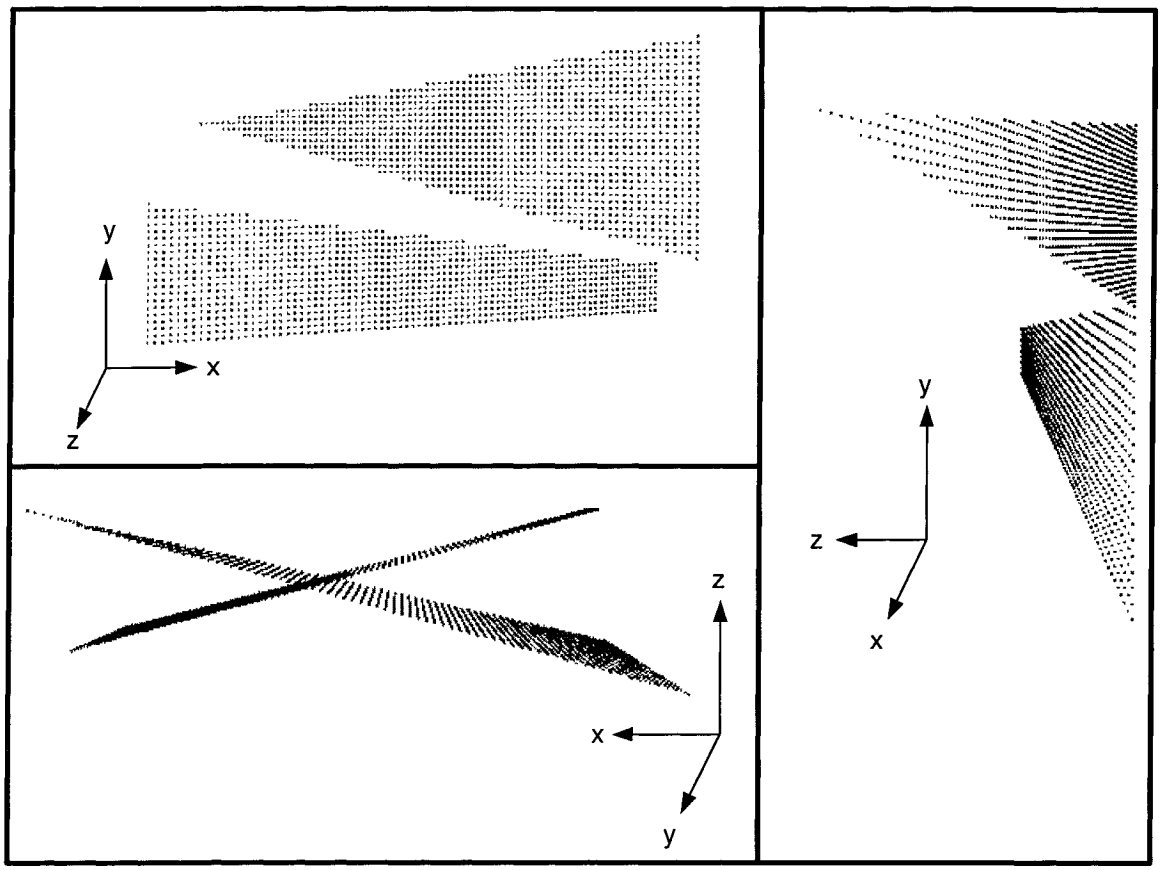

Fig. 23. Skewed Planar Objects Image from Different Viewing Perspectives

The experiments will compare the performance of the algorithm proposed by this thesis, against that of the original ICP algorithm presented in [2], as well as how this algorithm compares against that of Lucchese *et al.* [1]. An evaluation of which pulse shape provides optimal registration estimates, as well as the effect of resolution on registration is performed.

The testing methodology for the simulated house, the simulated computer, and the actual house involves doing a complete pair wise registration within each data source, at different maximum dimensional grid lengths (16, 32, 64, and 128) (see section 4.2.1) and

using a different number of frequency domain locations (400, 800, and 1600) for matching in the angle of rotation determination (see section 4.2.5). The testing methodology for the skewed planar objects involves registering images acquired with a specific calculated overlap, in order to determine experimentally the overlap required in order for the registration algorithm to succeed. The evaluation of the algorithm primarily uses the measure of rotational error as utilised by Lucchese *et al.* in [1], while the measure of translational error is the simple Euclidean distance:

$$\varepsilon_{Rotation} = \left\| I - R_{Actual} R_{Calculated}^{T} \right\| \tag{41}$$

$$\varepsilon_{Translation} = \left\| T_{Actual} - T_{Calculated} \right\| \tag{42}$$

The pulses that are investigated for the algorithm are the Gaussian, inverse decay, raised cosine, rectangular, sinc, and triangular pulses (see Fig. 24). In the computer code for the registration algorithm, the pulses are directly implemented as mathematically stated in the following equations:

$$\phi_{Gaussian}(\vec{x}) = \frac{e^{-\frac{|\vec{x}|^2}{2}}}{\sqrt{2\pi}} \tag{43}$$

$$\phi_{InverseDecay}(\vec{x}) = \frac{1}{1+|\vec{x}|} \tag{44}$$

$$\phi_{RaisedCosine}(\vec{x}) = \begin{cases} \frac{1+\cos(|\vec{x}|\pi)}{2} & |\vec{x}| \le 1 \\ 0 & otherwise \end{cases} \tag{45}$$

$$\phi_{Rectangular}(\vec{x}) = \begin{cases} 1 & |\vec{x}| \le 0.5 \\ 0 & otherwise \end{cases} \tag{46}$$

$$\phi_{Sinc}(\vec{x}) = \frac{\sin(|\vec{x}|)}{|\vec{x}|} \tag{47}$$

$$\phi_{Triangular}(\vec{x}) = \begin{cases} 1-|\vec{x}| & |\vec{x}| \le 1 \\ 0 & otherwise \end{cases} \tag{48}$$

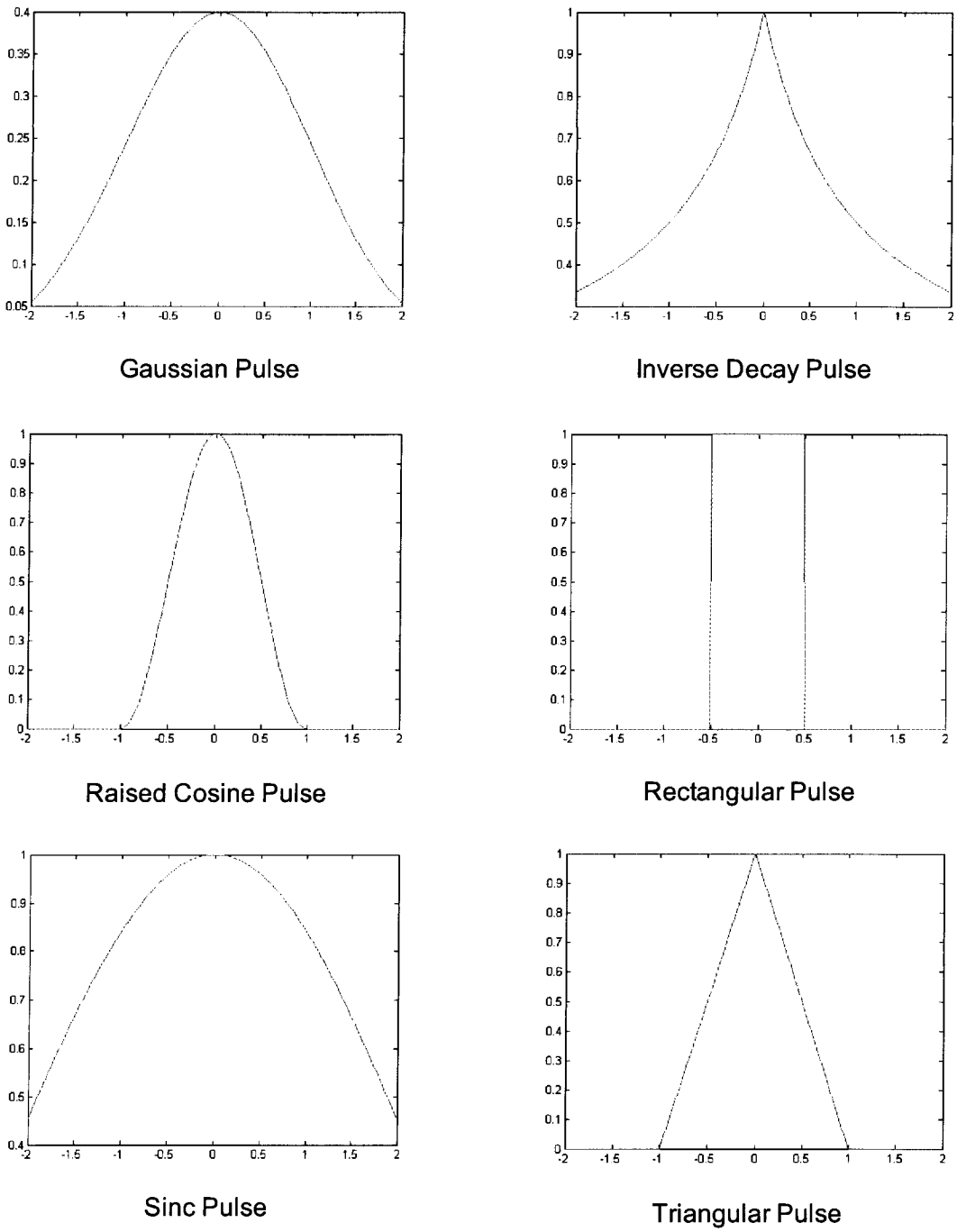

Fig. 24. Plot of Pulses

The analysis of the performance of the algorithm makes heavy use of frequency plots. The frequency plots (see Fig. 25) contained in this thesis count the number of

experimental results for the data set that lie within specific ranges of rotation error (as defined by eq. (41)). The rotation error, which is a unitless quantity, is divided into six different bins with the following ranges: 0.0 to 0.5, 0.5 to 1.0, 1.0 to 1.5, 1.5 to 2.0, 2.0 to 2.5, and 2.5 to 3.0. These bins can be further divided into 3 different categories, correct rotational estimate (0.0 to 0.5), correct rotational estimate except that the wrong solution was selected (2.5 to 3.0), and incorrect rotation estimate (0.5 to 2.5). Through the analysis of the relative quantities in these categories, the performance of the algorithm can be evaluated, with high quantities in the first and second categories (0.0 to 0.5, 2.5 to 3.0) and low quantities in the incorrect rotation estimate categories (0.5 to 2.5) corresponding to good performance. The results for each of the different pulse shapes are shown in a different colour on the same plot. The number of trials performed is the number of perspectives squared, due to the pair-wise matching between all perspectives, so that for the house there are 100 trials (10 perspectives), and for the simulated computer there 784 trials (28 perspectives). Fig. 25 below shows an example of good performance, with a small number of solutions where the incorrect solution was obtained.

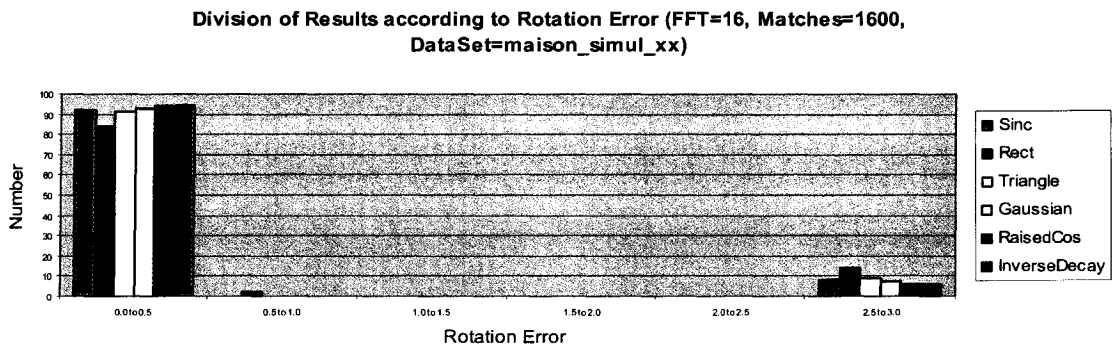

Fig. 25. Sample Frequency Plot of Rotation Error

5.2.1. Simulated House

The ten various perspectives of the simulated house that are used for the testing and evaluation of the registration algorithm are displayed in Fig. 26. As mentioned previously, these datasets have been registered pair wise using the six different pulses, at four different resolutions, and three different point matching levels. To avoid having many pages of results, only a small subset are presented that demonstrate the observed trends.

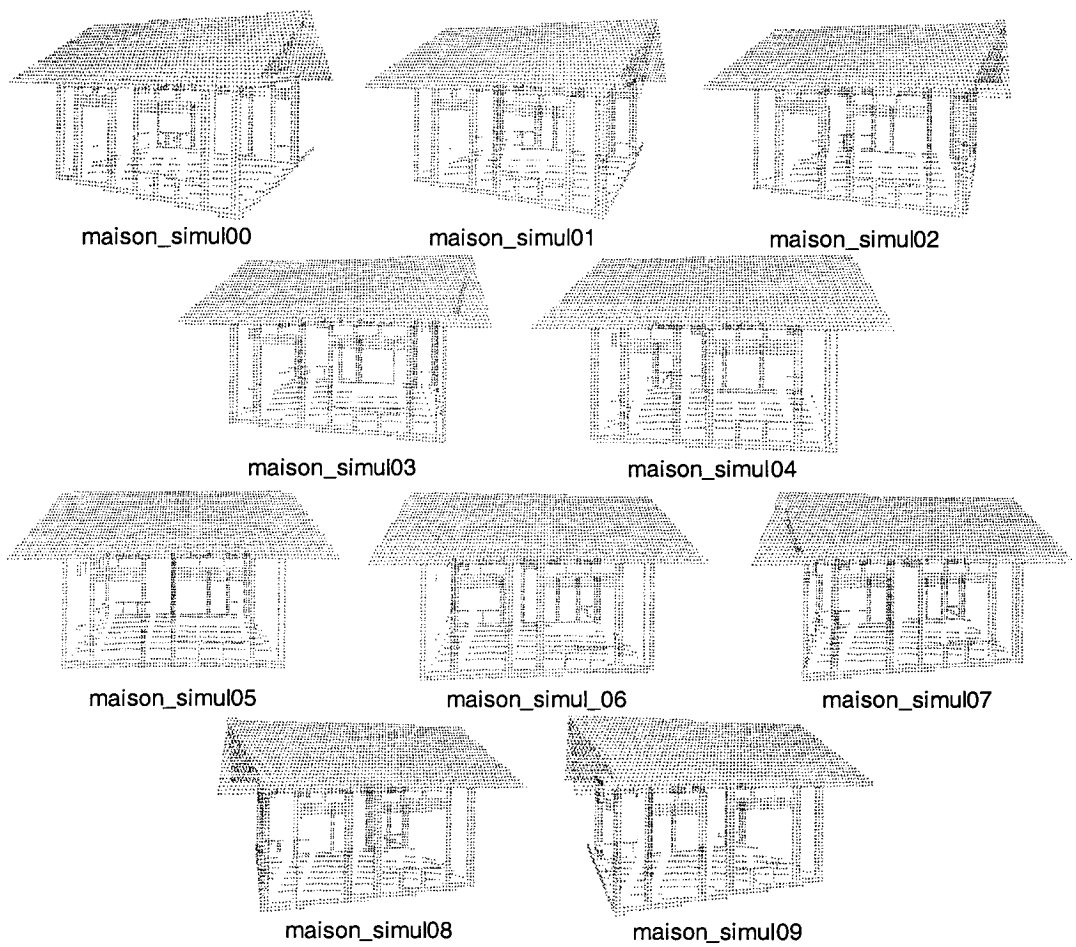

Fig. 26. Point Clouds of Simulated House Range Image Perspectives

The quality of the registration can be observed through the frequency plots of rotation error as defined by eq. (41), for the various pulses, at maximal dimensional resolutions of 16, 32, 64 and 128, and at 1600 frequency domain locations to match in Fig. 28. The rotational errors, as previously mentioned, have been grouped in 6 equally sized divisions. The first category (0 to 0.5) of rotational error corresponds to the correct rotation, while the last category (2.5 to 3.0) corresponds to the correct rotational parameters but incorrect solution selection. Due to the low degree of occlusion present in the simulated house, there is a high overlap present between images; hence this should not have contributed greatly to this solution selection issue. It is interesting to note that contrary to intuition, where one would expect that a higher resolution would produce results with less error, the lower resolution estimation produced the best result. Finally, all of the pulse shapes produce good results, with the exception of the rectangular pulse, which has some significant results in the incorrect solution bins (0.5 to 2.5). Fig. 27 illustrates a correct registration between the maison_simul_00 and maison_simul_02 data sets.

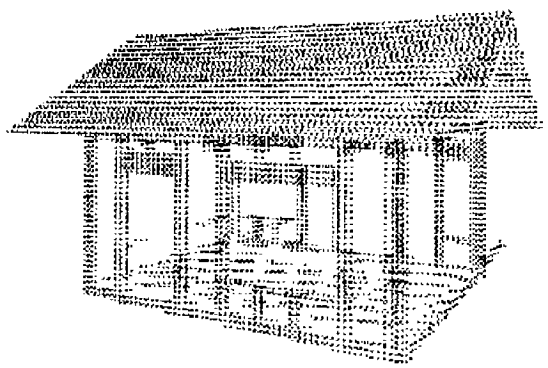

Fig. 27. Point Cloud of maison_simul_00 (red) Registered and Merged with maison_simul_02 (blue)

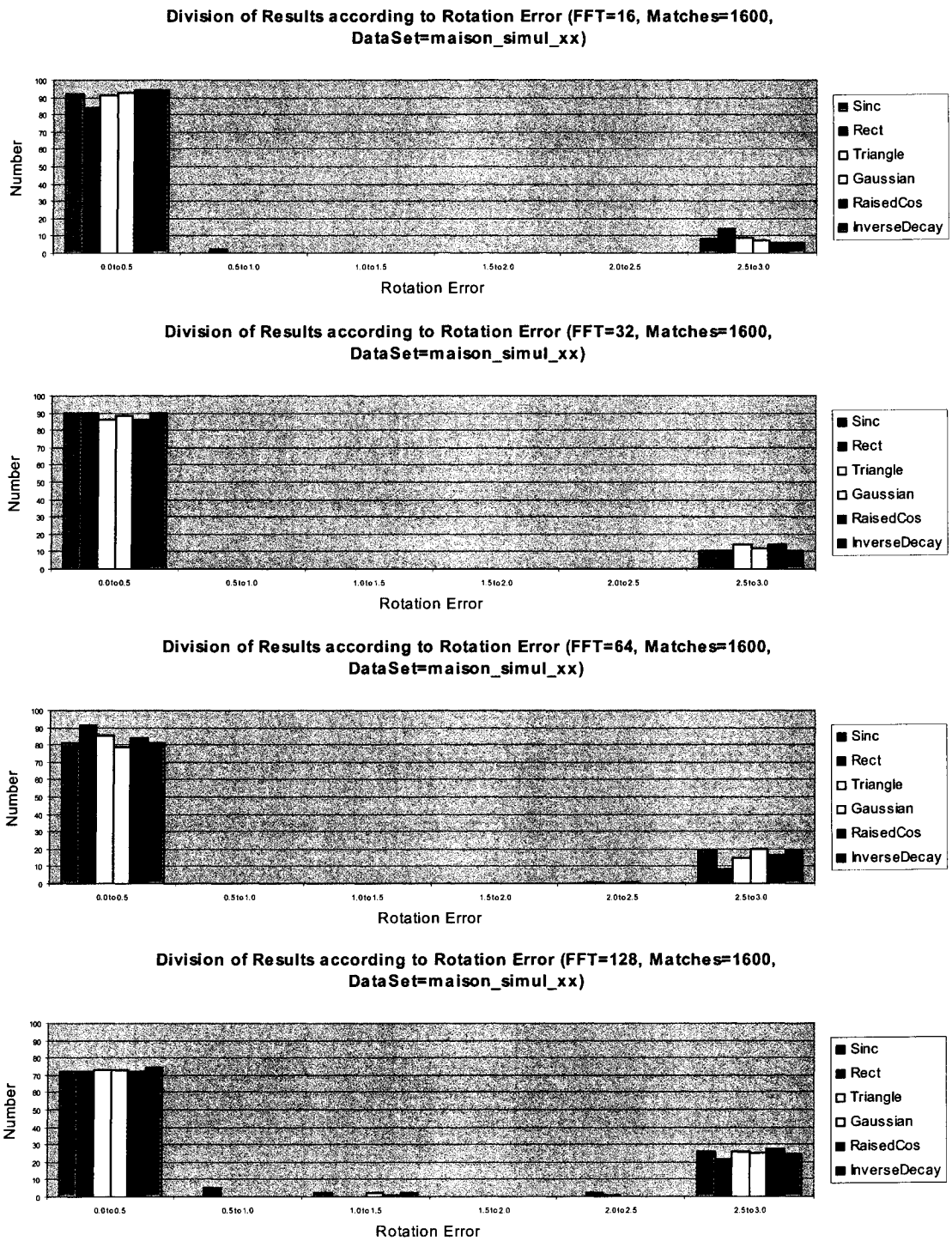

Fig. 28. Frequency Plots of Rotation Error for Simulated House Data Sets (Number of Frequency Locations to Match = 1600)

The results presented in Fig. 29 for a frequency plot of rotation error for maximal dimensional resolution of 64 for the number of frequency domain matches of 400, 800, and 1600 produce the expected trend of reducing the error in the rotation estimate, with increasing the number of points to match, but the unexpected trend of decreasing solution selection accuracy with increasing the number of points to match as is observed by the decreasing amplitudes in the middle four error bins (0.5 to 2.5), and increasing amplitude of the incorrect solution selection bin (2.5 to 3.0).

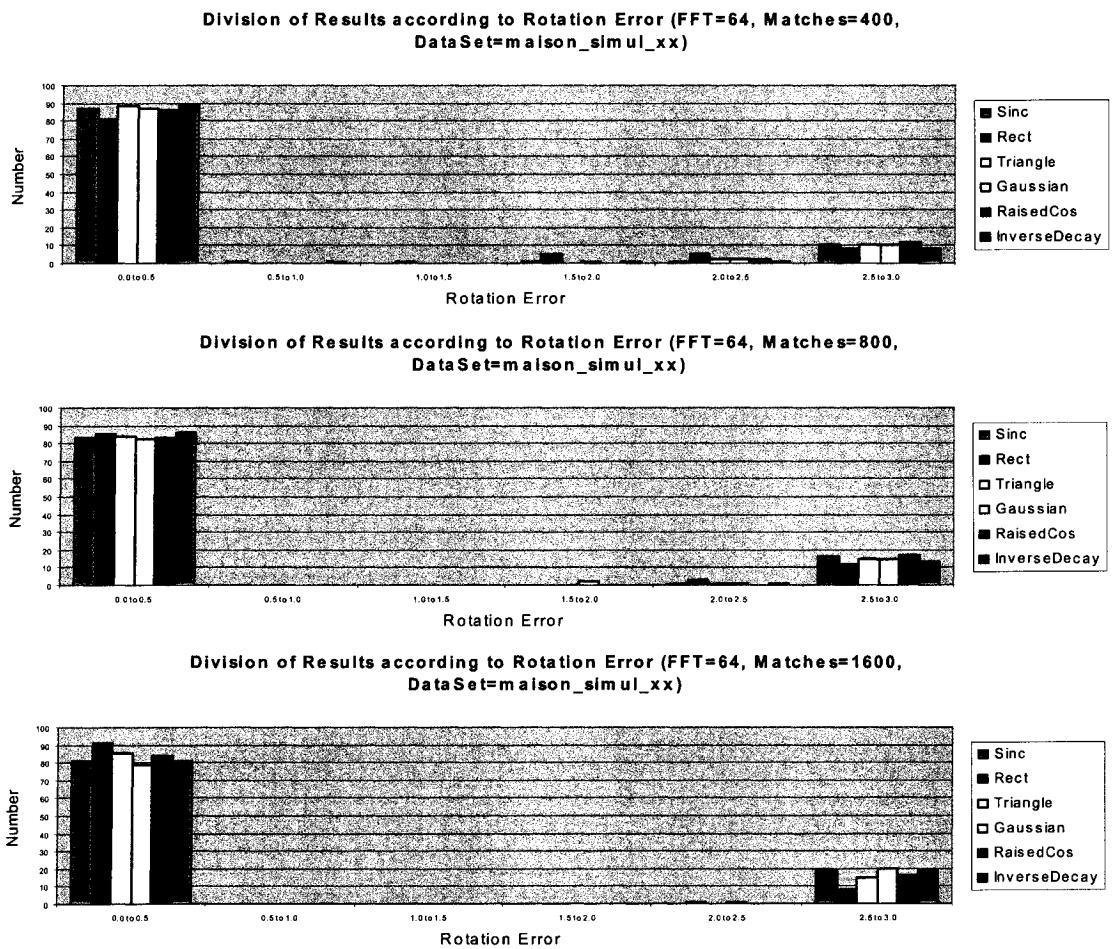

Fig. 29. Frequency Plot of Rotation Error for Simulated House Data Sets (N=64)

The translation error, as is expected from eq. (7), increases with an increase in rotation error, with a similar trend (observed in Fig. 30, which shows both the raw data points, top, and the categorised points, bottom) as presented by Lucchese *et al.* in [1].

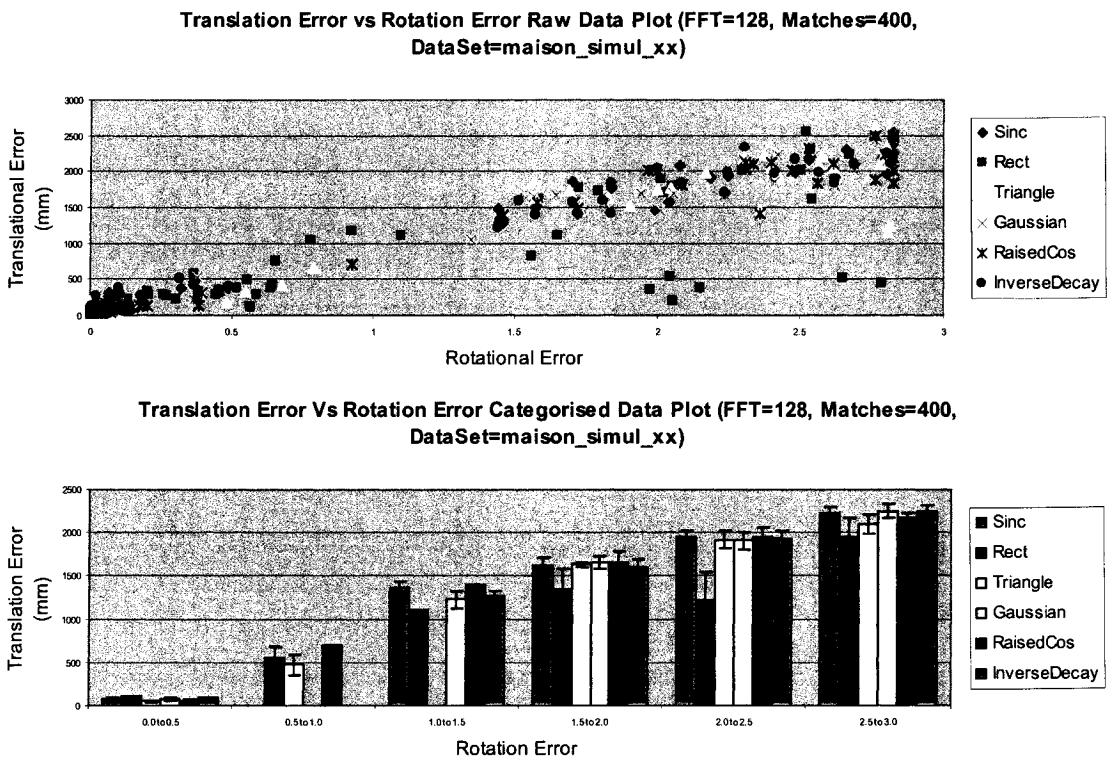

Fig. 30. Translation Error vs. Rotation Error for Simulated House Data Sets

The ICP algorithm, as implemented in this thesis, gave consistently incorrect registration results (errors in the 1.0-1.5, see Fig. 31) range. The reason for this error is probably due to convergence to a more dominant local minimum, which is a known drawback of the ICP algorithm (see [2], as well as chapter 2 in this thesis). As a result, the primary comparison with ICP will be made only with algorithm completion times. The parameters used for the ICP algorithm were the identity matrix for the initial

estimate, a maximum of 15 iterations, and an error threshold of 0.1. Besl and McKay in [2] specify that a precise estimation of the registration typically takes about 20 iterations. Taking into account that 15 iterations should be faster than 20 iterations, as well as the fact that both algorithms are implemented using the same data structures, comparing the execution times is a fair comparison.

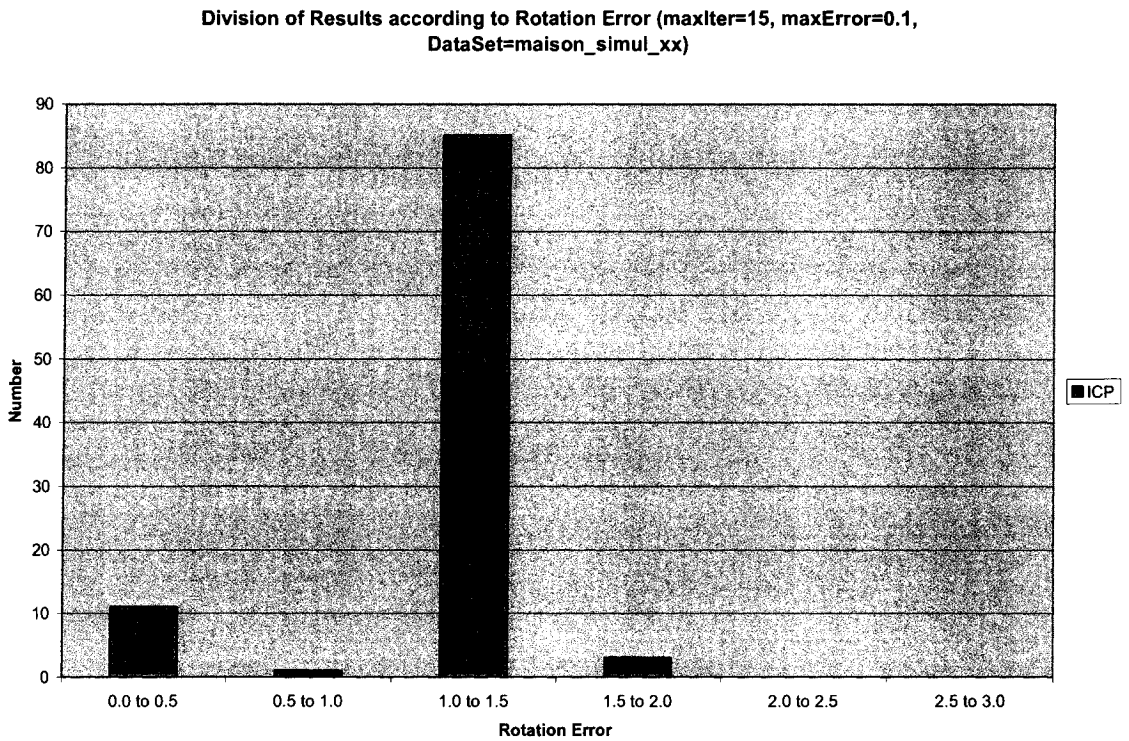

Fig. 31. Frequency Plot of Rotation Error for ICP for Simulated House Data Sets

The completion time of the frequency domain algorithm presented in this thesis wins over the ICP implementation by a huge factor (about 10x) at the smallest parameters set (FFT=16, Matches= 400), and by a factor of 2 at the largest parameter set used. In terms of which pulse produced the fastest results, the rectangular and triangular pulses

proved to be more computationally efficient, while the sinc and Gaussian pulses being the slowest. The expected trends of an increasing computation time with increasing FFT size, and increasing computation time with increasing number of frequency locations to match are also present in Table 1 results.

Table 1. Average Algorithm Completion Times (in Seconds) for the Simulated House

	Sinc	Gaussian	Rectangular	Triangular	RaisedCos	InverseDecay	ICP
FFT=16, Matches=400	5.277	5.2535	5.0387	5.3612	5.0235	5.0624	52.0081
FFT=16, Matches=800	7.7332	7.7414	7.6878	7.6649	7.662	7.7146	52.0081
FFT=16, Matches=1600	13.1953	13.37	13.5598	13.1526	13.5103	13.9128	52.0081
FFT=32, Matches=400	5.3542	5.3078	5.2644	5.0896	5.2323	5.1397	52.0081
FFT=32, Matches=800	7.9206	7.84	7.7811	7.7944	7.7839	7.7693	52.0081
FFT=32, Matches=1600	13.429	13.3972	13.3527	13.346	13.3166	13.2474	52.0081
FFT=64, Matches=400	6.2183	6.2901	6.1523	6.1485	6.4136	6.342	52.0081
FFT=64, Matches=800	9.0288	8.9231	8.8736	8.8502	9.0493	9.5026	52.0081
FFT=64, Matches=1600	14.885	14.6596	14.6373	14.6025	14.8626	14.5214	52.0081
FFT=128, Matches=400	17.348	16.9786	16.9632	16.8624	16.9322	16.972	52.0081
FFT=128, Matches=800	20.5226	20.3368	20.2224	20.2233	20.1857	20.3169	52.0081
FFT=128, Matches=1600	27.681	25.8461	25.8061	25.7468	25.8349	25.7467	52.0081

5.2.2. Simulated Computer

The twenty-eight various perspectives of the simulated computer that are used for the testing and evaluation of the registration algorithm are displayed in Fig. 32. As mentioned previously, these datasets have been registered pair wise using the six different pulses, at four different resolutions, and three different point matching levels. To avoid

having many pages of results, only a small subset are presented that demonstrate the desired trends.

The simulated computer has many images that will not register due to there being no or very little overlap (eg. CPU_02 (left rear corner) and CPU_05(right forward corner)). These poor overlap case occur in about half the possible matches, since 13 of the images (00, 01, 02, 07, 08, 09, 14, 15, 16, 21, 22, 23, and 24) are of the left hand corner, which will not register well with the other, more complete images. In addition to the overlap issue, which was not prevalent in the images of the simulated house, there is also more occlusions in the simulated computer image set. This should result in a greater amount of error.

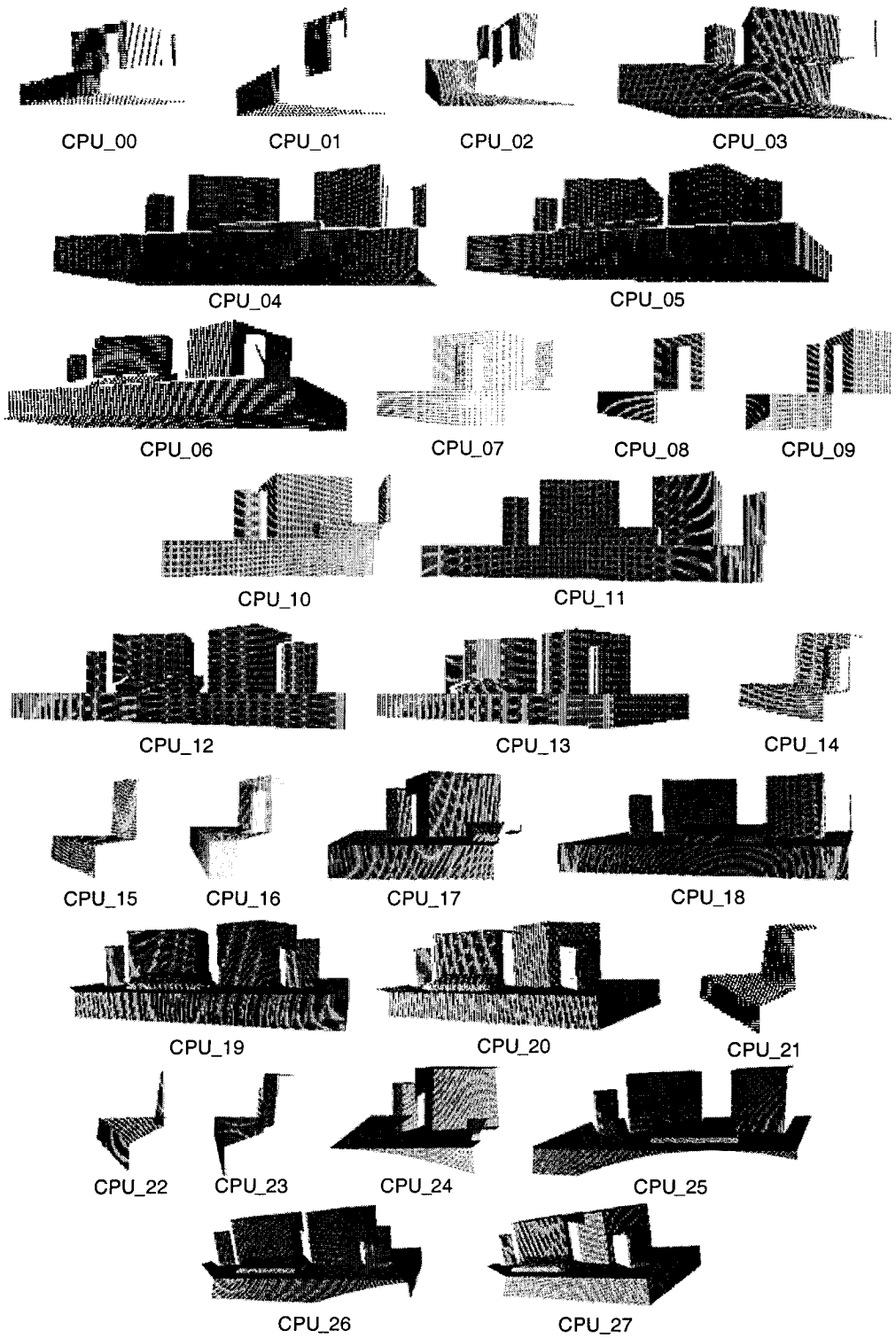

Fig. 32. Point Clouds of Various Perspectives of the Simulated Computer

When the frequency plot of rotation error is made (see Fig. 34), it is easy to see that the inter-registration results are indeed worse as expected. The four error categories in the middle (0.5 to 1, 1 to 1.5, 1.5 to 2, and 2 to 2.5) do not change with variations in resolution, and appear to contain about one half of the results. As was observed with the simulated house, the lower resolution results have less rotation error compared to the higher resolution, as well as the more accurate solution selection (with the exception of the plot showing the maximal dimensional grid length of 16). Again, as was observed previously, the worst performing pulse shape is the rectangular pulse. The other pulses appear to be fairly equivalent and consistent across resolutions. Fig. 33 shows the front and top views resulting from the registration and subsequent merging of the CPU_10 data set with the CPU_04 data set. There is a slight error in rotation, as evidenced by the front view. The error in translation is small, and is present mainly due to the difference in the concentration of points between the two sets: CPU_04 has no points of the right side of the computer model.

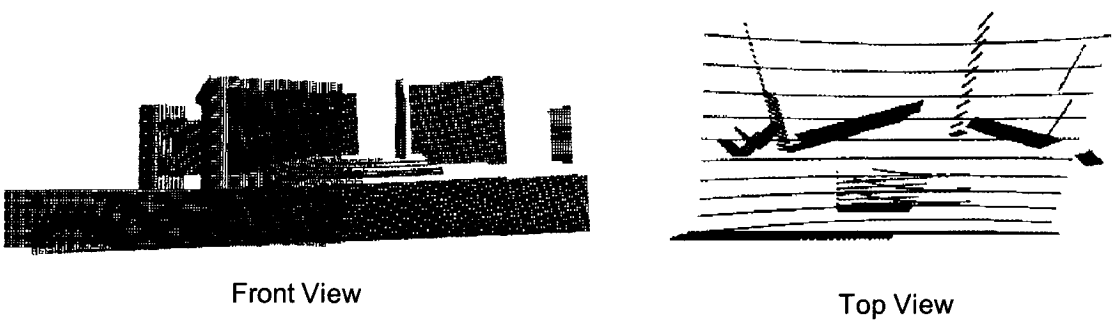

Fig. 33. Point Cloud of CPU_10 (red) Registered and Merged with CPU_04 (Blue)

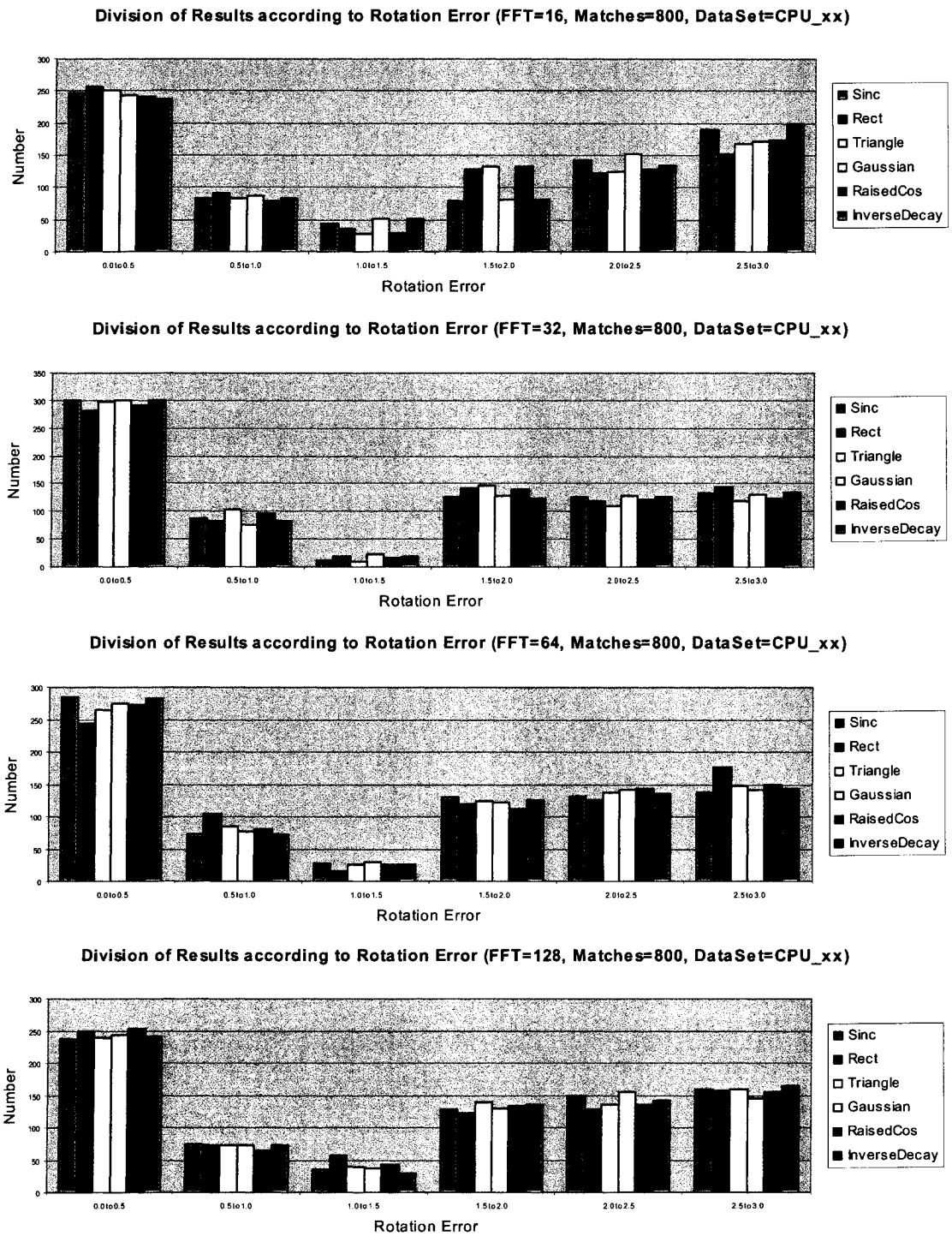

Fig. 34. Frequency Plots of Rotation Error for Simulated Computer Data Sets (Number of Frequency Locations to Match = 800)

The results presented in Fig. 35 for a frequency plot of rotation error for maximal dimensional resolution of 128 for the number of matches of 400, 800, and 1600 produce the expected trend of reducing the error in the rotation estimate, with increasing the number of points to match. As opposed to the trend encountered with the simulated house where there was decreasing solution selection accuracy with increasing the number of points to match, the result for the simulated computer shows increasing solution selection accuracy with increasing number of points to match.

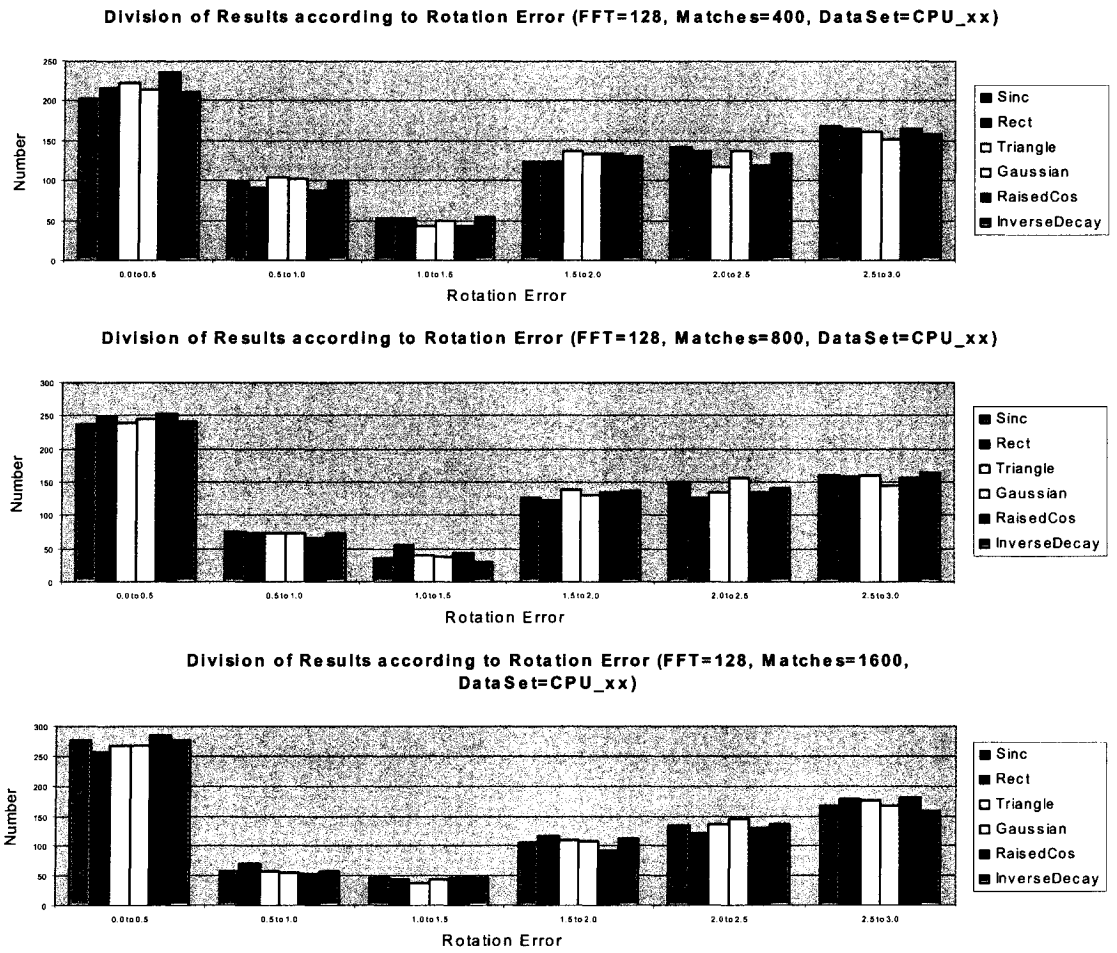

Fig. 35. Frequency Plot of Rotation Error for Simulated Computer Data Sets (N=128)

The translation error increases with an increase in rotation error (see Fig. 36), with a similar trend as presented by Lucchese *et al.* in [1]. This is the same trend as was observed for the simulated house results.

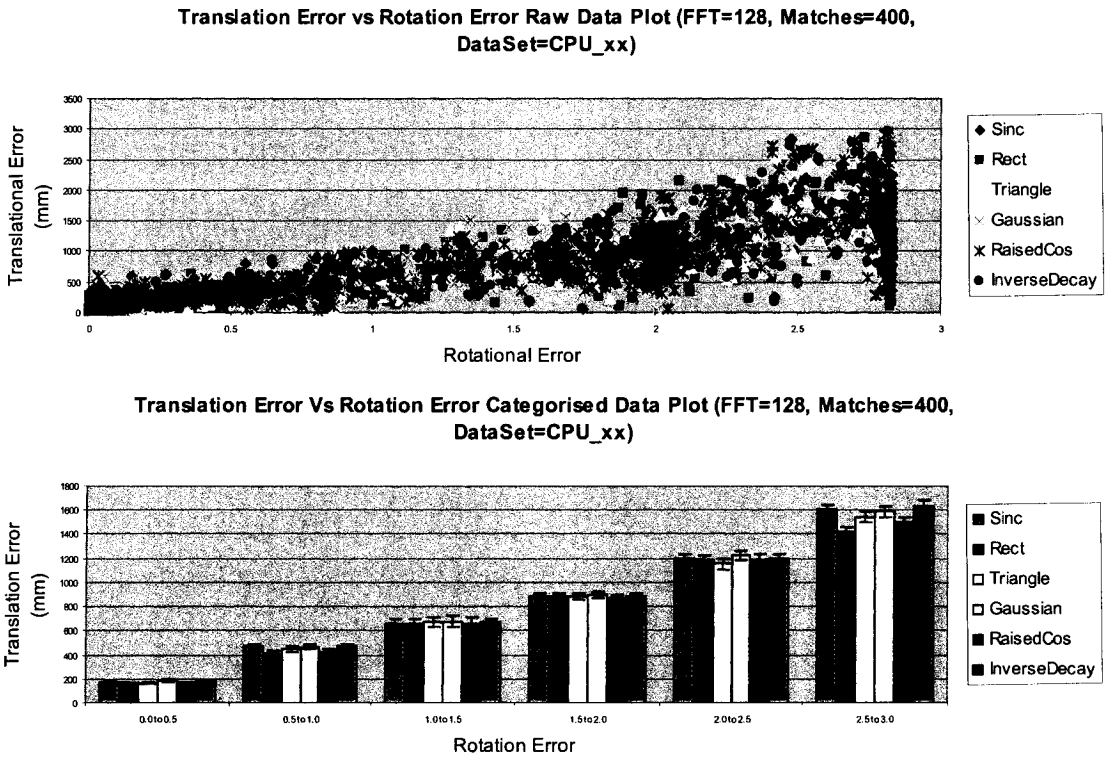

Fig. 36. Translation Error vs. Rotation Error for Simulated Computer Data Sets

The parameters used for the ICP algorithm are the same as what was used for the simulated house, and the same computer system (dual Intel Pentium 600 MHz, 512MB ram, running Windows 2000) was used to produce both sets of results. The completion time of the frequency domain algorithm presented in this thesis wins over the ICP implementation by a huge factor (about 35x) at the smallest parameters set (FFT=16,

Matches= 400), and by a factor of 12 at the largest parameter set used. These are larger factors than what was shown for the simulated house, and can be explained due to the fact that the simulated computer range images have more points than the simulated house images and this has a major impact on the performance of the ICP algorithm. The same trend in results as those acquired for the simulated house is present for the simulated computer, with the rectangular and triangular pulses proving to be most computationally efficient, while the sinc and Gaussian pulses being the slowest. The expected trends of an increasing computation time with increasing FFT size, and increasing computation time with increasing number of frequency locations to match are also present in Table 2 results.

Table 2. Average Algorithm Completion Times (in Seconds) for the Simulated Computer

	Sinc	Gaussian	Rectangular	Triangular	RaisedCos	InverseDecay	ICP
FFT=16, Matches=400	7.06892	6.948852	6.80956633	6.825179	6.8507398	6.82522959	247.1992
FFT=16, Matches=800	9.46606	9.432232	9.29381378	9.312398	9.331926	9.31678571	247.1992
FFT=32, Matches=400	7.18487	7.151403	7.00830357	7.012309	7.0384566	7.0208801	247.1992
FFT=32, Matches=800	9.66013	9.689515	9.65883929	9.728189	9.5606378	9.55068878	247.1992
FFT=64, Matches=400	7.87788	7.780536	7.63497449	7.649605	7.6921173	7.68057398	247.1992
FFT=64, Matches=800	10.6254	10.34401	10.2948342	10.26172	10.287143	10.2677423	247.1992
FFT=128, Matches=400	14.0135	13.81064	13.8759311	14.36361	14.296913	14.4977679	247.1992
FFT=128, Matches=800	15.665	15.65313	15.5156505	15.54842	15.706684	15.5631505	247.1992
FFT=128, Matches=1600	20.7805	20.86643	20.7528189	20.77778	22.109541	21.6756633	247.1992

5.2.3. Actual House

The ten various perspectives of the actual house that are used for the testing and evaluation of the registration algorithm are displayed in Fig. 37. As mentioned previously, these datasets have been registered pair wise using the six different pulses, at four different resolutions, and three different point matching levels. To avoid having many pages of results, only a small subset are presented that demonstrate the desired trends.

The acquired images of the house were chosen to be approximately equal to that of the simulated images, but due to the limitations of the range image acquisition tool the acquired images may contain the background, may have gaps where the house is out of range of the sensor, the house is wider than the field of view of the sensor, the data has a few outliers, and contains noise. Additionally, many more points are present in the actual acquired images as opposed to the simulated images. As a result of the additional sources of error, the registration results are expected to be worse than the simulated house results. The ICP algorithm was not attempted on this set of data, as it contains a much greater amount of points, and would have taken too long to compute.

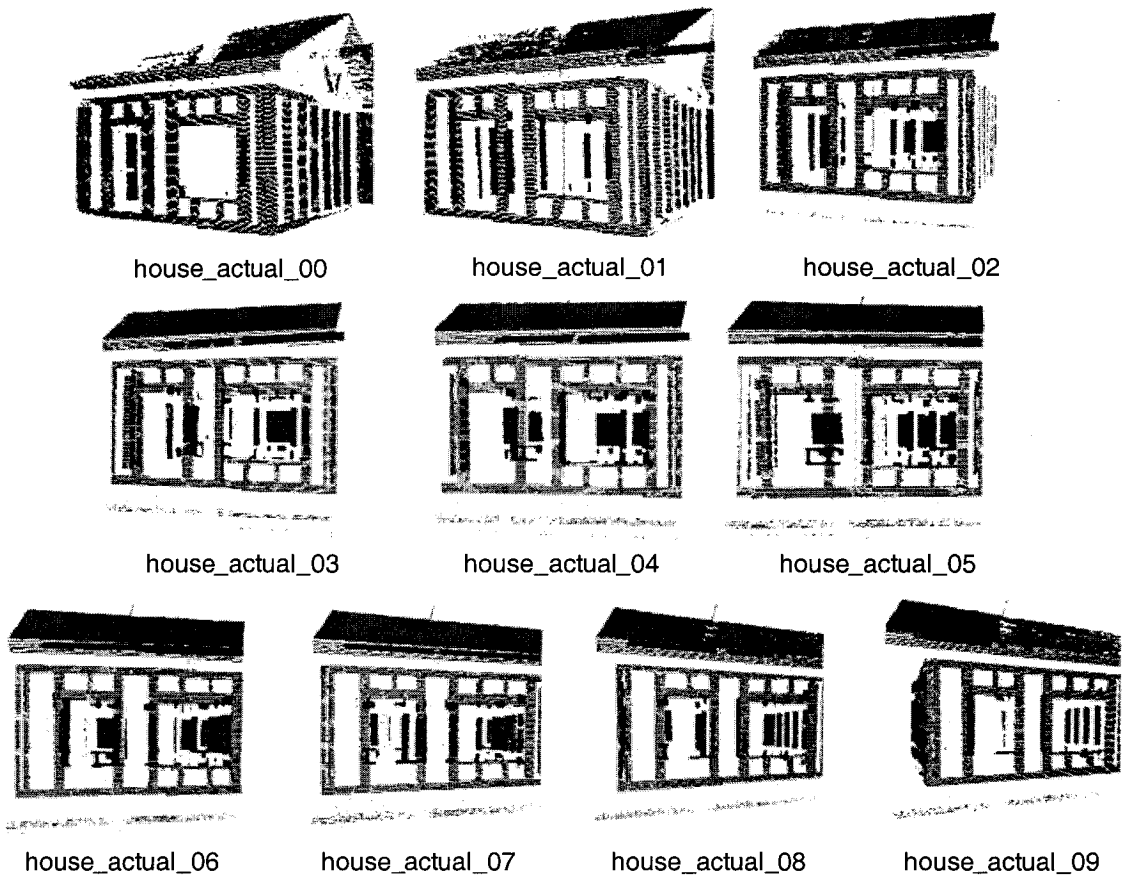

Fig. 37. Point Clouds of Actual House Range Image Perspectives

The frequency plots of rotation error, for the various pulses, at maximal dimensional resolutions of 16, 32, 64 and 128, at 1600 matching points are displayed in Fig. 39. The intuitive result of increased resolution producing better results holds up in this case. The solution selection issue observed previously, has been magnified, probably due to the presence of the background, noise, and outliers causing artificial symmetries. The number of results in complete error (the 2^{nd} through 5^{th} error categories) is only slightly greater than the simulated case, and hence the estimation of the rotation parameters can be deemed to maintain accuracy between simulated data and real data.

The pulse shape does not seem to be as important with this dataset, probably since there are many more points present than in the simulated data set, although the Gaussian pulse seems to produce consistently poor results regardless of the resolution. Fig. 38 shows the front and top views resulting from the registration and subsequent merging of the house_actual_05 data set with the house_actual_06 data set. There is a slight error in rotation as is shown from the top view in Fig. 38, and there is a small translation error that is visible in the front view.

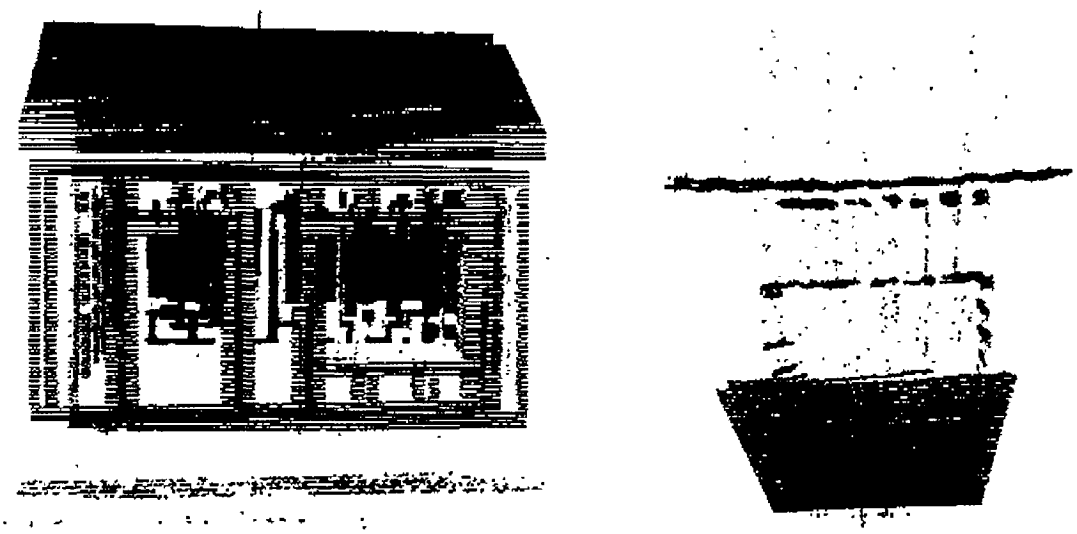

Fig. 38. Point Cloud of house_actual_05 (Red) Registered and Merged with house_actual_06 (Blue)

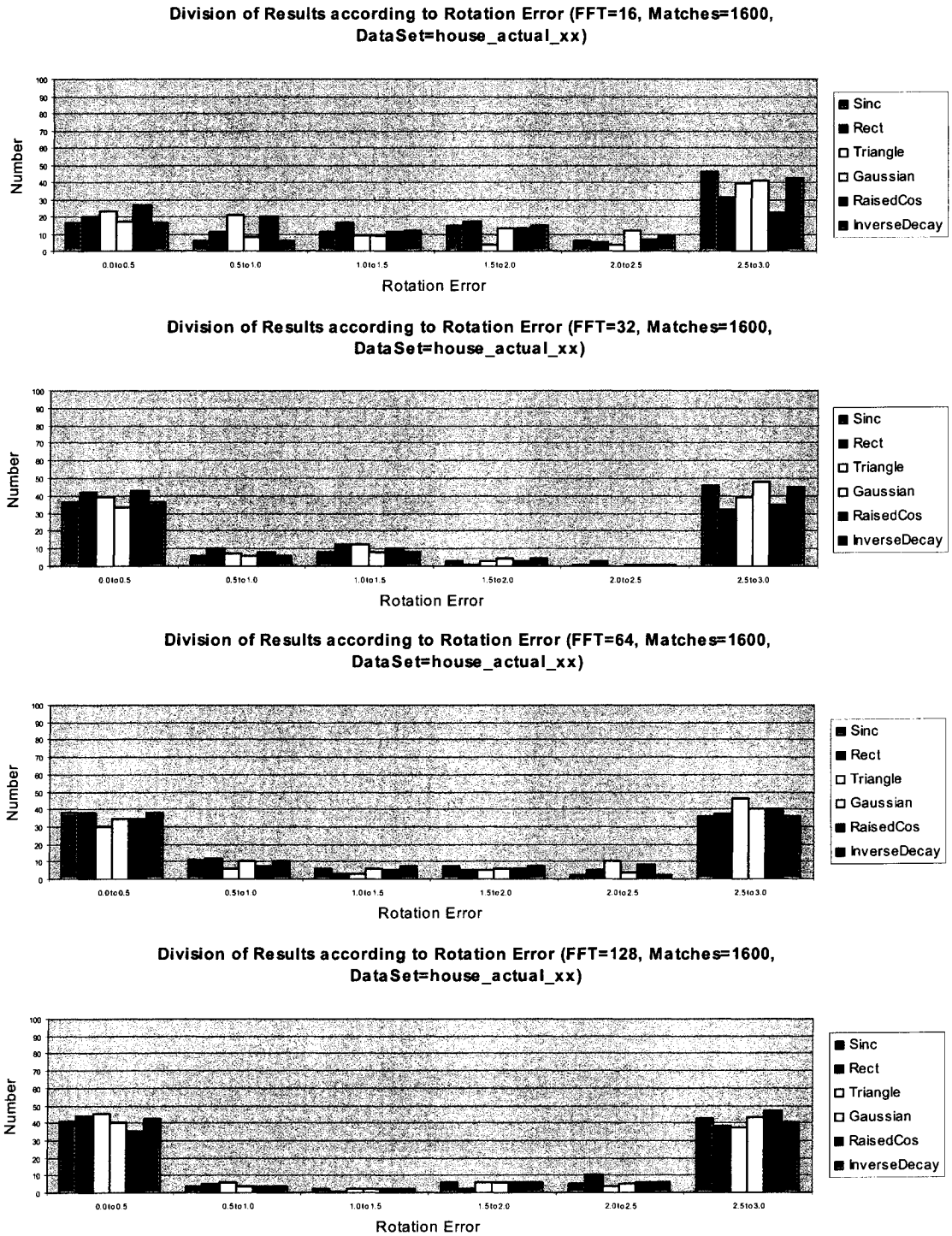

Fig. 39. Frequency Plots of Rotation Error for Actual House Data Sets (Number of Frequency Locations to Match = 1600)

The trend of decreasing rotational error when increasing the number of points to match is present, as is expected, in Fig. 40. The solution selection accuracy seems to have changed little with the difference of the number of points to match.

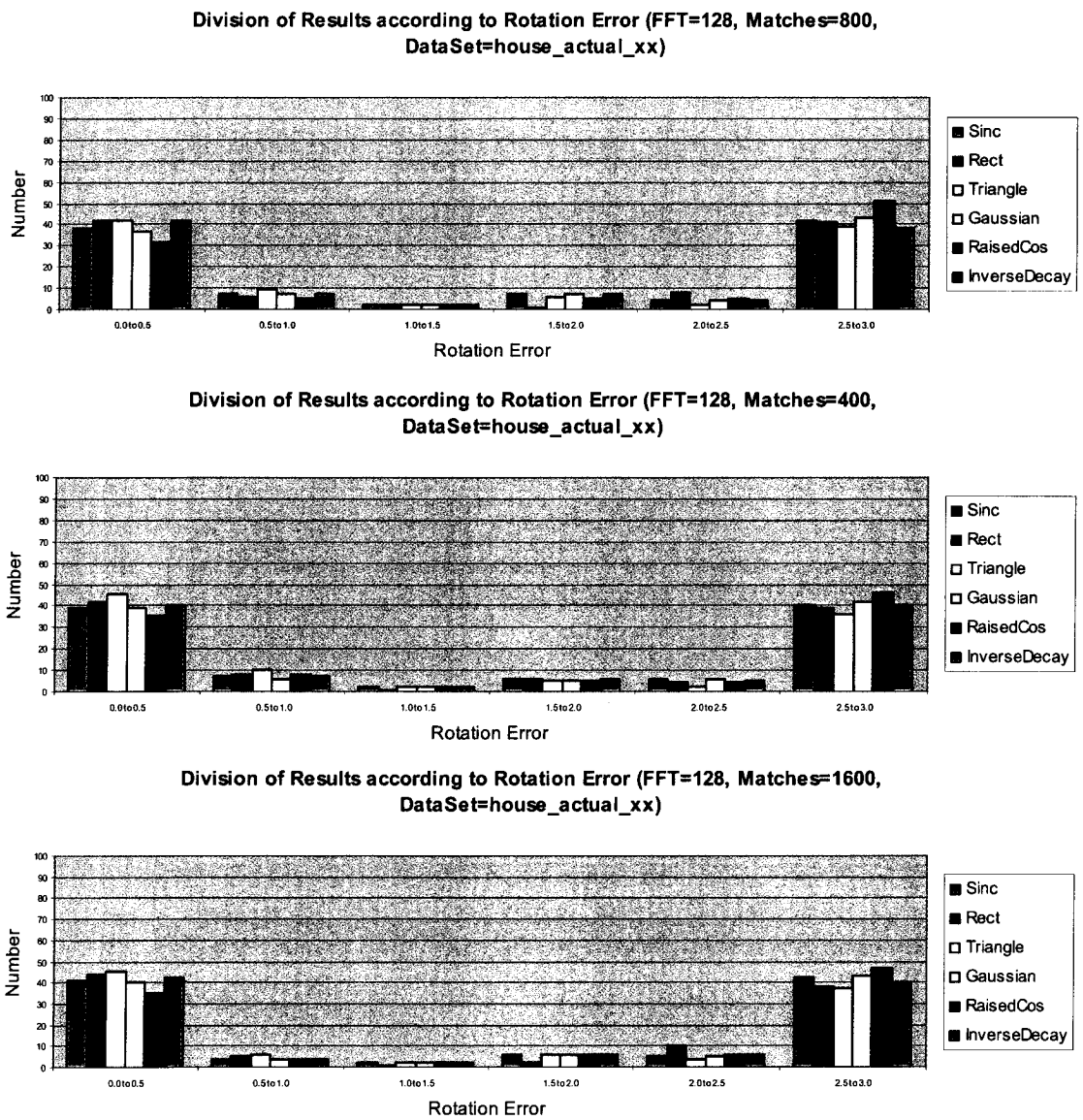

Fig. 40. Frequency Plot of Rotation Error for Actual House Data Sets (N=128)

The translation error increases with an increase in rotation error, with a similar trend as presented by Lucchese *et al.* in [1]. There is a noticeable difference with the translation error vs. rotation error plot in Fig. 41 when compared with the same plot for the simulated house in Fig. 30. The translation error for the actual house is more widely spread than that of the simulated house, possibly due to the background, and outliers.

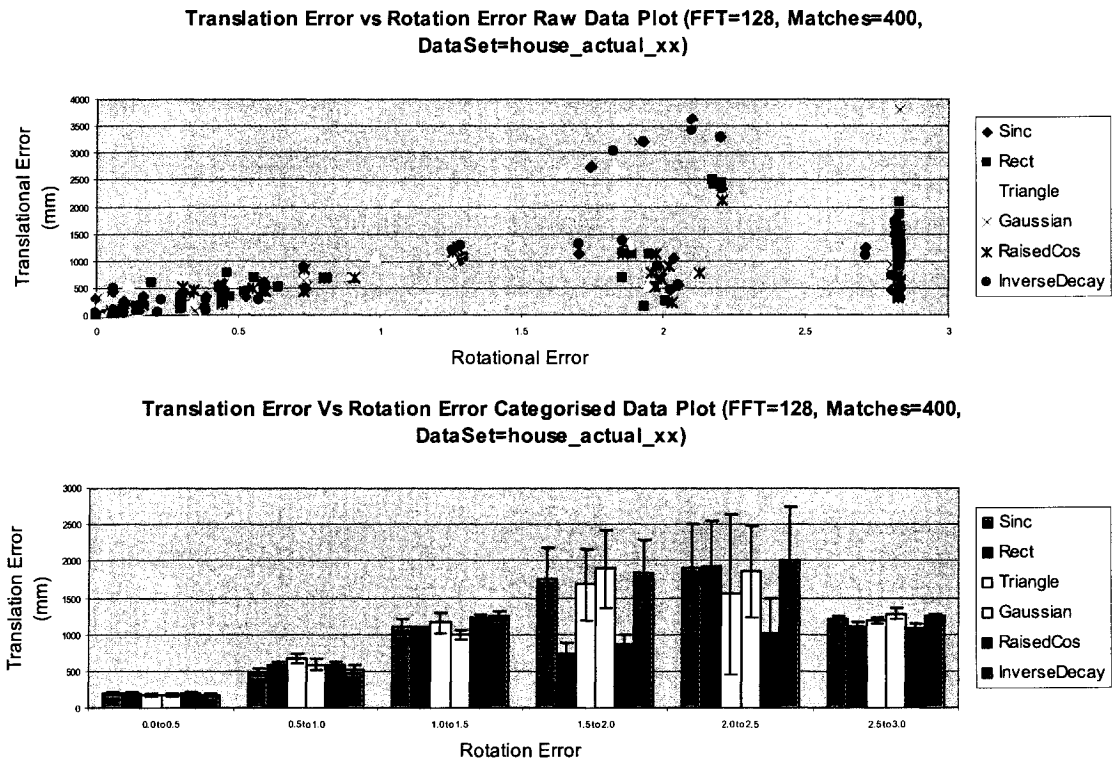

Fig. 41. Translation Error vs. Rotation Error for Actual House Data Sets

5.2.4. Overlap

The testing to determine the effectiveness of the proposed frequency domain algorithm with respect to the percentage of overlap was performed using two skewed planar objects. To obtain images with overlap, left and right images of the object were captured, with a percentage of overlap in the middle of the object belonging to both images. There are 41 different overlap percentage pairs, and as a result, only a small subset (16%, 48% and 90% overlap) is shown in Fig. 42.

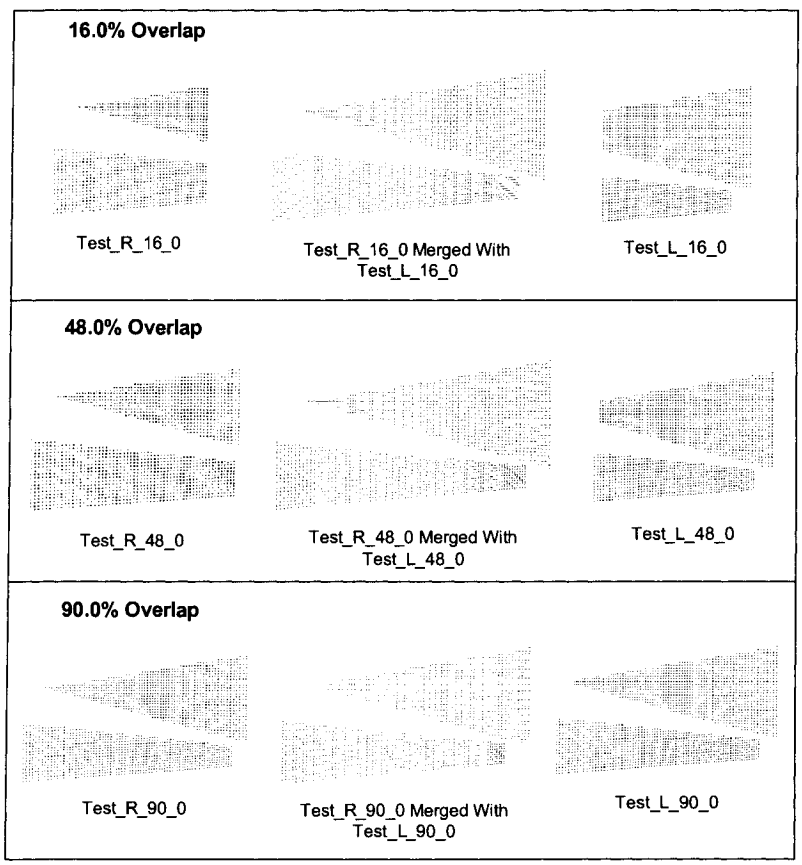

Fig. 42. Selection of Overlap Data Set

The determination of the overlap percentage, which the algorithm can handle, was determined by the point where the rotational error remains low. As shown in Table 3, the overlap percentage for which the algorithm performs acceptably differs according to the pulse used as indicated by the entries in bold. The bold and italicized entries indicate the lowest overlap percentage that the algorithm performs reasonably well at, while the bold and underlined entries mark the overlap percentage that the algorithm performs without error. The rectangular performs the poorest of the pulse shapes, with the minimal overlap percentage of 92.5%. The Gaussian pulse provides the best results, with the minimal overlap percentage of 13.5%. The remainder of the pulse shapes perform reasonably well with low minimal overlap percentage of 13.5% with the odd incorrect rotational parameters until between 55% and 85%.

Table 3. Overlap Determination For Each Pulse

Percentage Overlap	Sinc Rotation Error	Rect Rotation Error	Triangle Rotation Error	Gaussian Rotation Error	RaisedCos Rotation Error	InverseDecay Rotation Error
1.2	2.828424	2.820809	2.82814	2.828422	2.82762	2.828425
3.7	2.828424	0.001859	2.828277	2.828422	0.004339	2.828424
6.1	2.828424	2.827189	0.008058	2.828419	2.828427	2.828424
8.6	2.828425	2.827498	2.828425	2.828424	2.828427	2.828425
11.1	2.828425	0.11092	2.828425	2.828424	2.828427	2.828425
13.5	***0.001859***	2.826999	2.828419	***0.005578***	2.828422	***0.001859***
16	0.004339	2.828424	2.825845	0.003099	2.828425	0.004339
18.5	0.003099	2.828422	***0.004339***	0.003099	2.828412	0.003099
20.9	0.008058	1.992535	0.005578	0.006818	2.828419	0.006818
23.4	0.00062	0.001859	0.003099	0.008058	***0.001859***	2.828403
25.9	0.00062	2.828424	0.003099	0.00062	0.00062	0.00062
28.3	0.005578	1.98018	0.00062	0.004339	0.003099	0.008058
30.8	0.006818	1.989894	2.828419	0.005578	0.00062	0.006818
33.3	0.003099	2.828425	0.001859	0.001859	0.001859	0.003099
35.7	0.006818	2.010927	0.003099	0.005578	2.828424	0.006818
38.2	0.00062	2.828427	0.00062	0.00062	0.003099	0.001859
40.7	2.828422	0.003099	0.001859	0.004339	2.828412	2.828424
43.1	0.004339	2.82829	0.003099	0.005578	2.828424	2.828424
45.6	0.004339	2.828427	2.828427	0.005578	2.828427	0.004339
48	0.004339	0.00062	0.001859	0.005578	2.828427	0.004339
50.6	2.828427	0.016735	0.003099	0.004339	2.828425	0.001859
53	2.828416	2.828334	0.003099	0.001859	2.828427	2.828427
55.5	**0.004339**	0.00062	0.00062	0.004339	2.828403	2.828422
58	0.003099	2.828222	0.003099	0.001859	2.828427	**0.003099**
60.4	0.001859	2.828397	0.001859	0.001859	2.828427	0.001859
62.9	0.001859	2.827707	0.003099	0.001859	2.828422	0.001859
65.4	0.003099	2.828403	0.005578	0.001859	2.828425	0.001859
67.8	0.003099	2.828412	0.001859	0.003099	2.828425	0.001859
70.3	0.00062	1.978409	0.001859	0.001859	2.828427	0.00062
72.8	0.003099	1.960615	2.828422	0.003099	2.828422	0.003099
75.2	0.003099	1.965078	0.001859	0.001859	2.828424	0.003099
77.7	0.00062	2.034323	0.00062	0.00062	0.00062	0.00062
80.2	0.001859	0.00062	0.00062	0.00062	0.00062	0.001859
82.6	0.001859	2.009183	2.828412	0.00062	2.828412	0.001859
85.1	0.00062	0.003099	**0.00062**	0.00062	**0.00062**	0.00062
87.5	0.00062	2.828425	0.001859	0.00062	0.001859	0.00062
90	0.00062	2.828385	0.00062	0.00062	0.00062	0.00062
92.5	0.001859	***0.00062***	0.00062	0.001859	0.00062	0.001859
94.9	0.00062	0.001859	0.00062	0.00062	0.00062	0.00062
97.4	0.001859	0.00062	0.00062	0.00062	0.00062	0.001859
100	0.00062	0.00062	0.00062	0.00062	0.00062	0.00062

5.3. Analysis

The results presented in the previous section illustrate that the rotational parameter estimation is indeed accurate, but the solution selection stage causes the larger portion of incorrect registration results. Additionally, all the pulse shapes perform almost equivalently, with the rectangular pulse on average providing the poorest results, and the Gaussian pulse producing the best results, especially when the algorithm overlap performance is evaluated.

The registration, when the images have only a small amount of points (<10000), provides the best results under lower resolution condition as illustrated in the simulated data sets. If the number of points are large (>20000) then a larger resolution provides the best results, as shown in the case of the real life acquired data of the house model. When the number of frequency points used for matching in the rotation angle estimation is increased, there is less overall error in the estimation of the rotation parameters.

The algorithm as presented in this thesis performs equivalently to that discussed in Lucchese *et al.* in [1], (using the graphs of frequency of rotational error grouping, the graphs of translation error vs. rotation error, and the visual merged results of the registration process as the vectors of comparison), but with lower cost in the determination for the estimation of the rotational parameters. The translational parameters are determined using a method similar to that of Lucchese *et al.*, and as a result the overall efficiency (for both computational and memory resources) of the algorithm presented in this thesis is better than that of Lucchese *et al.* (as shown in

chapter 4), while maintaining an equivalent level of correctness in the registration estimation. Finally, as illustrated by the results shown in Table 1 and Table 2, the frequency domain registration algorithm is indeed more scalable and computationally efficient compared to the ICP algorithm, when average execution time is used as the measure.

Chapter 6. Conclusion

This chapter is divided into three sections. The first section summarizes the material presented within this thesis. The second section highlights the contributions resulting from the material contained in this work. The final section discusses possible future work.

6.1. Summary

The thesis commenced with a review of literature that contained the classical ICP approach by Besl and McKay [2], the frequency domain technique by Lucchese *et al.* [1], as well as many others. A large portion of the literature discussed various methods to improve the ICP algorithm in order to prevent the inherent drawbacks: the tendency of convergence to a local minima if the initial estimate is not close to actual registration, and execution time due to the slow matching process and iterative nature of the algorithm.

The frequency domain approach of Lucchese *et al.* in [1] requires no initial estimate, no other *a priori* knowledge of the images from the object that is being registered, and provides the separation of the estimation of the rotational parameters from the translational parameters. Unfortunately, the method requires much computation to and from the frequency domain, as well as many stages of statistical analysis and refinement. This provides the motivation of this thesis: to provide a more efficient method of utilising the promising frequency domain for registration.

The general mathematical theory behind frequency domain registration is presented in chapter 3. The theory starts with the mathematical justification of how the Fourier transform can segregate the rotational parameters from the translation parameters by using the magnitude and phase representations of the frequency domain. The rotational parameters can then be extracted by determining the zero line, corresponding to the axis of rotation, of the difference between two frequency domain magnitude images, and then determining the angle of rotation about the axis of rotation. The frequency domain registration method has an important ambiguity at this stage, due to the Hermitian symmetry problems (appendix 2), there exists two solutions, separated by 180°. A simple phase correlation of one image with the second image rotated by both solutions can disambiguate between these two solutions by determining which phase correlation provides the most impulsive function. The translational parameters are also determined by the location of the impulse of the phase correlation.

Chapter 4 discusses the algorithm for frequency domain registration, while pointing out the important differences between the implementation contained within this thesis and Lucchese *et al.* The algorithm also provides a framework for the general multiple dimension case for datasets of multiple data types (range, colour, etc.). The primary difference between Lucchese *et al.* and the method presented in this thesis lies within the estimation of the rotational parameters. A different and more efficient difference function is used to determine the magnitude difference between the two frequency domain images as well as a moving window minimal weight path finding algorithm to determine the axis of rotation. The angle of rotation is determined by an efficient

iterative coarse to fine least squares algorithm as compared to the statistical correlation approach used in Lucchese *et al.* The principle of using the phase correlation to determine the correct rotational solution, as well as estimating the translation parameters is the same as Lucchese *et al.*

The results of the execution of the described algorithm are presented in chapter 5, as well as the experimental setup developed primarily for collecting data for this thesis. The results show that the algorithm presented in this thesis is more computationally efficient than that presented by Besl and Mckay in [2], and provides similar error curves to that of Lucchese *et al.* in [1]. Some samples of the registered data were merged, and were presented in order to illustrate the quality of the registration estimates. The evaluation of several different pulses was performed, with the rectangular pulse providing poor results, and the other pulses (Gaussian, triangular, inverse decay, raised cosine, and sinc) producing similar good results. The Gaussian shaped pulse provided the best results in terms of the percentage of overlap required for a correct registration to proceed, while the rectangular shaped pulse proved to be the worse in most situations. The algorithm was shown to be able to successfully provide the correct estimation of the rotational parameters with as little as 13.5% overlap, although it is typically more successful with higher degree of overlap.

6.2. Contributions

This thesis proposed an automatic registration estimation algorithm that is non-iterative, does not require human intervention, does not require an initial estimation, and

is independent of the geometry of the object. The frequency domain registration algorithm is presented in a generalized framework, allowing for the multidimensional, multiple data type (such as position and colour) case. This is fully detailed in chapter 4.

The frequency domain registration algorithm proposed in this thesis illustrated scalability, through the varying of the maximal dimensional size (16, 32, 64, and 128) and varying the number of frequency locations used in determining the angle of rotation (400, 800, and 1600). The proposed frequency domain registration algorithm provides a noticeably slower growth of execution time with respect to the number of points (compare Table 1 and Table 2) as compared to the ICP algorithm, which means that the frequency domain registration algorithm presented in this thesis is more scalable than the ICP algorithm.

Chapter 4 demonstrates that the algorithm proposed in this thesis is more efficient at calculating the rotational parameters than the algorithm presented by Lucchese *et al.* in [1]. This fact is demonstrated by showing that the difference function developed in this thesis (eq.(18)) to determine the magnitude difference of the frequency domain elements is more computationally efficient than that proposed by Lucchese *et al.* in [1]. Also, the proposed method for the determination of the axis of rotation using the moving window method (section 4.2.4) is more memory and computationally efficient than that proposed by Lucchese *et al.* in [1] due to the calculation of fewer values (but exactly equivalent if the window size is large). Finally, by performing fewer Fourier transforms than Lucchese *et al.* [1] and utilising an efficient determination of the angle of rotation through the use of a coarse to fine least squares method reduces the computational workload.

An integrated range data collection system was created (section 5.1) to provide real world data to validate the algorithm. This led to several publications ([39], [40], and [41]) that describe the evolution and quality of the system and its capability to collect range data.

6.3. Future Work

The algorithm presented in this thesis, although successfully laying the foundations for a multiple dimensional frequency domain multiple typed data registration framework, has yet to be fully tested and developed. In order for this framework to be complete, the axis-angle method needs to be extended to multiple dimensions, and tested with multiple dimensional data. Additionally, during experimentation (chapter 5), the performance of the solution selection method needs to be improved. This could be achieved through a different process, or with different evaluation criteria. Finally, another transformation domain that provides a better separation in regards to rotation and translation needs to be investigated, which dissolves the rotation parameters ambiguity problem, while maintaining the useful properties found in the Fourier transform.

References

[1] L. Lucchese, G. Doretto, G.M. Cortelazzo, "A Frequency Domain Technique for Range Data Registration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 11, pp. 1468-1484, Nov. 2002.

[2] P.J. Besl, N.D. McKay, "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 239-256, Feb. 1992.

[3] M.A. Rodrigues, Y. Liu, "Registering Two Overlapping Range Images Using A Relative Registration Error Histogram", *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3, pp. 841-844, June 24-28, 2002.

[4] T. Masuda, N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images", *Proceedings of the IEEE Second CAD Based Vision Workshop*, pp. 106-113, Feb. 1994.

[5] D. Huber, O. Carmichael, M. Hebert, "3-D Map Reconstruction from Range Data", *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 1, pp. 891-897, Apr. 24-28, 2000.

[6] Y. Liu, M.A. Rodrigues, "A Geometric Histogram Method for Accurate and Robust Motion Estimation from Range Data", *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Vol. 4, pp. 2269-2274, Oct. 7-10, 2001.

[7] R. Benjemaa, F. Schmitt, "Fast Global registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique", *Proceedings of the IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 113-120, May 12-15, 1997.

[8] G.C. Sharp, S.W. Lee, D.K. Wehe, "ICP Registration Using Invariant Features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, January 2002.

[9] D.A. Simon, M. Hebert, T. Kanade, "Real-time 3-D Pose Estimation Using a High-Speed Range Sensor", *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 3, pp. 2235-2241, May 8-13, 1994.

[10] H-Y. Lin, M. Subbarao, "A Vision System for Fast 3D Model Reconstruction", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. II-663 – II-668, Dec. 8-14, 2001.

[11] C.-S. Chen, Y.-P. Hung, J.-B. Cheng, "A Fast Automatic Method for Registration of Partially-Overlapping Range Images", *Proceedings of the IEEE Sixth International Conference on Computer Vision*, pp. 242-248, Jan. 4-7, 1998.

[12] E. Gagnon, J-F. Rivest, M. Greenspan, N. Burtnyk, "A Computer-Assisted Range Image Registration System for Nuclear Waste Cleanup", *IEEE Transactions on Instrumentation and Measurement,* Vol. 48, No.3, pp. 758-762, June 1999.

[13] D. Tubić, P. Hébert, D. Laurendeau, "3D Surface Modeling from Range Curves", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. I-842 - I-849, June 18-20, 2003.

[14] C.M. Cyr, A.F. Kamal, T.B. Sebastian, B.B. Kimia, "2D-3D Registration Based on Shape Matching", *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 198-203, June 11-12, 2000.

[15] L. Lucchese, G.M. Cortelazzo, "A Noise-Robust Frequency Domain Technique for Estimating Planar Roto-Translations", *IEEE Transactions on Signal Processing*, Vol. 48, No. 6, pp. 1769-1786, June 2000.

[16] A.P. Ashbrook, R.B. Fisher, C. Robertson, N. Werghi, "Segmentation of Range Data into Rigid Subsets Using Surface Patches", *Proceedings of IEEE Sixth International Conference on Computer Vision*, pp. 201-206, Jan. 4-7, 1998.

[17] P. Payeur, C. Chen, "Registration of Range Measurements with Compact Surface Representation", *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No.5. pp. 1627-1634, Oct. 2003.

[18] K. Higuchi, H. Delingette, M. Hebert, K. Ikeuchi, "Merging Multiple Views Using a Spherical Representation", *Proceedings of the IEEE Second CAD-Based Vision Workshop*, pp. 124-131, Feb. 8-11, 1994.

[19] C.S. Yang, P. Payeur, "Calibration of a Multi-Modal 3D Scanner", *Proceedings of the IEEE International Instrumentation and Measurement Conference*, pp. 865-870, Ottawa, ON, May 2005.

[20] CRS Robotics Corp., *F3 Robot System User Guide*, Burlington, ON, 2000.

[21] CRS Robotics Corp., *Track User Guide,* Burlington, ON, 1999.

[22] CRS Robotics Corp., *C500C Controller User Guide*, Burlington, ON, 2000.

[23] CRS Robotics Corp., *RAPL-3 Language Reference Guide,* Burlington, ON, 1999.

[24] Servo-Robot Inc, *Jupiter 3-D Laser Vision Camera Installation and Operation Manual,* St-Bruno, QC, 1996.

[25] M. Rioux, "Laser Rangefinder Based on Synchronized Scanning", in *Applied Optics*, vol. 23, pp. 3837-3844, 1985.

[26] Servo-Robot Inc, *Cami-Box Vision System Control Unit User's Manual,* St-Bruno, QC, 2000.

[27] 3D Scanners, *ModelMaker range sensors technology,* http://www.3dscanners.com, UK, 2004.

[28] Inspeck, *3D Capturator digitizer*, http://www.inspeck.com, Québec, QC, 2004.

[29] 3dcgi.com, *3D cameras and digitizers*, http://www.3dcgi.com/cooltech/cameras/cameras.htm, 2004.

[30] VRex Inc., *CAM-3000C – Stereoscopic 3-D Video Camera*, Elmsford, NY, 1994.

[31] W.M. Keck, Virtual Factory Lab, *Virtual Robot System version 2,* Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, pp. 25-39, 2002.

[32] Servo-Robot Inc, *Data-3000 System Commands and Parameters,* St-Bruno, QC, 1997.

[33] Matrox Electronic Systems Ltd., *Matrox Image Library Version 6.1 User Guide*, Dorval, QC, March 1, 2000.

[34] Matrox Electronic Systems Ltd, *MIL/MIL Lite version 6.1 Board-Specific Notes*, Dorval, QC, March 1, 2000.

[35] Intel Corporation, *Open Source Computer Vision Library – Reference Manual*, Santa Clara, CA, 2001.

[36] R. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses", *Transactions on IEEE Journal of Robotics and Automation*, Vol. 3, Issue 4, pp. 323-344, August 1987.

[37] S. Birchfield, C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo", *Proceedings of the IEEE Sixth International Conference on Computer Vision*, pp. 1073-1080, 4-7 January 1998.

[38] C. Chen, *Registration of Range Measurements With Compact Surface Representation,* Master thesis, University of Ottawa, Ottawa, ON, 2002.

[39] P. Curtis, P. Payeur, "An Integrated Robotic Laser Range Sensing System for Automatic Mapping of Wide Workspaces", *Proceedings of the IEEE Canadian Conference on Electrical And Computer Engineering*, Vol. 2, pp. 1135-1138, May 2004.

[40] P. Curtis, C.S. Yang, P. Payeur, "An Integrated Robotic Multi-Modal Sensing System", *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference*, pp. 1991-1996, Ottawa, ON, 2005.

[41] C.S. Yang, P. Curtis, P. Payeur, "Calibration of an Integrated Robotic Multi-Modal Range Scanner", submitted to *IEEE Transactions on Instrumentation and Measurement*, 2005.

[42] H. Anton, *Elementary Linear Algebra,* 6th Ed., Toronto: John Wiley & Sons, Inc., 1991.

[43] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Ed., Cambridge: Cambridge University Press, 1992.

[44] A. Aquilera, R. Pérez-Aguila, "General n-Dimensional Rotations", *Proceedings of the WSCG International Conferences in Central Europe on Computer Graphics, Visualization, and Computer Vision,* pp. 1-8, Feb. 2-6, 2004.

[45] J.G. Proakis, D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd Ed., Toronto: Prentice-Hall Canada Inc., 1996.

[46] A.V. Oppenheim, A.S. Willsky, S.H. Nawab, *Signals & Systems*, 2nd Ed., Toronto: Prentice Hall Canada Inc., 1997.

Appendix 1. Rigid Linear Transformations

N-dimensional rigid linear transformations involve only a translation and a rotation of a set of non-deformable objects about a fixed reference frame. Translations are typically represented by a vector describing the offset along each of the coordinate axes of which the new reference frame will be defined. Rotations can be represented in many different ways, such as matrices, axis-angle, and quaternion to name a few. The matrix representation is a convenient form as it represents the rotation as a linear system, and rotation matrices are special matrices having mathematical properties, which are easily exploited. The key property involves orthonormality between rows, and between columns. An orthonormal matrix can be inverted by taking the conjugate transpose, also called the Hermitian transpose, of itself [42]. An orthonormal matrix multiplied by another orthonormal matrix results in a new orthonormal matrix, meaning that an arbitrary rotation matrix can be formed by multiplying several elementary rotation matrices together. All of the representations of rotation are equivalent, and can be converted from and to each other.

An N-dimensional rotation is represented by an NxN rotation matrix, and an N-dimensional translation is represented by an N-dimensional translation vector. Mathematically, the transform is as follows $O' = R \cdot O + T$, where O is the object in N-D space, T is the translation vector, R is the rotation matrix, and O' is the transformed object. A more convenient mathematical representation, which merges the rotation matrix and translation vector together, is the homogenous transformation matrix. A

homogenous transformation matrix is formed as follows $Q = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$, where R is the NxN rotation matrix, T is a Nx1 translation column vector, and 0 is a 1xN null row vector. Objects in homogenous space must have an additional one appended in the additional row and column. Homogenous transformations have the form $O' = Q \cdot O$, and are equivalent to the non-homogenous case. Due to this equivalence, and the fact that translations are easy to represent and apply, only rotations will be considered in the discussion below.

A1.1. 2-D Rotations

2-D rotations are the simplest rotation case, as rotations occur about the origin, and as such there is only one parameter, the angle of rotation. The 2-D rotation matrix is defined as:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{49}$$

The 2-D rotation matrix in eq.(49) specifies a counter-clockwise rotation of an angle θ about the origin, or in other words the angular shift of the x-axis towards the y-axis.

A1.2. 3-D Rotations

3-D space contains three axes, each of which a rotation can be centred about, and as such there are three separate elementary rotation matrices which can be combined to define an overall rotation about an arbitrary axis.

The elementary rotations are as follows:

$$R_x(\theta_x)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\theta_x) & -\sin(\theta_x)\\ 0 & \sin(\theta_x) & \cos(\theta_x)\end{bmatrix} \tag{50}$$

$$R_y(\theta_y)=\begin{bmatrix}\cos(\theta_y) & 0 & \sin(\theta_y)\\ 0 & 1 & 0\\ -\sin(\theta_y) & 0 & \cos(\theta_y)\end{bmatrix} \tag{51}$$

$$R_z(\theta_z)=\begin{bmatrix}\cos(\theta_z) & -\sin(\theta_z) & 0\\ \sin(\theta_z) & \cos(\theta_z) & 0\\ 0 & 0 & 1\end{bmatrix} \tag{52}$$

R_x describes a counter-clockwise rotation about the x-axis (y-axis towards the z-axis), by an angle of θ_x, R_y describes a counter-clockwise rotation about the y-axis (z-axis towards the x-axis), by an angle of θ_y, and R_z describes a counter-clockwise rotation about the z-axis (x-axis towards the y-axis), by an angle of θ_z.

A1.2.1. Axis-Angle to Matrix Conversion

In 3-D applications the axis-angle format is also useful. Axis-angle format defines an angle of rotation about an axis of rotation. In the physical world, all objects, which rotate, have an axis of rotation, and are easily exemplified by observing that the earth (as well as all celestial bodies) rotate about an axis. The axis vector in 3-D is defined as:

$$\vec{R}_{Axis} = [x \quad y \quad z]^T \tag{53}$$

where x, y, and z are the Cartesian co-ordinates representing the direction of the axis of rotation.

The conversion between axis-angle and matrix format is relatively easy through the use of similarity transforms. Similarity transforms have the property of not changing the eigenvalues of a system, and take the form:

$$X' = S \cdot X \cdot S^{-1} \tag{54}$$

In the case of the axis-angle format, the similarity transformation matrix, which is being used, is constructed via the normalized axis vector. The similarity transformation matrix that is used in this context will be given the notation $S\left(\vec{R}_{Axis}\right)$ to emphasize the fact that it is constructed with respect to the axis of rotation. To create the similarity transformation matrix, the Gram-Schmidt [43] or Householder [43] transformation methods are used,

followed then by ensuring that the resulting matrix is positive definite. This in practice is defined as follows:

1) Create a 3x4 matrix in the form A=[$\vec{R}'_{Axis}$ |I], where $\vec{R}'_{Axis}$ is the normalized column vector form of the rotation axis ($\vec{R}'_{Axis} = \frac{\vec{R}_{Axis}}{\left|\vec{R}_{Axis}\right|}$), and I is the 3x3 identity matrix.

2) Apply QR-decomposition [43] to the matrix A, which performs the Gram-Schmidt, or Householder transformation algorithms, to form the Q and R Matrices (note Q*R=A).

3) The Q Matrix will contain in its first column, $\vec{R}'_{Axis}$. To ensure that the similarity matrix is positive definite, multiply the third column of Q, by its determinant. The relationship between S and Q is:

$$S\left(\vec{R}_{Axis}\right)=[Q_{Column\ 1}|Q_{Column\ 2}|\det(Q)Q_{Column\ 3}] \tag{55}$$

Alternatively, the similarity transformation matrix, S, can be obtained via rotating an appropriately sized identity matrix (3x3 for the 3-dimensional case), until the first column becomes the normalize rotation axis vector.

The matrix, which we will be performing the similarity transformation on, X in eq.(54), will be simply the elementary rotation matrix for a rotation about the elementary

x-axis (this is due to the fact that the first column of the similarity transform corresponds to the axis of rotation, the other two columns correspond to the plane of rotation, which is where the rotation is to occur):

$$R(Angle)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(Angle) & -\sin(Angle)\\ 0 & \sin(Angle) & \cos(Angle)\end{bmatrix} \tag{56}$$

Since the similarity matrix is a real orthonormal matrix, and has the properties described at the beginning of this section, the rotation matrix representing the axis-angle format is:

$$R\left(\vec{R}'_{Axis}, Angle\right)= S\left(\vec{R}'_{Axis}\right)\cdot R(Angle)\cdot S^{-1}\left(\vec{R}'_{Axis}\right)$$

$$R\left(\vec{R}'_{Axis}, Angle\right)= S\left(\vec{R}'_{Axis}\right)\cdot R(Angle)\cdot S^{T}\left(\vec{R}'_{Axis}\right) \tag{57}$$

A1.2.2. Matrix to Axis-Angle Conversion

In the previous section, the conversion from Axis-Angle to matrix form was shown. In some situations, it is also useful to be able to convert from matrix form to Axis-Angle form. The conversion of a rotation from matrix form to its equivalent axis-angle form consists of a series of simple operations:

1) Determine the eigensystem of the Matrix: R*V=V*D, where V contains the eigenvectors of R arranged column-wise, and D contains the eigenvalues of R

arranged along the diagonal (this can be determined, as presented in [43], by reducing the rotation matrix to upper Hessenberg form, and then applying the Hessenberg QR algorithm with shifts to determine D, and then collecting the respective transforms to produce V).

2) The axis of rotation corresponds to the eigenvector whose eigenvalue is unique and real.

3) Form the similarity transformation matrix as described in section A1.2.1.

4) Determine the matrix representing the angle of rotation by performing the inverse similarity transform: $R(Angle)=S^T(\vec{R}'_{Axis})R(\vec{R}'_{Axis},Angle)S(\vec{R}'_{Axis})$

5) Find the angle of rotation by: $Angle=\text{atan}(R(Angle)_{3,2}/\ R(Angle)_{2,2})$, where the subscript represents the row, column indices of the rotation matrix.

A1.3. N-D Rotations

For the sake of completeness, N-D rotations are also examined. The general rotation matrix defines the angular motion of one axis towards another, and as such, the rotation occurs only on a specific 2-D plane in N-D space [44]. The generalized rotation matrix, which is defined by an angle (θ), a dimensional size (N), and an axis (i) that is rotated towards another axis (j) is:

$$R_{i,j}^{N}(\theta)=\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,1} & \cdots & r_{2,N} \\ \cdots & \cdots & \cdots & \cdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} \end{bmatrix} \quad , r_{m,n}=\begin{cases} 1 & m=n\neq i,j \\ 0 & m\neq n, m, n\neq i, j \\ \cos(\theta) & m=n=i,j \\ \sin(\theta) & m=j, n=i \\ -\sin(\theta) & m=i, n=j \end{cases} \tag{58}$$

Appendix 2. Hermitian Symmetry

Hermitian symmetry is an important property when dealing with Fourier transforms of real data, and hence it is a property that needs to be understood when dealing with frequency domain registration theory as presented within this thesis. The basic principle behind Hermitian symmetry is as follows:

$$F(\vec{w}) = F^*(-\vec{w}) \tag{59}$$

where F is the complex frequency function of the frequency vector $\vec{w}$. If eq. (59) is separated into magnitude and phase components, then the ambiguity caused by the Hermitian symmetry is put into evidence. Eq. (60) shows the symmetric nature of the magnitude due to the Hermitian symmetrical property of the Fourier transform, and hence the 180° ambiguity in the estimation of the rotation angle.

$$|F(\vec{w})| = |F(-\vec{w})| \tag{60}$$

$$\angle F(\vec{w}) = -\angle F(-\vec{w}) \tag{61}$$

Appendix 3. Fourier Transform

The Fourier transform is a powerful tool and is widely used in signal processing. A greater understanding of the various input and output relationships that exist within any particular system can be obtained through the analysis of the various interacting signals represented in the frequency domain, using only simple mathematical representations, as well as identifying particularities in the signal which may not be easily examined using other time or space domain techniques.

For example, it is easy to analyze the input/output relationship of the effects of a signal traversing a transmission medium using frequency domain techniques. We can also analyze speech of a person talking, and through the location of frequency spikes, determine with high probability the identity of a person (voice identification). The particular application to which the Fourier transform is applied in this thesis is the estimation of the registration parameters.

The classical 1-D Fourier transform is defined by the following relationship in which f(x) is the value of the signal at instant x, F(w) is the complex value of the Fourier transform at frequency w. The 1-D Fourier Transform is defined as follows [45][46]:

$$F(w) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi wx} dx \tag{62}$$

The inverse 1-D Fourier transform is defined as follows [45][46]:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w) e^{j2\pi wx} dw \tag{63}$$

These equations are sufficient when dealing with continuous signals, but most modern day systems and signals are digital in nature. With this in mind, the discrete Fourier transform, or DFT, has been developed, which enables the use of this powerful tool in digital systems. The 1-D DFT is defined as follows [45]:

$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-j\frac{2\pi kn}{N}} \tag{64}$$

where f[n] is the value of the signal at the discrete location n, N is the number of samples in the signal to be transformed, F[k] is the complex value of the DFT at the discrete frequency location k. The 1-D inverse DFT (IDFT) is [45]:

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} F[k] e^{j\frac{2\pi kn}{N}} \tag{65}$$

Both the Fourier transform and the DFT can be extended to cover multiple dimensions, so that multi-dimensional signals can be processed. The N-dimensional DFT is:

$$F[\vec{k}] = \sum_{n_N=0}^{M_N-1} \sum_{n_{N-1}=0}^{M_{N-1}-1} \cdots \sum_{n_1=0}^{M_1-1} f[\vec{n}] e^{-j2\pi \vec{k}^T M \vec{n}} \quad , \vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_N \end{bmatrix}, \vec{k} = \begin{bmatrix} k_1 \\ k_2 \\ \cdots \\ k_N \end{bmatrix}, M = \begin{bmatrix} \frac{1}{M_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{M_2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \frac{1}{M_N} \end{bmatrix} \tag{66}$$

where n_i is the discrete space or time domain location along i^{th} dimension, $f[n_1,n_2,\ldots,n_N]$ is the value of the signal at $[n_1,n_2,\ldots,n_N]$, k_i is the index along i^{th} frequency dimension, $F[k_1,k_2,\ldots,k_N]$ is the frequency value of the signal at $[k_1,k_2,\ldots,k_N]$, and M_i is the number of samples along each dimension in space or time.

Similarly, the N-dimensional IDFT is:

$$f[\vec{n}] = \frac{1}{\prod_{d=1}^{N} M_d} \sum_{k_N=0}^{M_N-1} \sum_{k_{N-1}=0}^{M_{N-1}-1} \cdots \sum_{k_1=0}^{M_1-1} F[\vec{k}] e^{j2\pi \vec{k}^T M \vec{n}} \quad , \vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_N \end{bmatrix}, \vec{k} = \begin{bmatrix} k_1 \\ k_2 \\ \cdots \\ k_N \end{bmatrix}, M = \begin{bmatrix} \frac{1}{M_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{M_2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \frac{1}{M_N} \end{bmatrix} \tag{67}$$

where M_i is the number of samples along each dimension in the frequency domain.