Université d'Ottawa · University of Ottawa

Registration of Range Measurements With Compact Surface Representation

by

Changzhong Chen

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master in Applied Science

Ottawa-Carleton Institute of Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

October 2002

0-612-76568-7

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Changzhong Chen

I further authorize the University of Ottawa to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Changzhong Chen

To my dear mother, father and two brothers

To my dear wife

Contents

List of Figures

List of Tables

Abstract

This thesis introduces an automatic approach for registration estimation between successive viewpoints of a laser range camera that takes advantage of the raw measurements and does not require any external device for pose estimation nor complex feature extraction or triangulation. Assuming only object rigidity and some overlap between the scanned areas, the approach allows one to estimate the six rotation and translation parameters that link scans gathered from different viewpoints in 3D space. A compact modified Gauss sphere representation is used to encode a simple planar patch approximation of the objects' surface and to validate the mapping between the measurements as the appropriate rotation and translation parameters are computed. This solution results in an important reduction of the computational workload and a sufficient accuracy for most robot navigation applications. The proposed approach is demonstrated in an experimental context using real range measurements collected from a series of viewpoints.

Acknowledgments

I would like to express my great appreciation to my thesis supervisor Dr. Pierre Payeur for his guidance, research facilities and financial support during my Master's program. I would also like to thank Dr. Robert Laganière from the VIVA research Laboratory at the University of Ottawa and Dr. Victor C. Aitken from Carleton University for their valuable comments.

My special thanks are due to all members of the VIVA research laboratory, Dr. Martin Bouchard and Ms. Lucette Lepage at the School of Information Technology and Engineering, and the University of Ottawa staff in general.

I am truly grateful to my parents and brothers for their consistent support from the beginning of my studies without which this work would not have been possible for me. Millions of thanks and appreciations are also given to my wife and my parents-in-law as they let me have a nice mood everyday in doing my studies.

Finally, I also show my eternal thanks to my friends for their kind help before coming to Canada, particularly, XiaoJie CHEN, ZhiWei CHEN, Zhong CHEN, Dan DAN, Ying GUO, Feng JIANG, Bing LIN, ZhiYong LIU, WeiXian MAO, Wei REN, YongQiang RUAN, Yang SONG, RuiFeng SUN, YongJie WU, YanDan YE, YuWen YE and XianYong YU etc.

Chapter 1 Introduction

1.1 Introduction

With the development of various technologies, and an increasing human's dependence on machines, artificial intelligence (AI) is becoming a hot research topic worldwide. Like the human body, artificial intelligence is very complex. AI requires knowledge from many disciplines, such as electrical engineering, computer science, chemistry and biology. Of all the branches of artificial intelligence, one of the most diverse is that of computer vision systems -- also called machine vision systems -- which tries to reproduce the human vision capabilities. However, the range of applications for this relatively new and still growing technology is as diverse as the principles and techniques that are used. These applications go from basic tasks like part inspection to exciting new technologies such as autonomous robot navigation systems. In these applications, the computer vision system captures images of its objective environment, interprets and renders them. These tasks seem easy for human's eyes to perform, but not that simple for a computer vision system.

A major goal of computer vision is to produce systems that enable successful interaction with the environment (rather than aiming only at producing a particular representation). Interaction may mean [1]:

a.) Navigating a robot or autonomous vehicle through obstacles or along a road;

b.) Moving a robot arm to manipulate parts for assembly;

c.) Recognizing human gestures and movements for computer control;

d.) Identifying images in a database on the basis of their content.

In order to achieve successful applications of these interactions, online building of accurate 3D models for the environment/objects is very important. For example, an autonomous robot navigation would fail if the system cannot obtain a good 3D model estimate of its surroundings that will lead to correct path planning. Large delays in building the 3D model would also make the navigation planning invalid. An incorrect 3D model also deviates the interpretation of human gestures and movements for computer control. As a result, building accurate 3D representations of the environment is a critical issue in the development of a successful computer vision system implying interaction.

Unfortunately, automatic 3D modeling still remains a challenge. A great deal of research work is still required to develop robust and effective techniques. Typically, automatic construction of 3D object models involves the following three steps:

a.) Data acquisition from multiple viewpoints;

b.) Registration of views;

c.) Integration of the data sets.

Data acquisition in computer vision involves obtaining either intensity images or range data of the object surfaces. Usually, we use regular cameras to get object texture (intensity images), and ultrasonic sensors or laser sensors for depth (range images). However, the construction of complete object models requires that data are obtained from multiple viewpoints because of the limited field of view of sensors and the complexity of the scenes that preempt the system from getting measurements on the complete surface of solid objects from a single viewpoint.

For this reason, the integration of these multiple views is unavoidable. The way to integrate them depends on the representation chosen for the model. However, knowledge of the transformation relating data obtained from different viewpoints is minimally required. Elfes [2] and Payeur [3] have developed some algorithms to deal with the integration of range data from different viewpoints. Other related research can also be found in [4, 5, 6].

The registration step consists of finding the transformations that relate the views from different viewpoints and that are required in the integration step. Each dataset gathered from a given point of view is defined with respect to a local sensor-based reference frame. The position and orientation of the sensors at each point of view must be precisely estimated to ensure that the information obtained from every source is merged in a consistent way to build a 3D model. Contrarily, a bad registration can preempt a correct integration of data. It ultimately results in holes and discontinuities, or even completely wrong 3D models, which are not reliable for collision avoidance or fine interaction control between a robot and its environment [4]. In short, accurate registration estimation is a vital step in 3D modeling. Developing some robust techniques to estimate registration parameters becomes the rule of thumb. In this thesis, we focus on the issue of pair-wise registration of range images.

Research on the problem of registration estimation has been performed for a while in the computer vision community and many methods were developed. For simple cases, registration can be estimated with the help of external devices such as magnetic position and orientation trackers, robotic arms or even cameras providing images from which position and orientation can be extracted. The latter solution implies very complex image processing and pattern recognition algorithms that are time consuming and rarely fully reliable. A magnetic position and orientation tracking device, such as the Fastrak System from Polhemus, seems to be a promising approach. However, the magnetic fields used to track the pose are very sensitive to the environment. In an experimental setup containing quite a large number of metallic parts such as computer boxes, power supplies and robotic equipments, such a device does not succeed in providing the required pose information except in a very limited workspace and under constrained displacements. When a robotic arm is used to move the sensor from one viewpoint to another, the registration between views can be estimated from the movement of the robot. However, the accuracy depends on the precision of the internal encoders of the robot and on its repeatability. Another problem is that the robot might reveal to be too expensive in some situations or might be unavailable as in hand-held scanning devices. Sometimes, the accuracy of estimation also varies with the cleverness of designers on how to make use of external devices. In summary, approaches based on external devices are not always suitable to provide an accurate, practical and automated estimation of registration parameters. Therefore, there is still a necessity to devise general techniques to solve the registration problem without external input.

An interesting solution in tackling this problem is to take advantage of the raw data provided by the sensor, here a range camera. Assuming that there is an overlap between the areas of the scene that are measured from each viewpoint, it becomes possible to search for some matching characteristics in both sets of information and then compute the necessary transformation parameters that would bring those matching points one onto the other. Based on some assumptions, several techniques have been proposed in the past to develop such a technique, but no extensive and definitive solution has been found yet. These techniques can be divided into two categories. The first group is based on interactive approaches [7, 8, 9, 10]. One of the two range images to be registered is called the model. The other one to be transferred over the model is called the scene. Corresponding points are found iteratively between the scene and the model until they converge. Finally, a rigid motion estimate is obtained from the corresponding points. The method works well if each data point has its corresponding point in the model. However, it requires an initial estimate and provides no guarantee of getting the correct solution. The second group of approaches relies on feature extraction [11, 12, 13, 14]. Some features are extracted from each set of data to compute the registration. No initial estimate is required for this kind of method, but it fails when no prominent local features exist in the dataset and is sensitive to noise. Another disadvantage of this kind of method is that much time is spent on extracting features. Intensity images are sometimes used to assist in extracting feature points [15].

For the techniques based on raw data discussed above, the objective function of the 3D registration process seeks to minimize the distance between corresponding points on overlapping surfaces associated with two data sets from two different points of view. In practice, it is made more complex because range data are often corrupted by measurement errors, outliers and, sometimes, a lack of data. The errors in surface measurements of an object include scanner errors, camera distortion, and spatial quantization, and missing data can result from self-occlusion or sensor shadows [16]. Because of these measurement errors, it is difficult to optimize one cost function to obtain a closed-form solution for a rigid transformation between two sets of noisy 3D points. For example, in the first category of techniques, the cost function is easily minimized locally but not globally based on the given initial estimate. Missing data makes some corresponding points impossible to find and this tends to deviate the computation of the cost function. Moreover, unlike regular grid-based images, raw range images usually have a varying resolution and a non-uniform distribution of measurement error over the entire image. This makes the estimation of registration parameters more difficult. Compared to 2D intensity images, the addition of a supplementary dimension in 3D range images is another concern in 3D modeling as it makes time consuming computation even with the same size of images. All the problems presented here exist in all techniques based on raw data. However, they are not properly overcome in existing techniques. One more robust technique is still expected.

Our objective in this thesis is to derive such a technique that can solve these problems in some optimal sense and estimate the registration parameters that globally register noisy range data provided from different viewpoints. Only a few basic assumptions are made on object surfaces, and the computation time is minimized while obtaining accurate registration.

1.2 Proposed algorithm

In this thesis, we propose a new method dealing with the registration problem that is based on a modified Gauss sphere [17] representation of the object surfaces and homogeneous transformation matrices. Starting from the fact that planar surfaces exist on most objects (even for round objects that can be approximated by many small planar patches), the only assumptions used in the proposed algorithm are that rigid transformations exist between objects or groups of objects, and that there is enough overlap between two views from two different viewpoints to be registered. These basic assumptions are the same that are usually imposed in current 3D modeling applications and that are required for techniques exposed previously with different extents on the overlap constraint.

The proposed algorithm is briefly described as follows. First, straight-line segments are interpolated in each range profile. Within a specific precision, object surfaces along the scan lines can be approximated by these straight-line segments. The normal vector, length and starting point of each line segment are extracted. Then, neighbor range profiles with similar global normal properties are grouped into planar patches. The surface normal vector and the centroid point are then computed for each patch. The area of the patch is considered as a weight that determines the length of the normal vector. Then, those normal vectors with similar orientation are combined together. This step does not only reduce the number of normal vectors, but also significantly shortens computing time. Moreover, this allows one to consider the scene globally which helps in obtaining an accurate transformation estimate. Aligning all compacted normal vectors together on the origin of the coordinate system produces a modified Gauss sphere representation of the object surface. Finally, three pairs of normal vectors extracted from two surface representations to be registered are compared and used to compute the three rotation parameters in a linear way. Next, the three translation parameters are computed by averaging the coordinate difference between centroid points of matching surfaces computed in the second step after applying the rotation transformation.

The advantages of the proposed approach are the following: First, no initial estimate is required. Second, no sophisticated features need to be extracted and therefore computing time is reduced. Third, no cost function is computed bringing no requirement for long iterative minimization. The registration parameters are obtained directly from matched normal vectors. Fourth, rotation and translation parameters can be obtained separately. There is no need to scan 3D rotation and 3D translation parameters by loops over given ranges of values thus saving much computing time. Fifth, the drawbacks of varying resolution and non-uniform distribution of measurement errors in 3D range images can be overcome to some extent. Most points on the object surface can be extracted if a sufficient precision is used to approximate the object surface. This facilitates the matching of points between two data sets. Finally, simplifying sparse 3D data points to planar patches enables fast and efficient computation by reducing the size of data to process. It also acts as a filter as it averages measurement errors which cause problems in other methods when cost functions are used.

Compared to other research works on 3D modeling, the proposed algorithm is also more reality-oriented because it is based on the raw range measurements (scan lines) directly provided by an off-the-shelf sensor. Most other research works assume that full range images are directly provided by the sensors. As a result, they search for matching characteristics between such full images and compute geometrical transformations from there. Such a framework does not correspond to reality because the majority of range sensors currently available on the market or even prototypes found in laboratories do not provide such full images by themselves. They rather generate single points or scan lines of range measurements [18]. Those sensors that artificially generate full images rely on an external mechanical device to translate the sensor or change its orientation [19], or even scale the data into grid-based images. This solution compares to the use of a robot to move the sensor and is sensitive in terms of registration errors.

1.3 Contributions

The contributions of this thesis can be summarized as follows:

a.) Proposing an automatic registration algorithm useful in 3D modeling that does not require initial estimation nor human interaction;

b.) Proposing a method of compact representation of object surfaces which facilitates the estimation of registration between viewpoints;

c.) Proposing an evaluation scheme to select the best solution within one group of choices without any cost function, which makes the algorithm computationally efficient;

d.) Proposing a way to register two data sets at a global convergence;

e.) Proposing a solution that estimates rotation and translation parameters separately which speeds up the estimation of registration.

1.4 Thesis organization

The rest of the thesis is structured as follows: Chapter 2 proposes a review of several relevant papers related with data registration. Next, a detailed description of the experimental setup is presented in chapter 3. This is followed in chapter 4 by the discussion of the principles of the proposed algorithm for both 2D scan lines and 3D range images. Then in chapter 5, considerations on how to implement the proposed algorithm are discussed for real range data. Experimental results are presented and analyzed in chapter 6. Finally, chapter 7 gives a summary of the thesis work and examines avenues for future developments.

Chapter 2 Literature Review

2.1 Introduction

This chapter presents a detailed review of some research efforts that have been performed over the past years to solve the registration problem. The approaches developed have different division schemes according to their characteristics. Here we divide them into two categories. The first category of approaches is based on external assistance. The other category takes advantages of raw data. The latter one is further categorized into two groups based on the techniques they use: Iterative Closest Point (ICP)-based and feature-based. Some approaches using some combined techniques may belong to different categories. The rest of this chapter follows the hierarchical structure of these categories and proposes a detailed discussion of them.

2.2 Techniques based on external assistance

This category of approaches relies on mechanical estimation of poses, manual assistance or both. The objects to be modeled or the sensors are attached to some precisely-controlled mechanical devices. Then the data are acquired from some predefined viewpoints in the experimental setup. Therefore, the registration of the raw data corresponds to the transformation of the mechanical devices and can be used directly for the integration of data in applications. For example, Normand [20] uses one mobile platform with fixed positioning sonar range sensors to construct 3D indoor models for autonomous robot navigation. Miller *et al.* [21] use an autonomous helicopter with a differential global positioning system (DGPS) to build terrain models. Inversely, Turk and Levoy [10] obtain the registration for 3D model construction by keeping the sensor fixed and moving the object on a calibrated platform. Neugebauer [22] extracts the registration by specifying manually some corresponding feature points in pairs of range images. Some other examples using pre-calibrated devices or manual assistance can be found in [23, 24, 25, 26]. Estimating registration based on this category of approaches is an easy and straightforward way. However, as views are restricted to rotation or to some known viewpoints only, the object surface geometry cannot be exploited in the selection of vantage views to obtain measurements [16]. These methods also tend to be expensive and time consuming. They might reveal impractical in some situations. For example, moving a sensor with an articulated arm might be unaffordable in some basic applications. On the other hand, articulated arms' workspace might be too large for small environments. Fortunately, other techniques have been developed to obtain the transformation estimation from the characteristics of range images rather than from external devices.

2.3 ICP (Iterative Closest Point) based methods

Besl and Mckay [9] propose an algorithm for computing the rigid transformation based on iterative theory. Their algorithm minimizes the average distance between two data sets by iteration over the following steps: The correspondence of closest points between the scene shape and the model shape is first computed (the two data sets are specified as the scene shape and the model shape respectively); then the registration is estimated from the closest matched points; finally, the latter registration parameters are applied to the scene shape and it is compared with the model shape. The error (distance difference) is then derived. Iterations stop when the change in mean square error falls below a preset threshold. Besl and Mckay allow the model shape to be represented in any of the allowable forms (point, line or triangle), but only points are used for the scene shape. A quaternion-based algorithm is applied to get the registration given known paired points. Besl and Mckay approach provides good results. The main advantages of the algorithm are that: a) it can handle the full set of six degrees of freedom; b) it does not require preprocessing on the 3D point data; and c) it does not require derivative estimation or local feature extraction. However, it can also be shown that the algorithm tends to converge to a local minimum of the error. This is intuitively clear since the registration stage decreases the average error at each iteration while the closest point determination decreases the individual error. But this does not guarantee that a global convergence can be achieved, and therefore a reasonable initial guess for the rigid transformation has to be provided as a reference to tell if the computed result is right or completely wrong when it is not estimated based on a global convergence.

Many researchers try to improve the algorithm proposed by Besl and Mckay by assuming some initial knowledge or combining it with other techniques. For example, Chen and Medioni [8] compute iteratively incremental transformations that would minimize the distance between the transformed points from the data shape and the points from the model shape by assuming an initial approximate transformation for registration. However, the problem remains that there is no guarantee that point-to-point match exists when computing the distance from one view to another view. The particularity of the approach is that they use range data directly without computing any surface representation or extracting features. As such, a method must provide a way to select a subset of the points, known as control points.

Dorai *et al.* [7] obtain these control points by minimizing the distance from one point to one plane. Therefore only the direction along which the distance can be reduced is constrained. It is shown that the effectiveness of the algorithm for establishing the correspondence of control points in two views depends on the accuracy of the initial transformation and on the smoothness of surfaces to be registered. If the initial estimation is accurate and the object surfaces are smooth, reasonable approximation of the true corresponding control points can be obtained. Otherwise, bad estimation located far away from the true ones are obtained in some cases. Dorai *et al.* also improve the method proposed by Chen and Medioni in [8] by introducing a control point verification process. One equation is applied to check the compatibility of control point pairs. Those that are not compatible with others are removed. Ultimately, the registration is estimated only from these control points that are compatible. Compared to the technique of Chen and Medioni, a higher accuracy can be achieved. However, a drawback of this method is that the extent of surface overlap directly influences the number of corresponding points that can be established as control points. As a result the accuracy of the estimated transformation is affected.

Instead of obtaining control points by extracting features, Masuda and Yokoya [27] integrate random sampling with the ICP algorithm. They classify each range data point into one of the four categories: Occluded, Unpaired, Outlier and Inlier. A point of the scene that is hidden by other points projected on the same pixel of the image is an occluded point. A point that has no paired point on the same pixel in another range image is an unpaired point. The other points are separated into inliers and outliners by thresholding the distance between paired points. In each iteration, Masuda and Yokoya take sub-samples from the given data set. The algorithm succeeds when one sub-sample including no outlier is selected after several trials. It is clearly shown that the effectiveness of the technique depends on the size of the sub-sample, the number of trials of taking sub-samples, and the rate of outlying points. Masuda and Yokoya do not provide a general way to select these factors. Also, computation time is shown to be long (up to 12 minutes) from their experiment.

The algorithms introduced above in [7, 8, 9, 27], as well as some other methods like [16, 28, 29], all need some *a priori* knowledge on the transformations. It makes them less general in solving the registration problem. Other more flexible techniques could advantageously be developed.

2.4 Feature-based approaches

Object surfaces may have many features. Usually corners and edges (both in 3D range images and 2D intensity images) are used as features. However, control points, planar surfaces and spin images are also good features. Under the assumptions of rigidity constraint and a sufficient overlap between the data of two views to be registered, the characteristics of these features on the rigid object surfaces are preserved. Therefore, they can be used to match two sets of data and extract the registration parameters. The following sections review research works related to different types of features.

2.4.1 Control point-based

Chen *et al.* [30] improve the approaches in [8, 9, 31] by choosing some control points based on the geometry constraint instead of using all initial points or a subset of sampled points. Three control points or more are found in the scene data set, called primary point, secondary point, and auxiliary point(s) respectively. Then an exhaustive search is applied in the model data set to find each possibility of three corresponding points with similar geometry constraints. For each possible three-point correspondence, a unique rigid transformation can be determined. Then after the application of the transformation on the scene data set, the number of occurrences where the points in the scene data set are successfully aligned on the model surface is computed. This is called the overlapping number. Finally, the rigid transformation with the largest overlapping number is selected as the solution for the registration. Chen *et al.* [30] introduce one approach with no preprocessing, no local features required and no initial estimate, which seems reasonable and easy to understand. However, there are some limitations in each step. For example, there is no precise way to choose control points in the scene data set. A lot of time is spent on searching correspondence in the model data set. Moreover, using the overlapping number as an evaluation factor is not practical because every range measurement has a different uncertainty level that depends on its position in the scene. Also the whole process has to be rerun if the corresponding control points in the model data set cannot be located as a consequence of noise or missing data. Finally, Chen *et al.* [30] propose to use a random sample consensus scheme in partial overlap cases, but this solution meets the same difficulties encountered in [8, 9, 31].

Roth [15] also addresses the registration problem by an exhaustive search of correspondences based on the rigidity constraint. The difference is that he does not compute the three control points in the scene data set. Alternatively, he extracts feature points from the intensity image and then uses the associated 3D values collected from the range image to compute the registration. After the computation of a 3D Delaunay tetrahedrization based on these 3D range points, all the compatible triangle pairs are tested for matching. The registration is estimated and applied to the scene data set for each triangle pair. He also takes the number of aligning points as an evaluation factor to select one transformation as the solution. Roth provides a very good theoretical description. The proposed algorithm is able to handle geometrically symmetric objects, and it does not require significant sub-sampling of the range image, so the whole information is preserved. However, the algorithm requires that the viewpoints of the intensity images are relatively close together (1/4 of the image size) and there must be a lot of textures on the objects being scanned. This is a significant limitation as most man-made objects are textureless. Another limitation is that a lot of time is spent on finding the interest points, computing the Delaunay tetrahedrization, and searching for matched triangle pairs. Computation time depends on the number of interest points. The experimental results that are presented demonstrate the computational inefficiency and limited robustness of the algorithm originating from the previous limitations. For example, the registration of the 'head' model fails 2 out of 7 times, and succeeds 3 out of 7 times.

Some other research works regarding control points-based approaches can be found in [27, 32, 33].

2.4.2 Spin image-based

Johnson *et al.* [35] as well as [13, 14, 31, 34] propose a new idea in solving the registration problem based on the computation of local surface signatures, called spin images. This method does not need any initial estimate nor any explicit feature extraction. These papers also discuss some problems in dealing with real application data, such as varying resolution, absence of reliable features and the large data set. The signature of a given basis point P (figure 2.1) is computed by projecting every point x within a support region centered at P onto a 2D spin coordinate system, defined by the distance of x to the tangent plane ρ, β, and the distance to the normal line n, α. The spin image at P then can be defined by histogramming the values of α and β. It is shown that the spin image meets the rigidity constraint in transformation, and can be used to estimate the transformation. In a way similar to other feature-based approaches, Johnson *et al.* define the approach as follows: First, they convert the two range images into triangular surface meshes and spin images are computed for every vertex point in the mesh. Then, control points are found and matched by doing a correlation computation on the spin images between two views. Transformations are derived for each group of matched control points and the one that best aligns the corresponding points is selected as the solution; Finally, Johnson *et al.* also refine the solution by the ICP method.

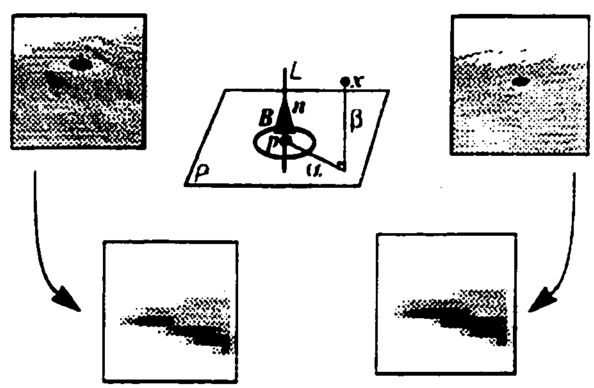


Figure 2.1 Spin image definition (Reprint from [14]).

The main problem in this algorithm is the spin image computation. Not only is it computationally intensive, it is also sensitive to the mesh resolution. If the mesh resolution of a region is too low for the chosen spin image region, too few data points will be accumulated in each histogram bin, which will degrade the spin images. Initially, Johnson *et al.* employed a vertex-based mesh decimation algorithm, which resulted in significant lost of information, and was computationally expensive. In subsequent work they extended the approach to use face-based spin images. Each mesh face is subsampled by linear interpolation in a raster scan pattern before computing its spin image. Significant improvement has been achieved with this strategy. But it is still sensitive to the choice of the sampling rate used for interpolation. The results presented in [13] show that much time is required (~5 minutes) to register two terrain range images.

2.4.3 Triangular mesh-based

Bergevin *et al.* [36] do some interesting research in tackling the registration problem making use of hierarchical surface representation and 4 by 4 homogeneous matrix representation. The approach proposed is basically a three-stage process in which a number of transformations are initially generated before being sorted. A refinement process is added at the end. From the hierarchical surface representation, range measurements are triangulated and represented in a very effective way. Meanwhile, a number of geometrical attributes are extracted for each triangle such as the area, the normal vector, the centroid and the number and label of each of its adjacent triangles. During initial transformation, this approach takes advantage of the fact that corresponding regions (triangles) in two views should have similar global properties such as centroids and normal vectors. This provides some means for pairing the correspondences rapidly while generating reliable transformation. A 4 by 4 homogeneous matrix provides a direct way of getting registration parameters given the correct correspondence. To simplify, a 2D overlap is used to evaluate the initial transformation in the intermediate step instead of considering the full 3D space. This scheme eliminates many bad transformations and only keeps some candidates close enough to the right one. This saves much time for the final refinement process. In spite of its significant advantages, the approach also has some limitations. First, it uses a small triangular representation, which is very sensitive to measurement errors, especially for its normal vector computation. Lots of time is also required to compute the attributes of triangles coming from a large number of small triangles. The authors state that about 20~30 minutes is spent for this triangulation step for each view which is quite unpractical for real applications. The authors also state that the triangulation is not perfectly invariant with respect to the viewpoints. As a consequence, they have to accept a large tolerance level when selecting matched triangle pairs (10 mm distance difference between two centroids; 20^{0} angular difference between two normal vectors). Therefore, the accuracy of the transformation based on this first estimate is usually unreliable. The authors add two final steps to refine the transformation but this still increases computing time.

2.4.4 Edge-based

Stamos and Allen [37] propose to build a system to create geometric and photometric 3D models by extracting 3D linear features. In a similar way to the method proposed by Bergevin *et al.* in [36], Stamos and Allen employ surface representation and make use of the normal vectors to surfaces. The difference is that Bergevin *et al.* triangulate the surface into facets, which makes the representation sensitive to noise and computationally expensive. On the other hand, Stamos and Allen group as many points as possible into a set of planar surfaces, such that noise is minimized when using many raw data and the data set size is reduced. This way the computational efficiency is increased. After having segmented range data, Stamos and Allen detect the 3D lines that correspond to the intersection of two neighboring planar regions. These lines are considered as the candidate correspondences between two views. A quaternion-based solution is used to get the registration estimation. Unfortunately, the approach described in this paper cannot automatically detect the correspondence. The users have to manually select some of the corresponding 3D lines generated above. Some improvements are still required before this approach can be made practicable. Even though no complexity analysis is presented, some interesting ideas are introduced in this paper.

2.5 Other related techniques

This section reviews some techniques that cannot be grouped into one of the above categories but that introduce some interesting ideas.

Szeliski [38] derives a method for motion estimation between two range image frames of the same terrain without feature extraction and point matching. He makes a smoothing spline approximation of data given the assumption that there is no abrupt variations on object surfaces, which is not always the case. A cost function indicating the average of the covariance-weighted Z difference between the points and the surface is minimized while applying a conventional steepest descent algorithm to transform one data set to another data set. He gives each range measurement a different weight based on its distance from the sensor. This can decrease some negative effects coming from the uncertainty in real data points, which vary significantly from the foreground to the background. Szeliski scales the range measurements into a grid-structure image. But this tends to introduce some errors artificially. He presents some interesting results but they are just simple cases with a very limited number of degrees of freedom that are far from real application requirements.

Horn and Harris [39] try to address the same problem as Szeliski [38] by extending the optical flow equation to some constraint equations for the sequentially grided range image frames. This method works faster than the one proposed by Szeliski and provides six degrees of freedom motion estimation. However, it cannot extract a rigid motion with large rotation and translation and thus resumes to the refinement of some initial estimates of the transformation. Kamgar –Parsi *et al.* [40] also derive one technique for registration without distinctive features, but the method is restricted to three degrees of freedom.

Taubin [41] presents a different method for position estimation without feature extraction based on algebraic representation. He interprets the raw measurements in some implicit algebraic forms with one predefined resolution using an approximate distance metric. Global shapes (not occluded shapes) can be identified based on generalized eigen values, and the registration transformation can be recovered. The method works well with complete planar curve and space curve shapes, but has difficulties with complex surfaces. Taubin introduces an interesting theoretical way to solve the registration problem. But much research work is necessary before it can be practicable.

Wong *et al.* [42] and Li [43] propose another idea using attributed graphs for 3D matching. An attributed relational graph of fundamental surface regions for data and model shapes is first built. Then some algorithms perform the graph matching. It is shown that this method has a large tolerance on the variability in attributes and in graph adjacency relationships. But derivative-based attributes extraction still makes it sensitive to noise and the complexity of the scene.

Chapter 3 Experimental Setup

3.1 Introduction

This chapter describes the experimental setup used for our development effort that is available in the Vision, Image, Video and Audio (VIVA) Research Laboratory at the University of Ottawa. The specific configuration of the setup, shown in figure 3.1, has an impact on many decisions made in the design phase. A Jupiter laser range finder, which scans one line on the object surface at a time, is mounted on a 7 degree-of-freedom CRS F3 robot arm to collect the depth information on the scene from a series of viewpoints. The 7th DOF of the robot is provided by the track on which the arm is mounted. There is one controller for the range finder and one for the robot arm. To collect scan lines, the range finder is moved in constant increments along a straight line (along the X axis in the camera coordinate system). while its orientation is kept approximately constant. The steps of this movement are kept relatively small, 5 mm or 3 mm in our experiments, to get a good coverage of the entire object surface. Displacements are under the control of the F3 robot arm operated in the world coordinate mode in order to ensure a precise control of the sensor position along the straight line. After one set of scans from one viewing line is completed, the sensor is moved to a completely different viewing area and the process is repeated. This is illustrated in figure 3.2. Figure 3.3 shows one configuration used in our experiments for one viewpoint, and the associated plot of the depth information collected on the object (a small piece of foam looking like a little chair) is shown in figure 3.4.

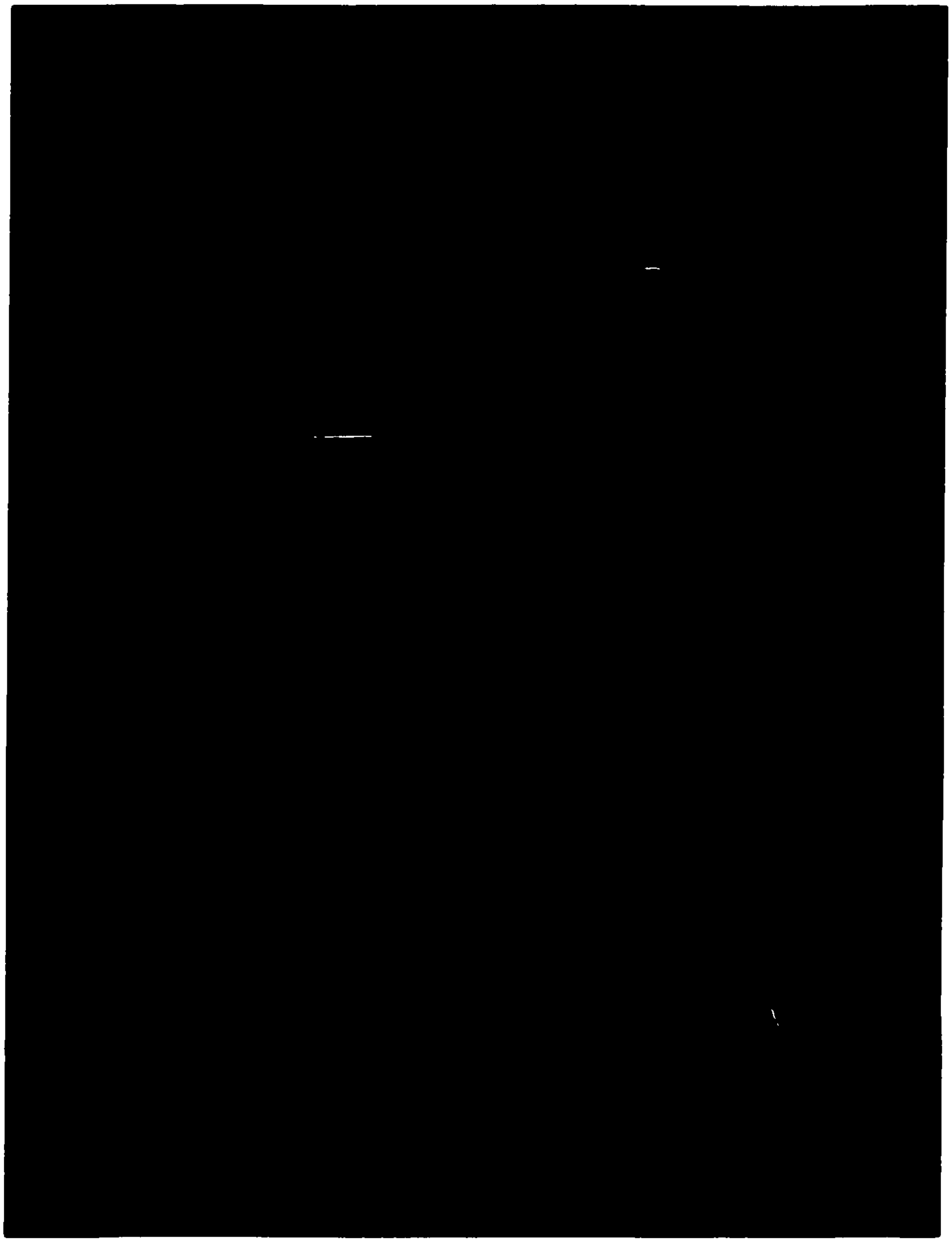


Figure 3.1 Experimental setup.

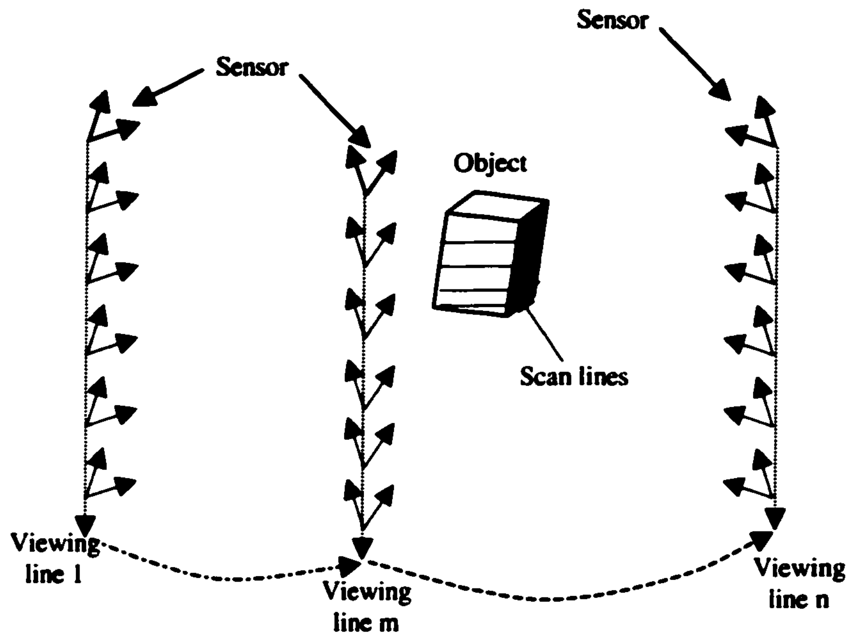


Figure 3.2 Viewing lines.

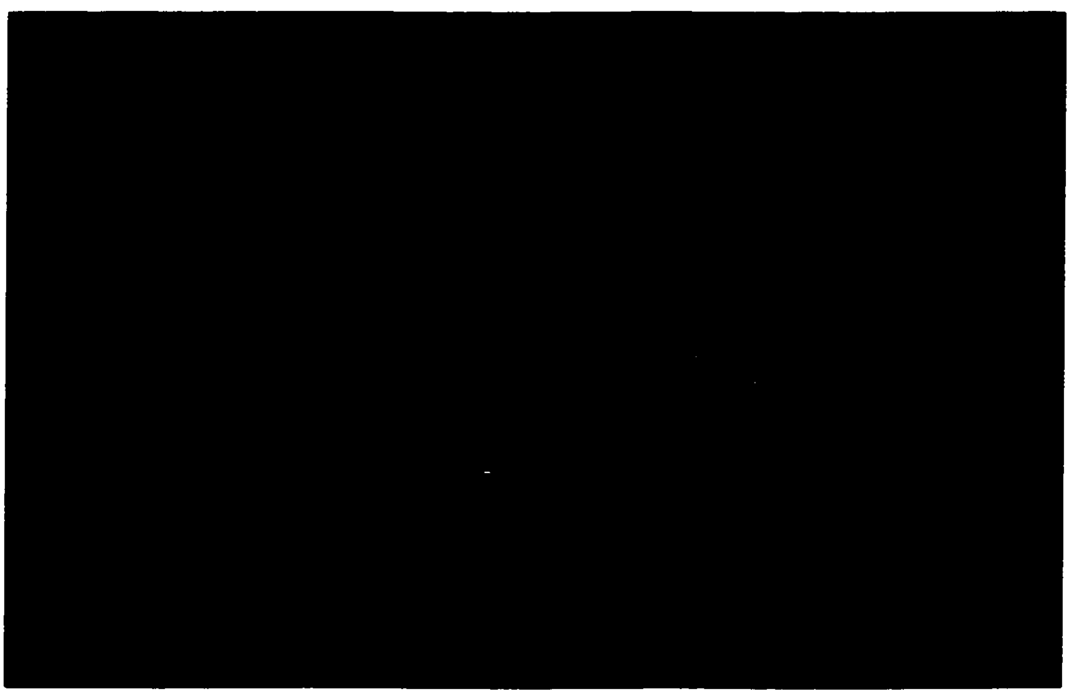


Figure 3.3 Range sensor scanning from one viewpoint.

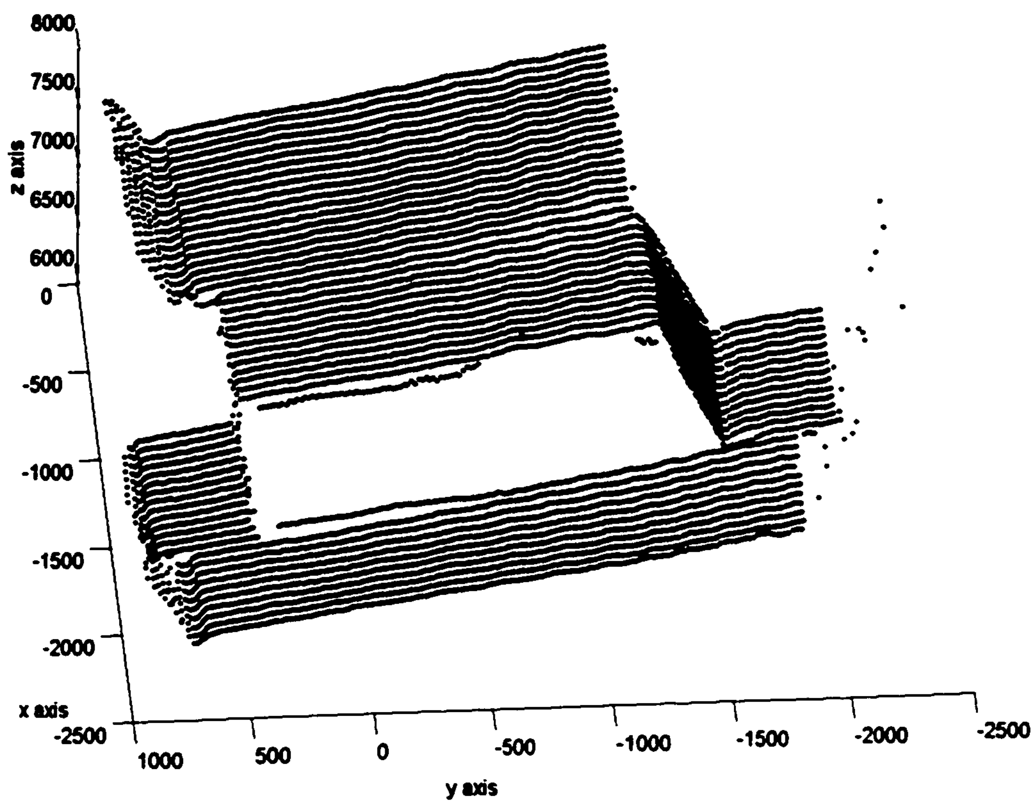


Figure 3.4 Range measurements collected on the piece of foam in figure 3.3.

The measurement precision achieved by the Jupiter range finder depends on the depth and on the angle between the laser beam and the object surface. The precision can be up to 0.3 mm in depth and 0.7 mm along the direction orthogonal to that of depth. On the other hand, the CRS F3 robot arm provides a resolution of 0.05 mm in translation and 0.005 degrees in rotation. Compared to the Jupiter range finder, the F3 robot arm has a much higher resolution. Therefore, we use it to provide the theoretical registration parameters for the evaluation of our algorithm. The extraction of the theoretical registration parameters is illustrated in figure 3.5. R_1 and R_2 are the robot arm base reference frames for two points of view. T_1 and T_2 are the corresponding end effector (Jupiter sensor) reference frames. The transformation matrices Q_1, Q_2 and Q_3 are obtained directly from the joint parameters provided by the robot controller for each configuration. As a result, the transformation matrix, Q, from reference frame T_2 to T_1 is easily computed from the following equation:

$$Q = Q_1^{-1} Q_3 Q_2 \tag{3-1}$$

Where Q corresponds to the registration between the two sets of measurements captured from these two points of view.

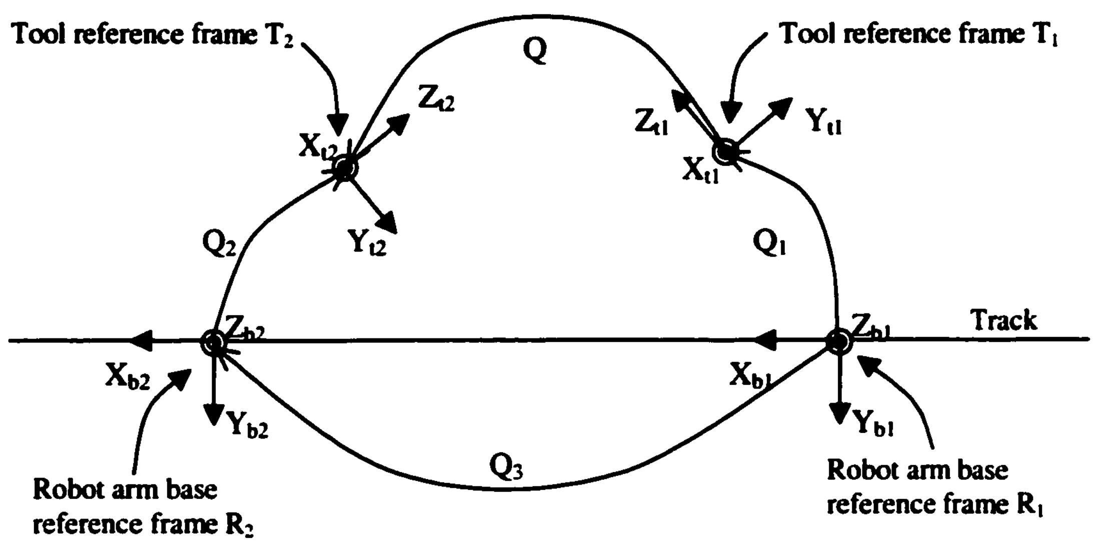


Figure 3.5 Theoretical registration parameters extraction.

As a first part of our research work on the registration problem, we realized a close investigation on the Jupiter laser range finder and the CRS F3 robot arm in order to make the setup work properly and to be able to calibrate the measurements according to the configuration. The following sections examine these two devices in more detail.

3.2 Jupiter laser range finder

3.2.1 Introduction

The Jupiter laser range finder is a 2D vision system from Servo-Robot Inc that is based on a technology developed at NRC. It has two major components: the camera head and the controller, which are interconnected by a cable. The camera head, or range finder, is a distance-measuring device based on an active optical triangulation principle. The controller maintains the camera head in its optical operating condition by adjusting in real-time the laser power and/or the sensitivity in order to cope with varying surface conditions and textures. As shown in figure 3.6, the Jupiter sensor can provide range measurements up to 1005 mm beyond a blind zone of 170 mm and over a wide field of view (close plane: 108 mm, far plane: 579 mm). Jupiter scans a single line profile and can work in two modes. The first mode collects 512 points per profile. The second one gathers 256 points per profile with up to 40 profiles per second.

3.2.2 Reference frames

The raw range profile obtained from the Jupiter sensor is based on a spatial orthogonal reference frame. According to the manufacture calibration, the measurement reference frame, X'Y'Z', is located 1389.6 mm in front of the camera along the Z axis. There is also an offset of -38.9 mm along the Y axis. Therefore, to facilitate the visual interpretation of the raw data with respect to the reality, we have to make some transformations. The data processing module that we developed operates the following coordinate transformations to represent the depth information on the object surface with respect to a reference frame XYZ rigidly attached to the front part of the camera:

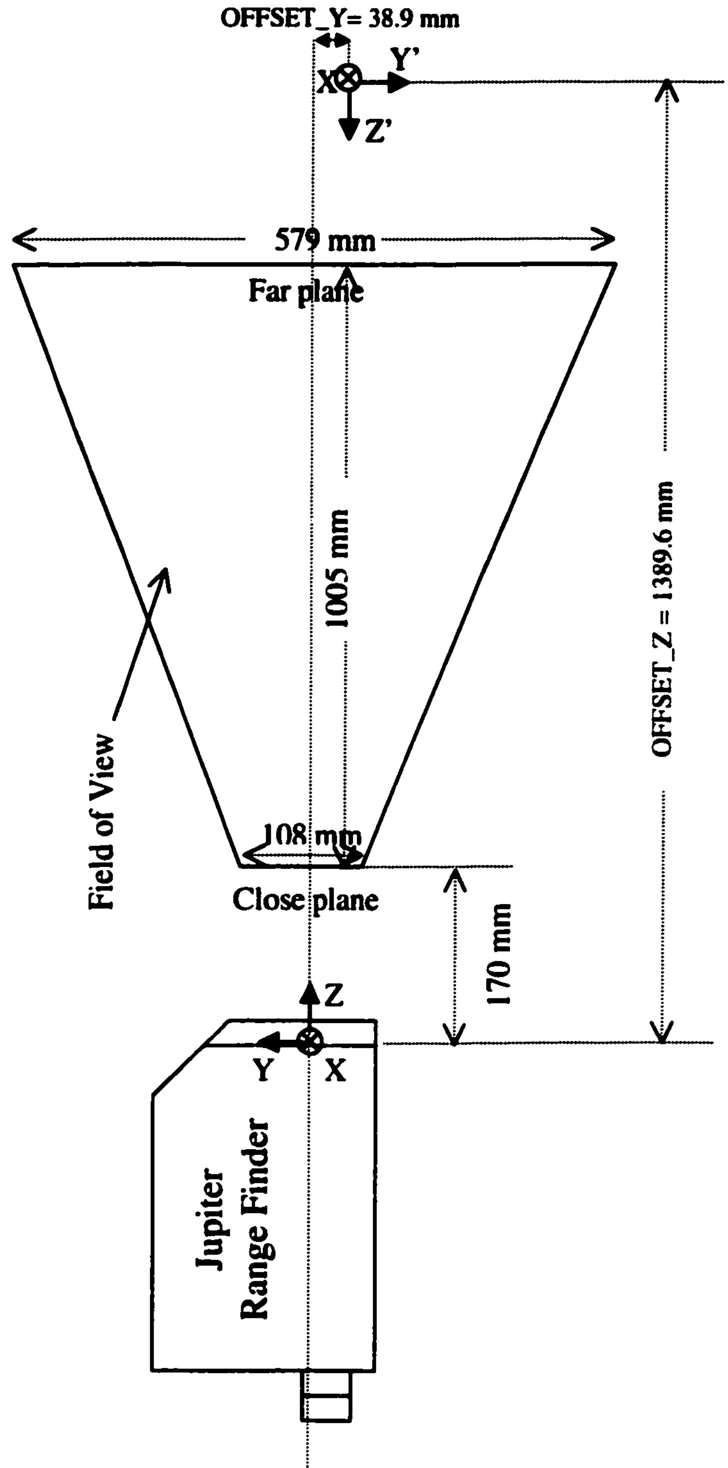


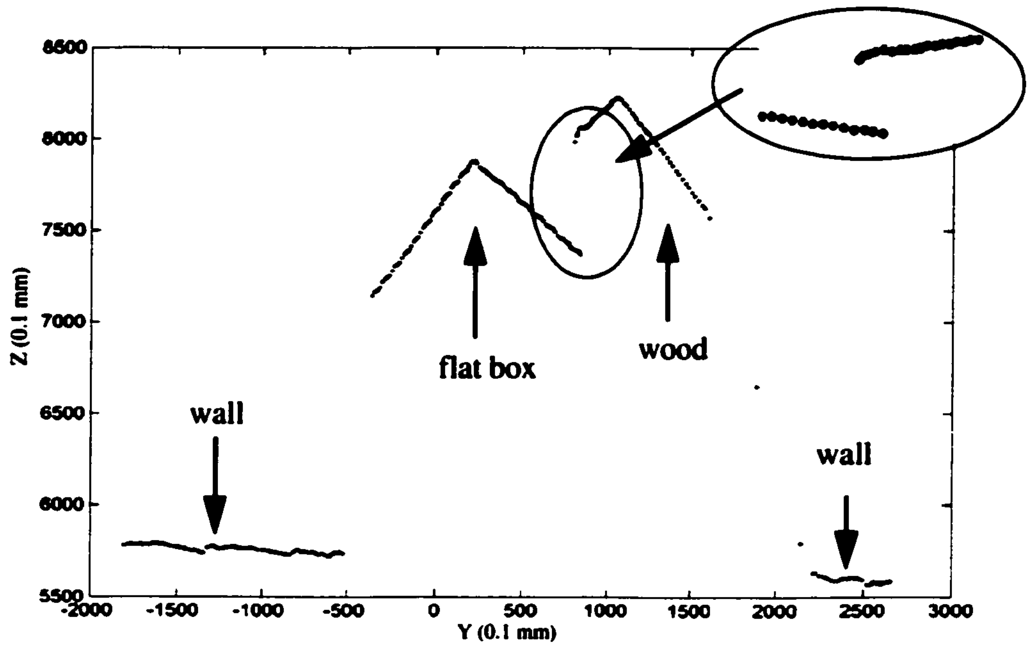
Figure 3.6 Jupiter field of view and reference frames.

$$Y = OFFSET\_Y - Y' \quad (3\text{-}2)$$

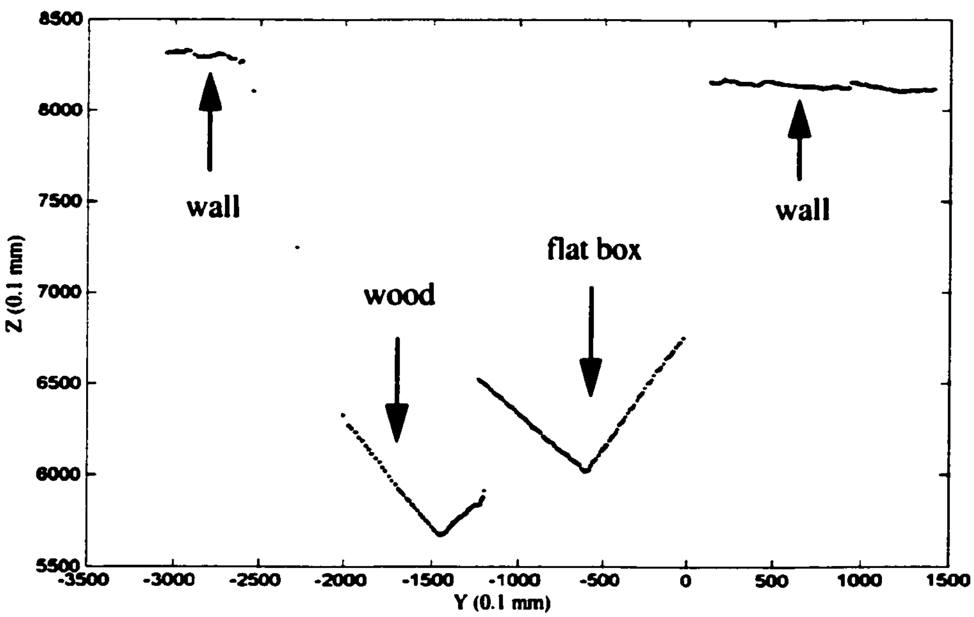
$$Z = OFFSET\_Z - Z' \quad (3\text{-}3)$$

with OFFSET_Y = -38.9 mm and OFFSET_Z = 1389.6 mm. Figure 3.7 shows an example of a characteristic scene that has been used for initial investigation. One of its corresponding 2D profiles is shown in figure 3.8. Figure 3.8 (a) illustrates the raw profile generated by the sensor and that corresponds to the surface shape in inverse order. Figure 3.8 (b) shows the profile after the coordinate transformation has been applied and the right surface representation is recovered. The WinUser software interface provided with the camera stores each profile in a *.3dx format file, which uses –32768 to record points where no relevant measurement is obtained because of the lack of a sufficient reflection of the laser beam on the surface towards the sensor. Such invalid points originate from holes on objects surfaces or from areas that are out of the camera field of view as well as from surfaces not properly oriented. Appendix A provides a detailed description of the useful part of a 3dx file format to allow extraction of relevant information.

Figure 3.7 One simple scene for profile investigation.



(a) One raw scan profile before reference frame transformation.



(b) Corresponding scan profile after reference frame transformation.

Figure 3.8 One scan profile collected on the surface of objects in figure 3.7.

3.2.3 Precision analysis

In order to get a close idea of the accuracy of the raw data provided by the sensor, an investigation has been conducted on a large number of profiles. These profiles were obtained by scanning a flat surface, which was always kept parallel with the Y' axis as shown in figure 3.9. The flat surface has been placed in front of the camera at different distances successively (Z' = 200, 300, 400, 500, 600, 700, 800, 900, 1000, and 1100 mm with respect to the X'Y'Z' reference frame). The ten corresponding raw range profiles are presented in figure 3.10. From these profiles, we found that the measurements have varying resolutions along the Y' and Z' axes, and contain some errors. The following sections detail this investigation.

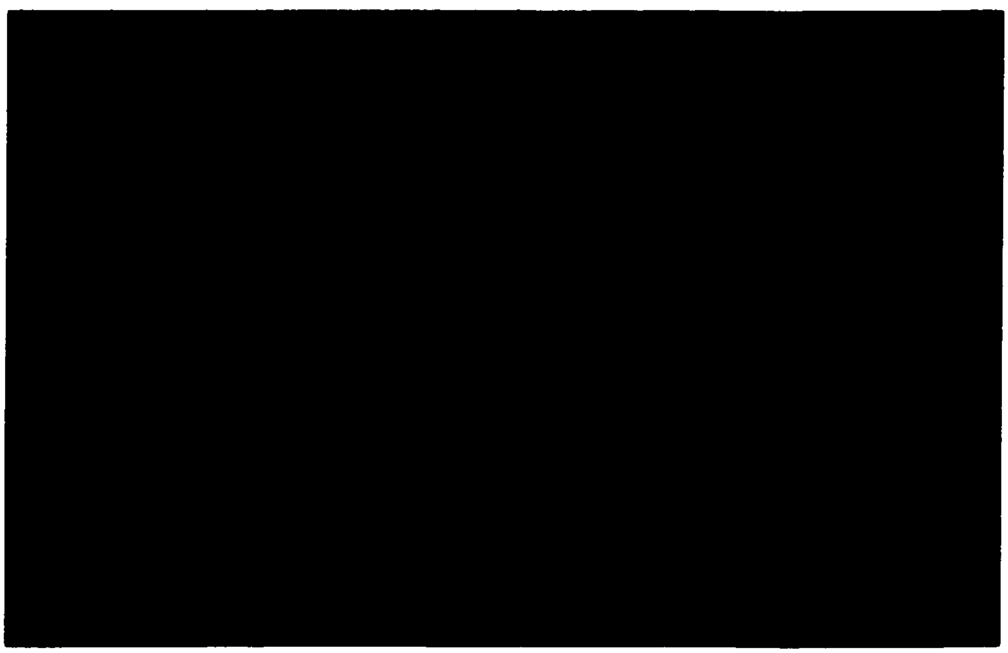
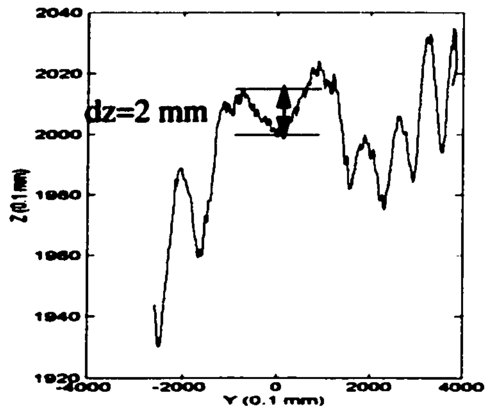
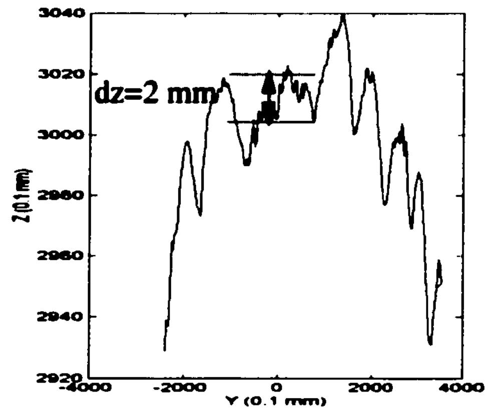


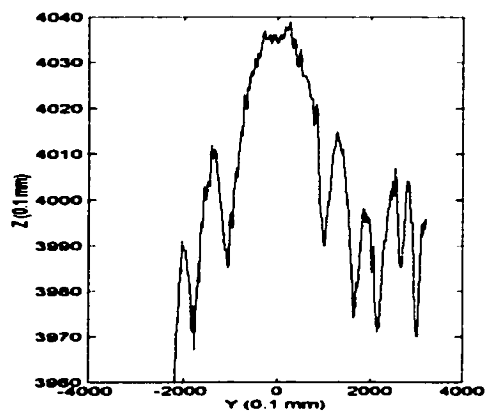
Figure 3.9 Experimental setup for precision analysis.



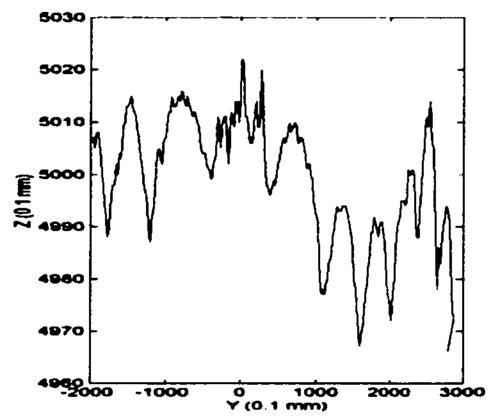
(a) Z' = 200 (mm)



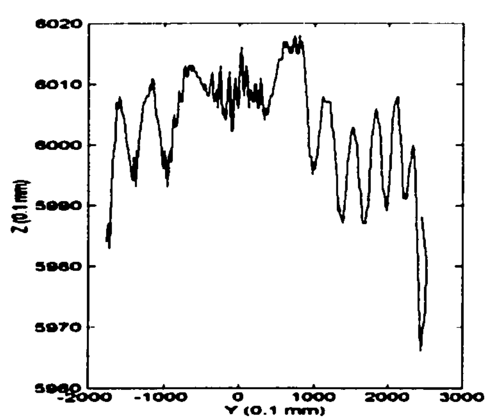
(b) Z' = 300 (mm)



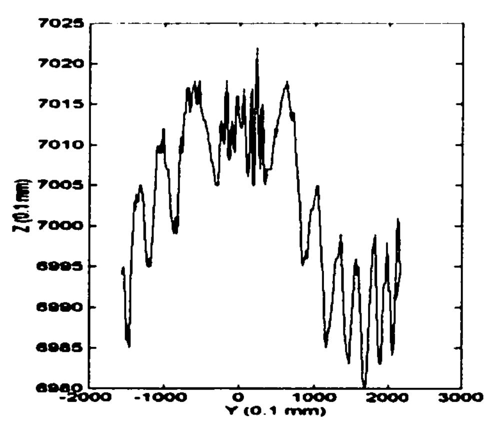
(c) Z' = 400 (mm)



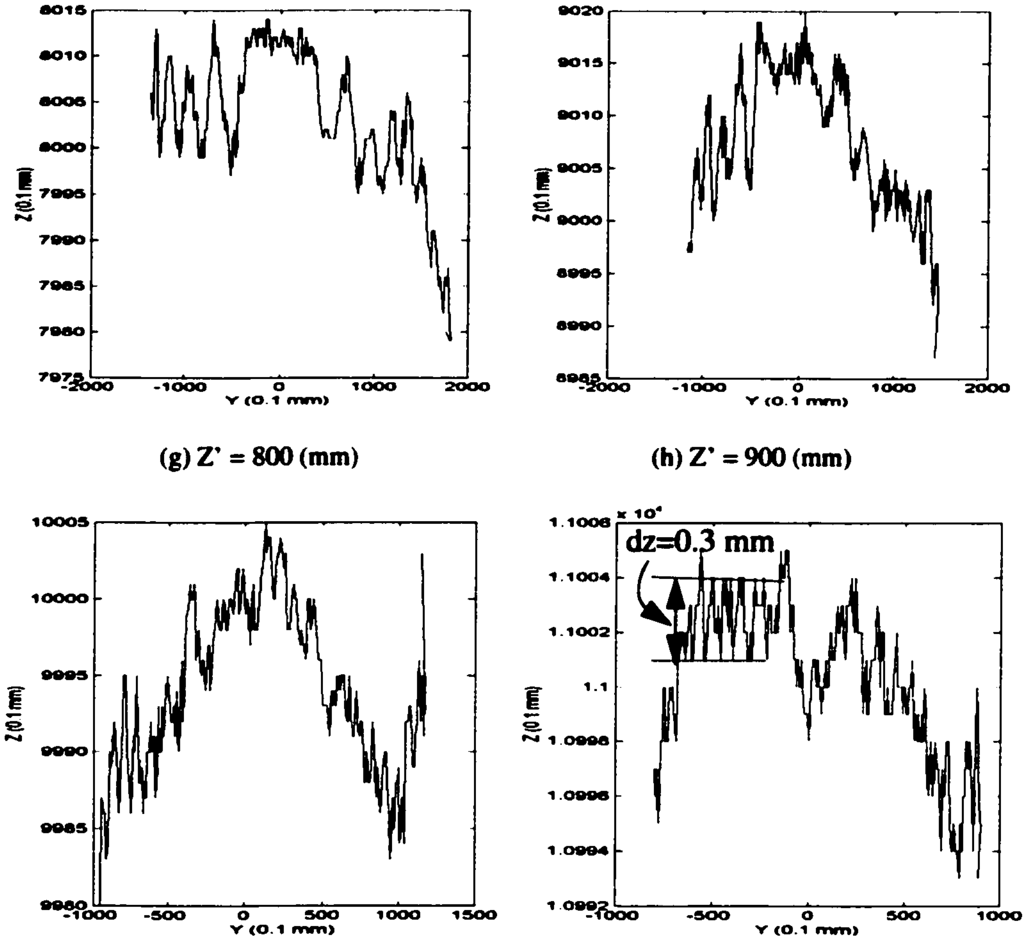
(d) Z' = 500 (mm)



(e) Z' = 600 (mm)



(f) Z' = 700 (mm)



(g) Z' = 800 (mm)

(h) Z' = 900 (mm)

(i) Z' = 1000 (mm)

(j) Z' = 1100 (mm)

Figure 3.10 Ten raw range profiles obtained with the experimental setup in figure 3.9.

3.2.3.1 Resolution along the Y' and Z' axes

By computing the distance between two successive measurement points, figure 3.11 shows the average resolution along the Y' axis at different depth levels. When the surface is close to the camera, the resolution is maximum (say if Z' = 1100 mm, $\overline{\Delta y}$ = 0.7 mm) and decreases with the increase of the distance between the surface and the camera (say if Z' =

200 mm, $\overline{\Delta y}$ = 2.5 mm). This comes from the fact that the camera always takes 256 or 512 samples whatever the width of the field of view is. Actually, the resolution still changes along Y' even for a constant depth. It is higher in the middle than at the extremities of the field of view as shown in figure 3.11. This disparity is explained by the triangular-based principle of the Jupiter laser camera as shown in figure 3.12. The constant angular resolution of the laser beam between successive measurement points leads to different distances (Δy) along the Y' axis. Space between measurement points is smaller in the middle of the field of view and it goes larger at the extremities. If we let Δy_{e1}, Δy_{e2} and Δy_m represent the resolutions at the extremities and in the middle of the field of view respectively, Δy_{e1} and Δy_{e2} are always larger than Δy_m, while they are similar to each other because the field of view is approximately symmetric with respect to the Z' axis. Figure 3.13 and 3.14 show two examples where the flat surface is measured at Z' = 200 mm and Z' = 700 mm respectively. The horizontal axis is the index of sample points from the range profiles. The vertical axis presents the resolution along Y' at each sampled point. When Z' = 200 mm, the resolution in the middle of the range profile is about 2.3 mm, while it reaches about 2.8 mm and 2.9 mm at the extremities. When Z' = 700 mm, the resolution in the middle of the range profile is about 1.3 mm, while it is about 1.7 mm at the extremities. The full plot of the Y' resolution (Δy) versus Y' and Z' is shown in figure 3.15. As described above, when Z' increases, Δy becomes smaller thus providing a higher resolution along Y'. Δy also demonstrates some symmetry along the Y' axis, being smaller around the center of the profile.

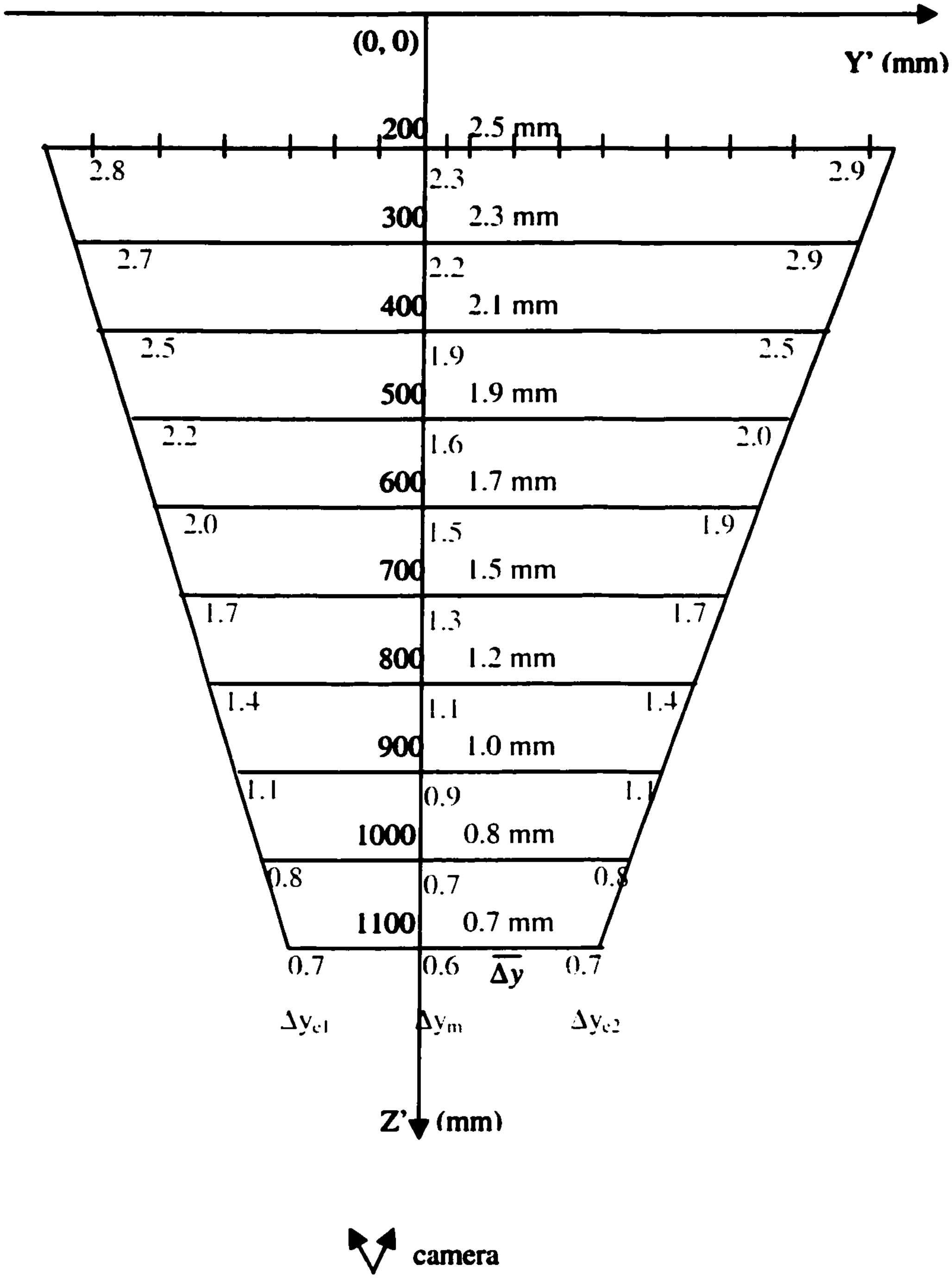


Figure 3.11 Resolution along Y' at different depth levels.

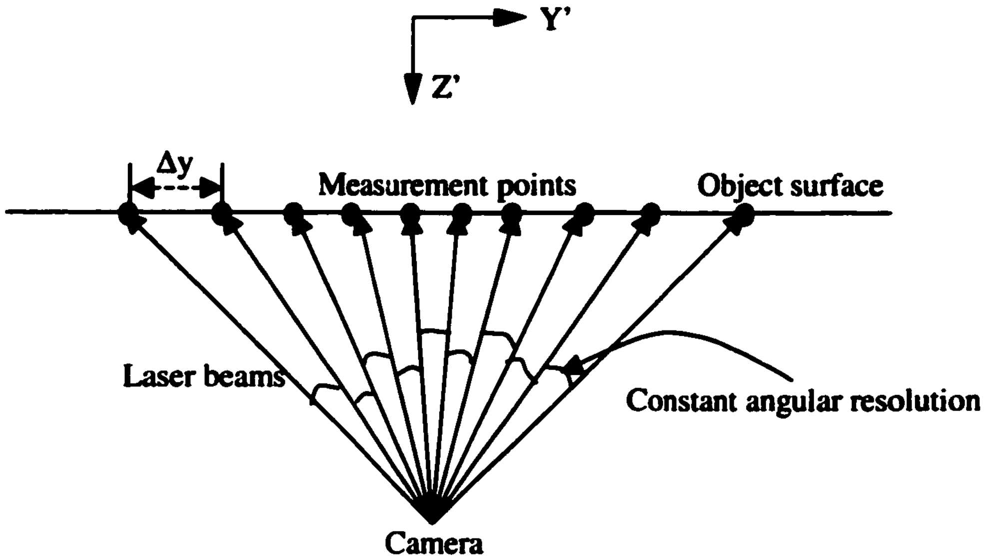


Figure 3.12 Varying resolution along the Y' axis for a constant depth.

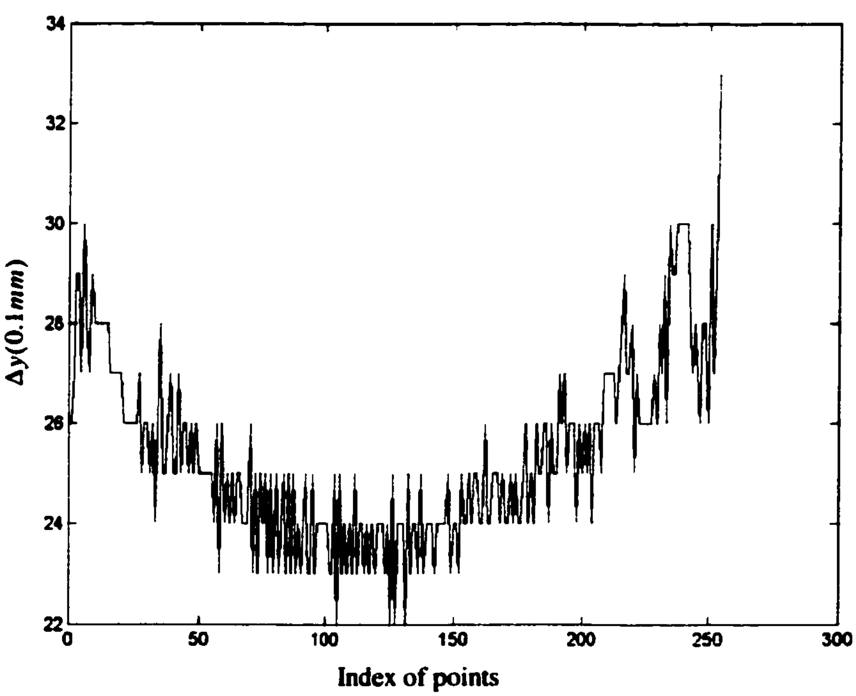


Figure 3.13 Measurement resolutions along the Y' axis at Z' = 200 mm.

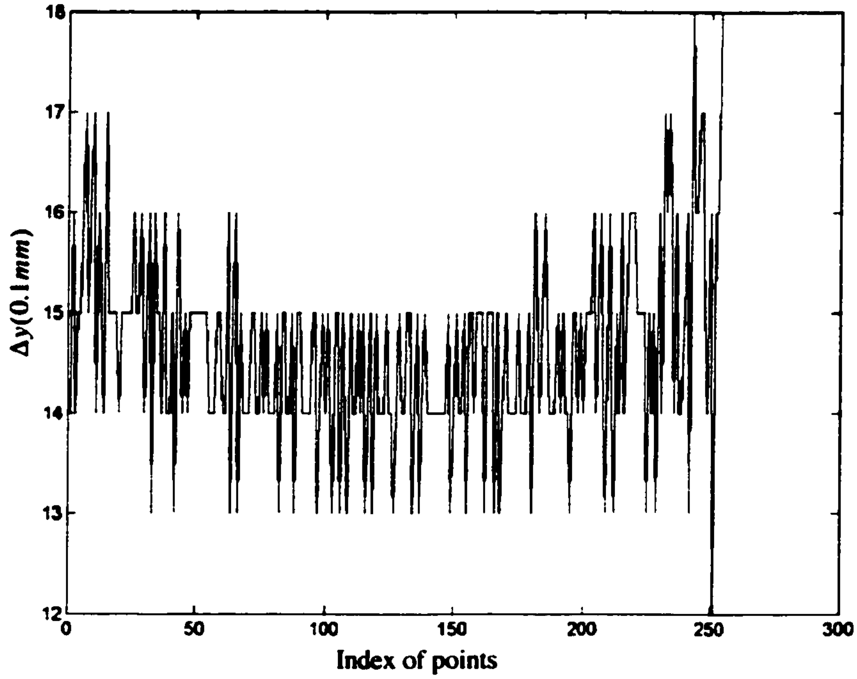


Figure 3.14 Measurement resolutions along the Y' axis at Z' = 700 mm.

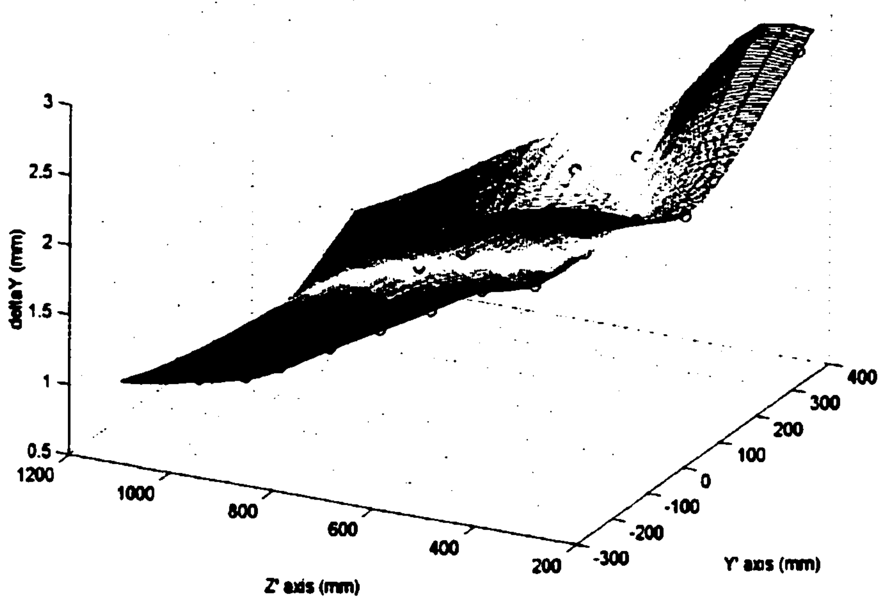


Figure 3.15 Y' resolution (Δy) versus Y' and Z' axes.

From the profiles in figure 3.10, we can also compute the resolution along the Z' axis in Y'-Z' space. It varies from 0.3 mm in (j) to 2 mm in (a). In a way similar to the extraction of the resolution along the Y' axis, an estimate of the depth resolution along Z' versus the Y' and Z' axes has been computed by estimating the resolution for some specific points in each range profile. Figure 3.16 shows a plot of the sensor resolution on depth measurements (Δz). It increases with Z' as objects are closer to the camera and shows symmetric behavior on both sides of the Z' axis with higher resolution for small Y' coordinates.

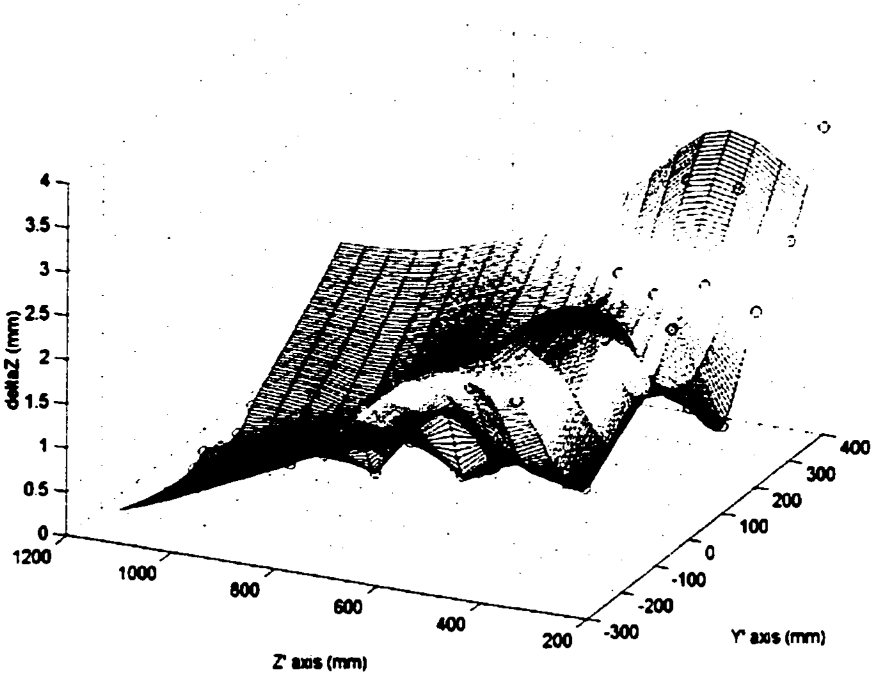


Figure 3.16 Z' resolution (Δz) versus Y' and Z' axes.

3.2.3.2 Measurement errors

Pursuing the investigation on the profiles, we also observed three main types of measurement errors. First, from the above experimental setup, we should ideally get straight-line profiles, but real profiles are somewhat different as shown in figure 3.10. The sensor tends to provide somewhat larger measurements around the center of the scan with variations going from 2 mm to 8 mm in magnitude. This may result from the optical triangulation principle of the Jupiter sensor. When the laser beam is perpendicular to the object surface, the CCD in the sensor receives a higher laser intensity in spite of the automatic adjustment on the laser power provided by the controller. This phenomenon introduces variations in the depth estimation of those points scanned by a perpendicular laser beam in spite of the fact that the distance between the Y axis of the sensor and the surface is constant. In the above experimental setup, the surface is perpendicular to the laser beam in the center of the field of view, and inclined with respect to the laser beam at the extremities. The magnitude of these variations in the depth estimation also increases when the object is moved away from the sensor. As a result, some compensation might be considered in future work to avoid too important deviation from the reality, and to build more accurate 3D models.

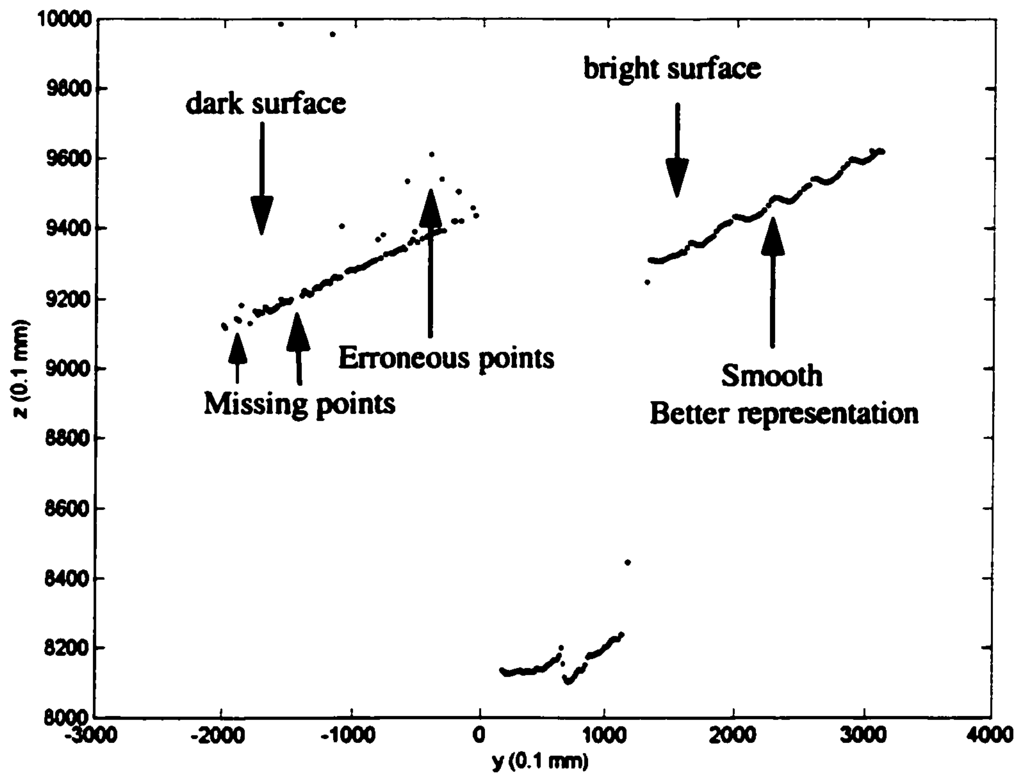
Second, using the object setup shown in figure 3.7, we observed the occurrence of some overlapping points within the region of $Y' \in [500, 1000]$ (0.1 mm) shown in figure 3.8 (a). Practically, this phenomenon happens when two object surfaces overlap along the Y' axis when they have a large depth variation along the Z' axis, and when the farther object surface has a good reflection and is inclined with respect to the laser beam. In figure 3.7 specifically, the flat box surface, with a depth variation to the wood surface, has a good reflection and it causes the problem. This issue may result in some troubles when we try to locate the surface by interpolation points in between such profiles. In order to limit the negative impact that this may have on the registration estimation, these points can be removed if the depth variation is large. Another solution consists in selecting the end points of the line segments that will be introduced in the next chapter using the Y values rather than the index of the sampled points.

These methods can minimize the problem because the number of such overlapping points is usually limited and they can be considered as outliers. The overlapping points can also sometimes cause difficulties when extracting points for planar patch fitting. Therefore, an efficient algorithm has to be built during the point extraction process.

Third, the scan profile is also sensitive to the texture of the object surfaces. Smooth objects show much better in the scan profiles than rough objects. This phenomenon is illustrated in figure 3.17 where two similar foam objects are scanned. One of them has a rough gray surface. The other one is covered with white paper to make it smoother. The corresponding scan profile is shown in figure 3.17 (b). It is clear that the profile collected on the object covered with white paper is more representative of the reality but more Gaussian noise. However, the quality of scan profiles also depends on the angle between the laser beam and the normals to the objects surfaces. For example, on a specular surface like the PC screen, we get clear scan profiles except when there is a large angle between the laser beam and the normal vectors to the screen. We also validated above statements by scanning objects with many different textures, such as foam, a gray computer box, a wood bar, a red plastic cone and a chair covered with brown fabric.

There are some other sources of errors in the measurement process, such as the errors originating from the object surface diffusion, the complexity of objects, and the optical properties of the laser beam. Fortunately, the automatic power adjustment function of the sensor controller succeeds in significantly reducing the impact of some of these problems. However, we had to ensure a limited complexity of the scene in our experiments in order to obtain clear and meaningful range profiles.

(a) Two similar foam objects with different texture on their surfaces.



(b) One scan profile from the object setup in (a).

Figure 3.17 The sensitivity of Jupiter sensor on different textures.

3.3 F3 robot

The F3 robot arm is a 'human scale robot' from CRS Robotics Inc. It provides a powerful combination of high–speed flexible automation, increased reliability and ease-of-use. The six DOFs (Degrees Of Freedom) of the robot allow its end effector to cover a relatively wide workspace both in position and in orientation in an easy and efficient way. The track provides a 7th DOF to the robot that still enlarges the workspace and provides a more flexible setup.

The F3 robot arm employs four coordinate systems to describe its workspace: World, Tool, Joint and Cylindrical systems. The world coordinate system is defined with respect to the world Cartesian reference frame located at the base of the robot (see XYZ in figure 3.1). We can easily control the rotations and translations of the end effector, here holding the Jupiter camera, in this reference frame. The tool coordinate system of the F3 robot is located at the robot end point as shown in figure 3.18. It is useful when we care more about the local movements of the end effector than that of the robot itself. The F3 robot arm also allows more flexible control of the movement of the end effector by defining the tool center point (TCP) with respect to the tool coordinate system. This feature is interesting as we are usually more concerned about the movement of the TCP rather than that of the point at the extremity of the robot. If a TCP is defined, all movement information provided by the F3 robot arm system is referred to the TCP directly. Therefore, no transformation is required. In our experiments, we have set TCP = $[0|_{Tx}, 0|_{Ty}, 210\ \text{mm}|_{Tz}, 0|_{Yaw}, 0|_{Pitch}, 0|_{Roll}]$. Theoretically, this value is measured from the transformation between the robot tool coordinate system and the sensor coordinate system (the one after transformation), as shown in figure 3.18. However, we did not have a precise mechanical calibration for the bracket that we used to attach the Jupiter sensor to the robot end effector. Therefore, the TCP location that we used is only an estimate, which will influence the accuracy of the theoretical registration parameters we extract for reference. Once the TCP is set, the F3 robot arm can provide the registration between the sensor coordinate system and the robot base system directly. This representation can be used to obtain the theoretical registration parameters between two sensor coordinate systems corresponding to two different points of view, as described in the first part of this chapter. The joint system is defined with respect to the robot's joints, which are precisely controlled by pulse counters. Each joint contains an encoder that generates pulses as it rotates (569 pulses per degree for joint 1, 2 and 3, and 455 pulses per degree for joint 4, 5 and 6.) In our experiments, we used the joint system to accurately control the rotation of the sensor. Specifically (with respect to the tool system), joint 1 controls the rotation around the X axis (Yaw); joint 5 controls the rotation around the Y axis (Pitch) and joint 6 controls the rotation around the Z axis (Roll). The joint specification is shown in figure 3.19.

The F3 robot arm provides two types of movements: straight-line motion and rotation. Straight-line motion can only work in world coordinate mode, but it is useful when we must maintain the level or the orientation of the payload. In our experiments, we made use of it to move the camera within one viewing region along a straight line to collect a series of parallel range profiles separated by 5 mm or 3 mm increments. Rotation motion was used along with straight-line motion when the camera was moved from one viewing line to another.

Figure 3.18 F3 robot tool coordinate system and TCP (Tool Center Point).

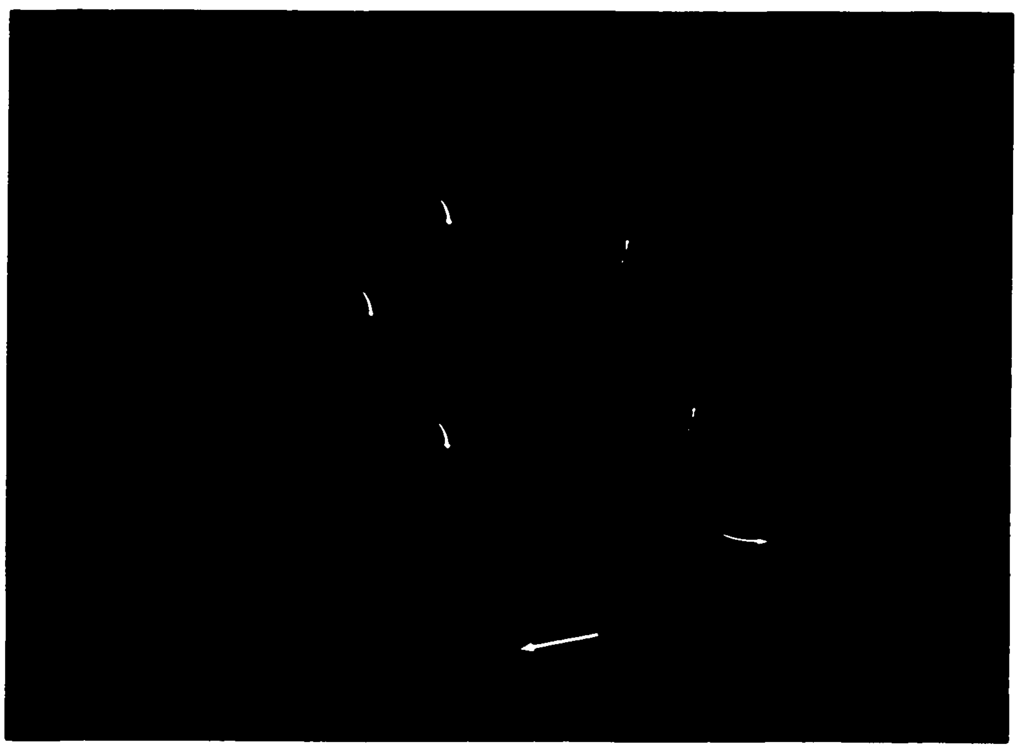


Figure 3.19 F3 robot arm joints specification.

Finally, the movements of the F3 robot arm can be controlled in three ways: with the teach pendant device, from terminal commands or by RAPL-3 programs. The teach pendant can move the robot in world, joint and tool systems. It is easy to use but it cannot move the end effector to a precise pose because it is done manually. We mainly used it to move the sensor to an initial pose for scanning, for which high precision is not critical. Terminal commands operate the robot arm in a similar way as the teach pendant. However, it is not really automatic control and we didn't use it extensively. The RAPL-3 language, which is similar to C programming language in syntax, is a powerful way to precisely control the movements of the F3 robot arm. It consists of many easy-to-understand commands for basic applications and powerful commands for experienced programmers and system integrators. We developed a program to move the camera during our data gathering process and it is presented in appendix B. This RAPL-3 program can be run both by the teach pendant or terminal commands.

Chapter 4 Algorithm Description

4.1 Introduction

Starting from design requirements and goals analysis, this chapter presents a detailed description of the proposed algorithm for the registration estimation system. The first assumption that is made is that most man-made objects are composed of a set of planar surfaces. Even objects of higher complexity can be approximated by a set of planar surfaces that are easy to represent. Assuming also that the rigidity constraint is validated (i.e. solid objects are assumed.), these planar regions remain unchanged when the whole objects are moved or the viewpoint is changed. As a result, the reality that is projected on the image plane of a range sensor only depends on the relative position and orientation of the sensor. This observation is useful in estimating the registration between two or more sets of measurements gathered from different viewpoints on the same group of objects. These assumptions are equivalent to those made in most research works on the issue of registration estimation. The difference with other related techniques is that the approach introduced here relies on a simpler surface representation – a Gauss sphere that maps sparse 3D range points to planar patches. Patches with the same orientation (within some tolerance level) are united together without taking into account the geometrical connectivity between them. It makes the representation very compact. Line segment approximation applied on the raw range profiles speeds up the merge of the profiles, and in turn, the computation of the planar surfaces. This way, the main part of the effort is dedicated to the estimation of the position and orientation variations of the sensor between viewpoints in 3D space, that is the computation of registration parameters, rather than to a sophisticated surface modeling technique or feature extraction procedure.

Following the assumptions discussed above, we list the design requirements and goals for the proposed registration estimation algorithm as follows. The approach should:

a.) provide an accurate registration estimation for sensor positioning systems;

b.) not require any *a priori* knowledge about the transformation information;

c.) place minimum requirements on the object surfaces;

d.) exclude human interactions;

e.) be computationally efficient;

f.) make use of the raw data provided by Jupiter range sensor;

g.) be easily adaptable to various types of range sensor data;

h.) be implemented using the experimental setup described in chapter 3.

Starting from these requirements and goals, the architecture of the proposed algorithm illustrated in figure 4.1 has been developed. The algorithm begins with segmenting the input data to locate the linear or planar sections and to approximate them with straight-line segments, $Line_i$ (2D case), or planar patches, $patch_i$ (3D case). The normal vectors, n_i, and the centroid points, cp_i, are extracted for each segment, as illustrated in figure 4.2 and figure 4.3. The normal vector and the length (2D case) or the area (3D case) of each segment are used to encode the surface representation as a modified Gauss sphere, which significantly simplifies the search for rotation parameters. For the 2D case, since the normal vectors in the Gauss sphere are distributed on a plane only, the compact representation is a circle rather than a sphere. Therefore, we call it a modified Gauss circle rather than a modified Gauss sphere for the 2D case in this thesis.

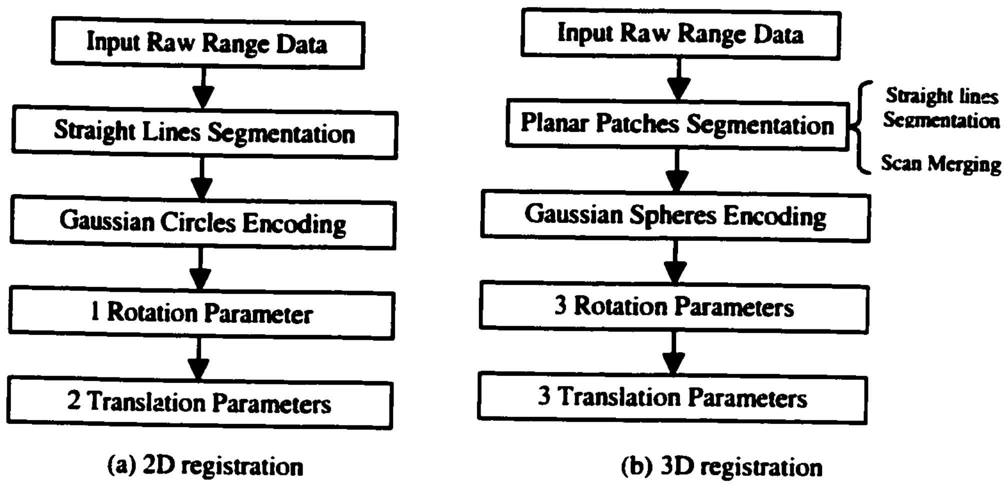


Figure 4.1 Theoretical system architecture.

Provided two corresponding Gauss circles or spheres, the next step consists in estimating the rotation parameters by finding the appropriate rotation matrix that makes similar vectors on the circles or spheres overlap. Finally, the translation parameters associated with the displacement of the sensor between viewpoints are estimated by computing the necessary shift of the centroids that makes corresponding segments match.

In the remaining part of this chapter, we discuss the 2D and 3D registration cases separately because different elements (line segments in 2D case; planar patches in 3D case) are employed to represent the object surface. However, 2D registration can be seen as a special case of 3D registration, where the object surface is assumed to be always vertical. Some issues about the implementation are discussed in the next chapter. In most practical applications, the aim is to estimate registration for the 3D case. However, the 2D case is detailed for clarity and because the ideas to obtain 3D registration are an extension of the solution for 2D registration.

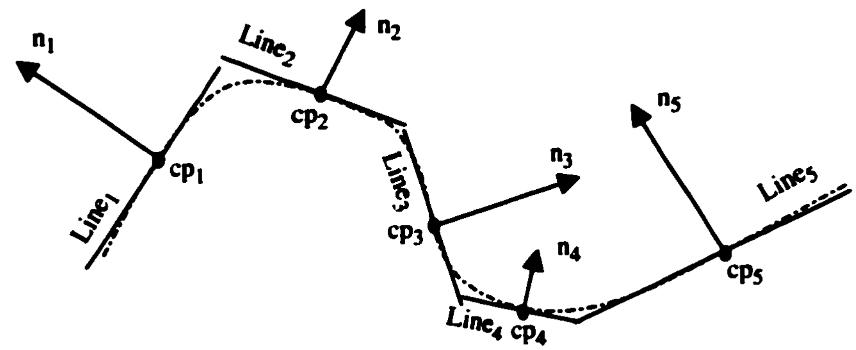


Figure 4.2 Segmentation of one 2D scan profile.

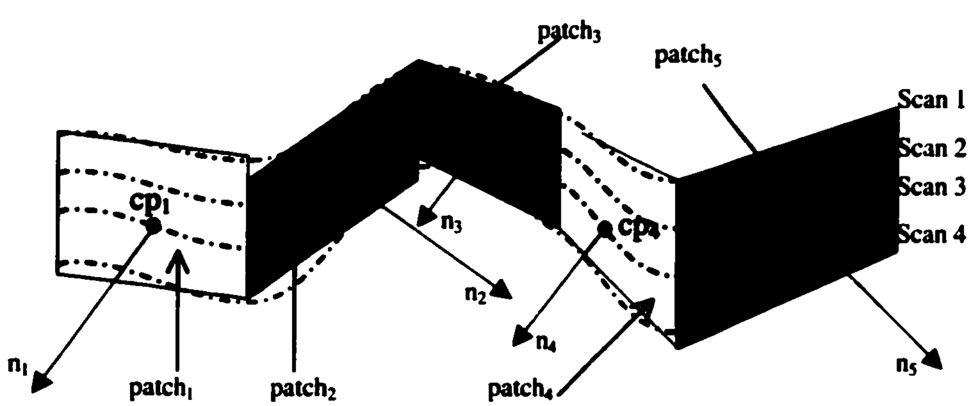


Figure 4.3 Segmentation of a set of 3D range profiles.

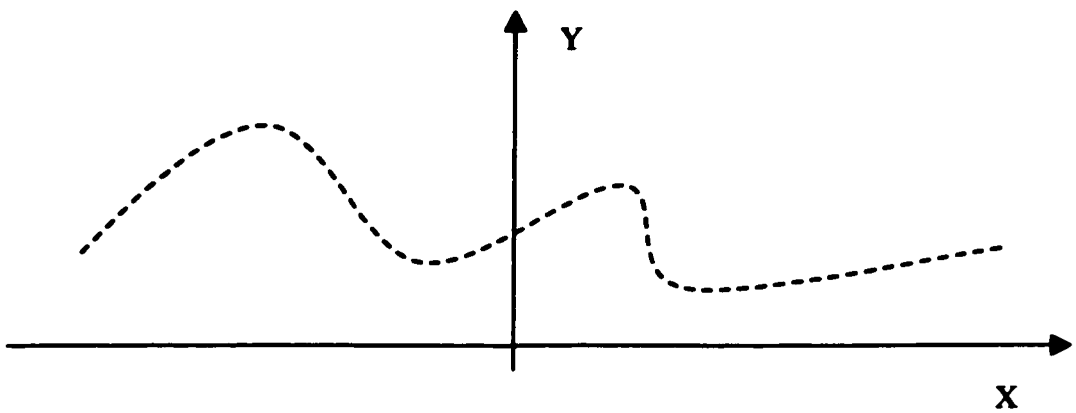
4.2 2D Registration

When registering two 2D range profiles, only three registration parameters need to be estimated: one rotation $d\varphi$ and two translations dx and dy. The input 2D range profiles are segmented into some linear sections and a modified Gauss circle is built for each profile to provide a compact representation of the shape of the profile. This section describes the process of registration using a specific example.

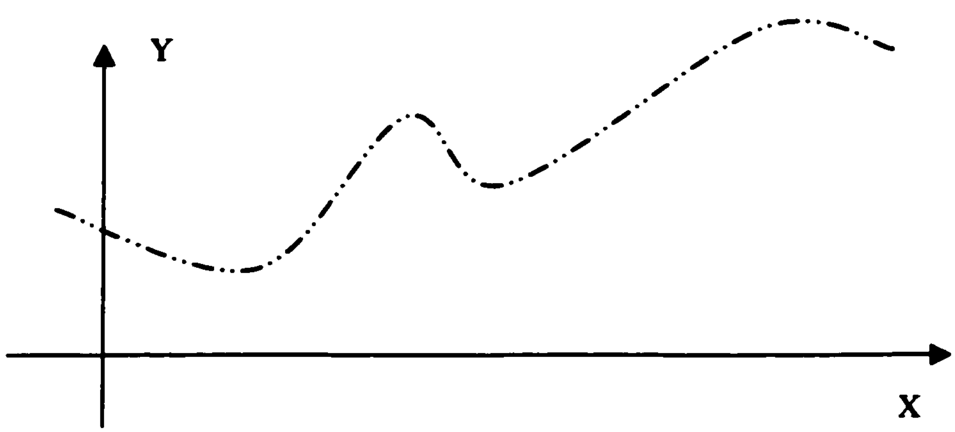
4.2.1 Segmentation and encoding

Figure 4.4 (a) and (b) show two simulated range profiles of the same object surface from two different points of view. In the simplest case, computation loops can be used to estimate the registration parameters $d\varphi$, dx and dy. With such a solution, all possible combinations of $d\varphi$, dx and dy contained within their respective ranges and based on the required precisions are tested. Only the combination that minimizes a given cost function is selected as the solution. An algorithmic description of such computation loops is presented below:

```
dφ = φ_minimum;
for dφ < φ_maximum
    dx = x_minimum;
    for dx < x_maximum
        dy = y_minimum;
        for dy < y_maximum
         { Compute some cost function, compare it to the previous cost,
           and update with the smallest one}
        dy = dy + y_step;
        end
    dx = dx + x_step;
    end
dφ = dφ + φ_step;
end
```



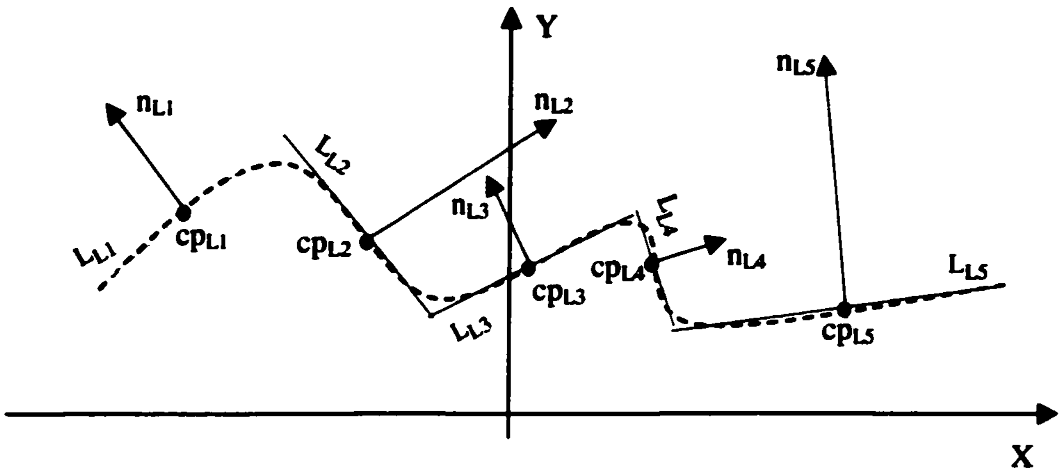
(a) Range profile from the left viewpoint.



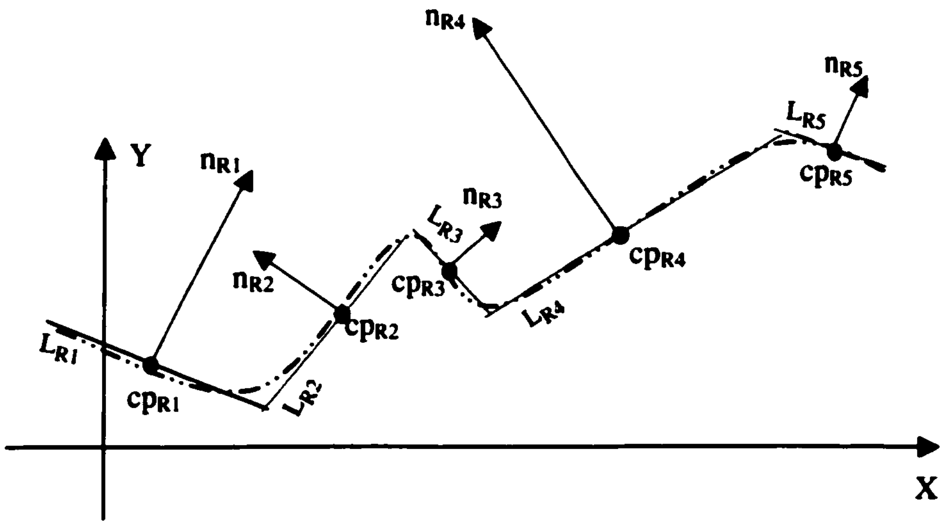
(b) Range profile from the right viewpoint.

Figure 4.4 Two range profiles collected on the same surface from different viewpoints.

As a result, when the ranges of all possible values that $d\varphi$, dx and dy can take are large ($\varphi_{maximum}$ - $\varphi_{minimum}$, $x_{maximum} - x_{minimum}$, and $y_{maximum}$ - $y_{minimum}$ are large), and a good precision is required (φ_{step}, x_{step} and y_{step} are small), the computation time increases as a power of 3, and the algorithm becomes intractable. Fortunately, this problem can be simplified. As illustrated in figure 4.5 (a) and (b), range profiles can be approximated by a set of line segments by means of some simple processing. Once each profile has been segmented, the normal vector and the coordinates of the centroid point are extracted for each line segment. The length of each line segment defines the length of its corresponding normal vector. A range profile of 256 measurements can therefore be encoded as a set of 2N parameters (N centroid points plus N normal vectors), where N corresponds to the number of straight-line segments associated with the given profile (N = 5 for both range profiles in this example). Then a polar map of each range profile can be obtained by representing these normal vectors all together with their extremity located on the origin of the map coordinate system as shown in figure 4.6 (a) and (b) respectively. As mentioned in the previous section, this polar map is called a modified Gauss circle in this thesis. This simplified notation allows both the compactness of the representation and an approximation of the surface structure along a single line in 3D space. It also facilitates the merge of range profiles to obtain the planar patches that are to be used when registering 3D range profiles.

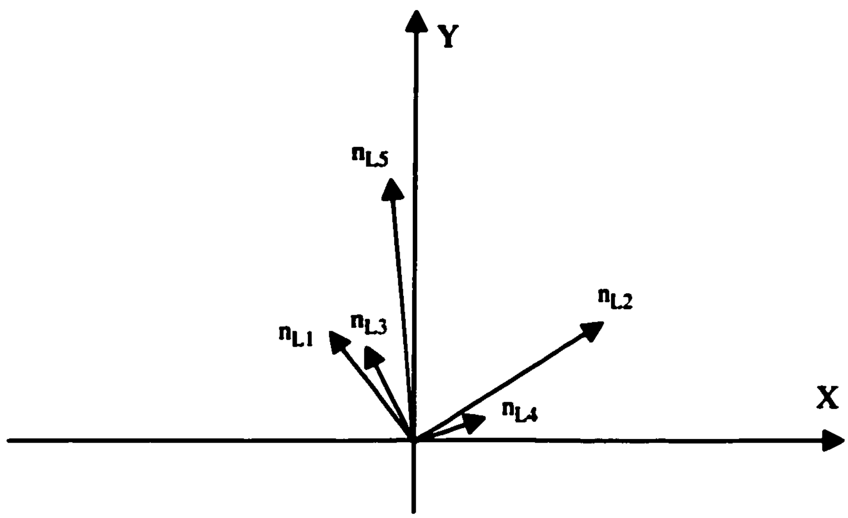


(a) Line segmentation for the left range profile.

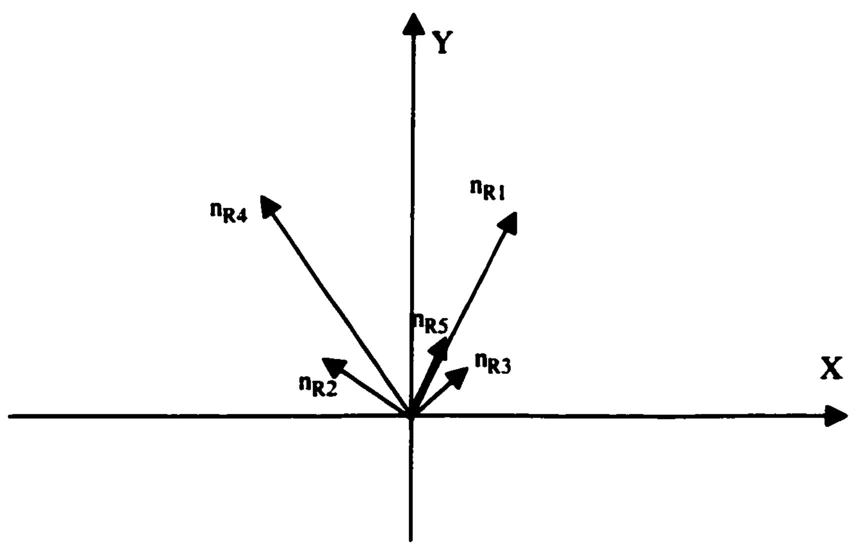


(b) Line segmentation for the right range profile.

Figure 4.5 Line segment approximation of the range profiles.



(a) Left range profile.



(b) Right range profile.

Figure 4.6 Polar representation of the range profiles (modified Gauss circles).

4.2.2 Registration estimation

Using the polar representation for each range profile, the rotation parameter $d\varphi$ between successive viewpoints can be directly estimated from the sets of normal vectors. Next the translation parameters can be computed from the centroid points coordinates, cp_i, while taking into account the rotation value previously obtained.

Under the rigidity constraint, these two Gauss circles in figure 4.6 (a) and (b) possess a similar normal vector distribution except for some rotation difference that results from the change in the viewpoint. If we rotate all vectors in (b) by a $d\varphi$ angle, most of them will overlap with the vectors in (a). This is illustrated in figure 4.7. Two vectors are considered to overlap when they have a similar orientation (within some tolerance level). The amount of overlap of the normal vectors depends on the overlap of two associated range profiles representing the object surfaces. Say that n_{R2} overlaps n_{L3}, n_{R4} overlaps n_{L5}, n_{R1} overlaps n_{L2}, and n_{R3} overlaps n_{L4} when a rotation $d\varphi$ is applied to the polar map of the right range profile. Obviously, the required angle $d\varphi$ corresponds to the rotation parameter we need to estimate. As a result, $d\varphi$ is obtained independently from dx and dy, thus providing a higher computational efficiency in comparison with the computational loops previously considered.

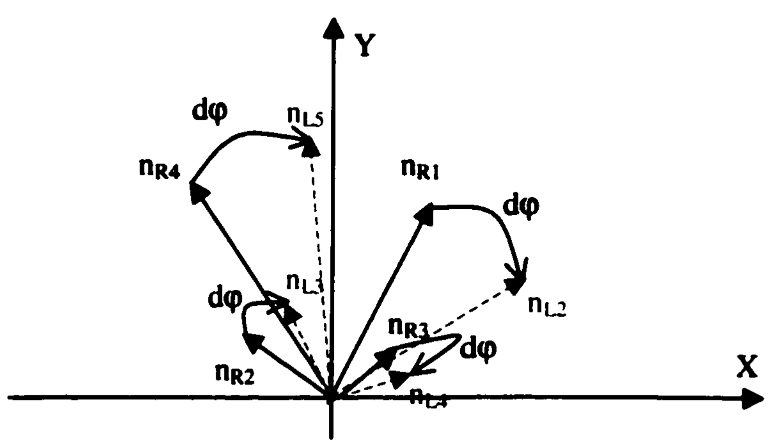


Figure 4.7 Rotation estimation from two modified Gauss circles.

Once the rotation is estimated, the application of the rotation parameter to the right range profile leads to two range profiles having a similar orientation with one being shifted in 2D space. Therefore, the translation parameters *dx* and *dy* can be estimated as the translations along the reference axes between the centroid points associated with each line segment, as shown in figure 4.8. If we translate all centroid points from the right scan profile (cp_{R1} ~ cp_{R5}) by *dx* along the X axis and by *dy* along the Y axis, then cp_{R1} overlaps cp_{L2}, cp_{R2} overlaps cp_{L3}, cp_{R3} overlaps cp_{L4}, and cp_{R4} overlaps cp_{L5} within some tolerance level. At this time, their associated line segments overlap each other too, i.e. L_{R1} overlaps L_{L2}, L_{R2} overlaps L_{L3}, L_{R3} overlaps L_{L4}, and L_{R4} overlaps L_{L5}. The total overlap is $L_{R1} + L_{R2} + L_{R3} + L_{R4}$, which is the maximum as well as the best overlap between left and right range profiles. As a consequence, the translation parameters *dx* and *dy* are determined. As we can anticipate, the estimates of the coordinates of centroid points maybe slightly deviated under the effect of noise, which would lead to a deviation of dx and dy estimates. Therefore, averaging and weighting operations are applied to compensate the problems. Details are presented in chapter 5.

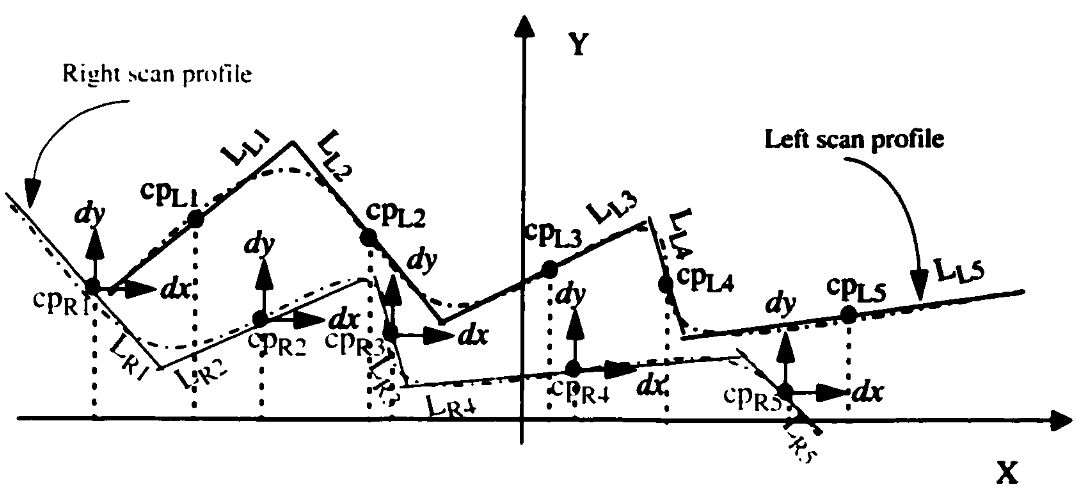


Figure 4.8 Translation estimation from the centroid points of line segments.

Figure 4.9 shows a superposition of the two original scan profiles shown in figure 4.4 (a) and (b) after applying the three transformation parameters (*d*φ, *dx* and *dy*) obtained through the above process. The efficiency of the algorithm is verified in theory. The validation on real applications will be presented in chapter 6.

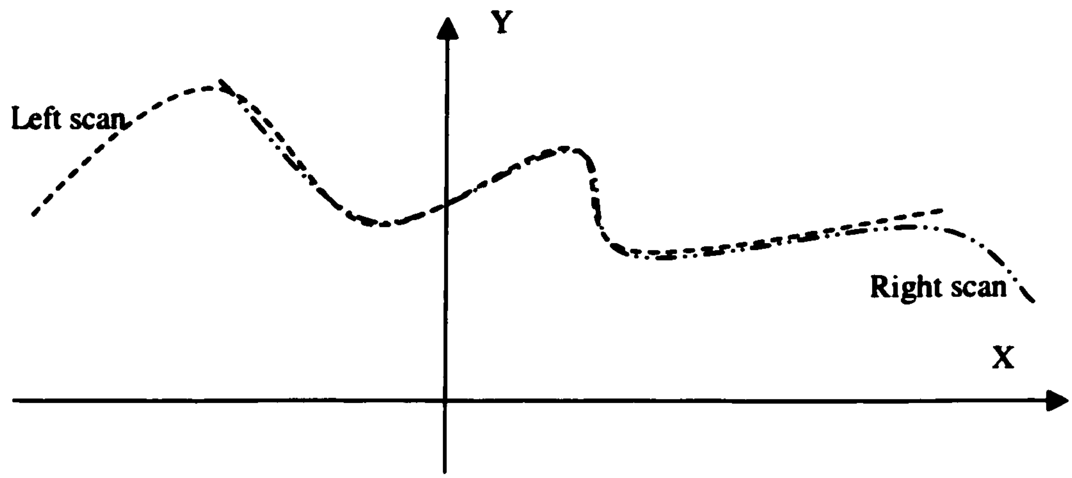


Figure 4.9 Superposition of left and right range profiles after transformation.

The registration parameters (*d*φ and *dx*) can be computed by the loop method discussed previously, i.e., scanning all the possible values in their respective ranges and selecting those that maximize respectively the overlap of the normal vectors in the polar maps and the overlap of line segments generated from the centroid points. Ideally if the same line segments are obtained on the overlapping part between the two range profiles, then the *y* coordinate difference (Δy) for each correspondence of centroid points is the same, and this is the expected translation parameter along the Y axis (*dy*). But in practice, some deviations occur as a consequence of measurement errors and line segmentation approximations. However, the final *dy* can be refined by averaging the *y* coordinate difference of all correspondences of centroid points as it tends to reduce the impact of the noise and increase robustness. The revisited procedure to obtain the complete set of parameters can be presented as follows:

```
Loop
   Obtain rotation dφ
End
Apply dφ
Loop
   Obtain translation dx
End
Apply dφ and dx
   Obtain translation dy
```

With this improved approach, each of the $d\varphi$, dx and dy parameters is obtained separately even though the latter are based on previous ones. No recursive loop is required, which makes the approach much more efficient than the initial strategy.

This process can still be improved to compute $d\varphi$, dx and dy in a more efficient way. Instead of performing loop computation, an exhaustive search can be applied to find all possible correspondences of normal vectors. For each correspondence, $d\varphi$ is then estimated as the angular difference between two corresponding normal vectors as shown in figure 4.7. dx and dy are computed as the shifts in 2D space between corresponding centroid points after the application of a $d\varphi$ rotation on one polar map. The resulting $d\varphi$, dx and dy are then applied to all other line segments, and the overall overlap of line segments is computed as an evaluation factor. Those $d\varphi$, dx and dy values that lead to a maximum overlap are selected as the best estimates for the registration parameters. Given one pair of corresponding normal vectors and the associated centroid points, the closed-form solution to compute the transformation parameters is:

$$d\varphi = \operatorname{atan}\left(\frac{y_2}{x_2}\right) - \operatorname{atan}\left(\frac{y_1}{x_1}\right) \tag{4-1}$$

$$dx = x_{p1} - x_{p2} \cdot \cos d\varphi - y_{p2} \cdot \sin d\varphi \tag{4-2}$$

$$dy = y_{p1} + y_{p2} \cdot \cos d\varphi - y_{p2} \cdot \sin d\varphi \tag{4-3}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of corresponding vectors, and (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}) are the coordinates of associated centroid points. A detailed development of this closed-form solution can be found in Appendix C.

In practice, the number of normal vectors for each scan profile is very small compared to the number of raw data points, resulting from the line segmentation and the combination of line segments having a similar orientation. Therefore the exhaustive search for matching normal vectors is realistic and the computation time is reasonable. For example, with the 5 normal vectors describing each profile in the above example, only 25 (5 times 5) loops are required. The advantages of this solution will be fully revealed in solving the 3D registration problem.

4.2.3 Discussion

The main characteristic of the algorithm described above is that it relies on a straight-line segments approximation that leads to a compact range profile description, instead of directly using raw range measurements. The advantages of this approximation can be summarized as follows:

1.) It simplifies the representation of the data. Initially, up to 256 points are contained in the representation of one range profile. However, depending on the precision required a limited number of line segments are necessary to represent the same range profile. Therefore, a higher computational efficiency is achieved in comparison with a point representation.

2.) No complex cost function needs to be computed when matching two range profiles. A widely used cost function to evaluate the correctness of registration is:

$$E = \sum_{i=1}^{n1} (R \cdot P_i - Q_j)^2 \tag{4-4}$$

where, E is the overall square error between two matched profiles; R is the registration matrix; P_i is the point in one range profile; Q_j is the corresponding point to P_i in the other range profile.

For each estimate of R, the cost function has to be recomputed. However, in the proposed algorithm, instead of computing the difference (errors) between two matched profiles, we compute the overlap between the range profiles, which is equivalent to the overlap between the normal vectors in the polar maps because the length of normal vector stands for the length of its associated line segment. And this can be obtained by simple additions:

$$TotalOverlap = \sum_{i=1}^{n} len_i \tag{4-5}$$

where len_i is the overlap of the i^{th} pair of normal vectors with similar orientation (within some tolerance level). The overlap values, len_is, are computed only once at the beginning as the smallest one between one pair of normal vectors.

3.) The difficulties originating from measurement errors and varying resolutions can be reduced by the straight-line segmentation process. For example, considering some raw range points in figure 4.10, a good estimate of the actual object surface is provided by the approximate line segments. Moreover, the computation of a cost function based on the raw data points would result in a large error when matching the profiles in (a) and (b) together, even though the corresponding surfaces would be properly aligned. As this error tends to grow when the number of data points increases, it would be difficult to define a threshold value based on such a cost function. Moreover, assuming that the object surfaces in figure 4.10 (a) and (b) are the same, it is shown that p_1 in (b) cannot find its corresponding point in (a) because of the varying distances between two successive measurement points. But if the approximate line segments representation is used, p_1 can be approximated by p_1', then a point q_1 located on the corresponding line segment can be matched with it. In general, registration based on point representation appears to be more sensitive to measurement errors and resolution than registration based on line segments. The same phenomenon is observed for triangular mesh representation [15, 36].

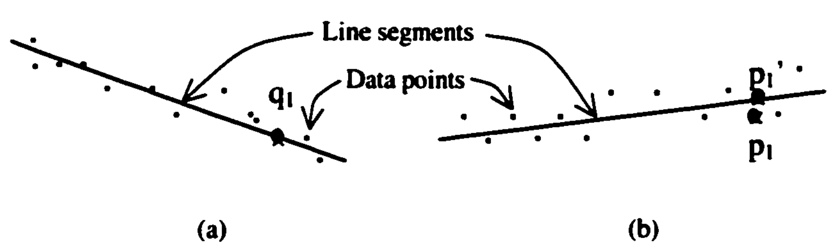
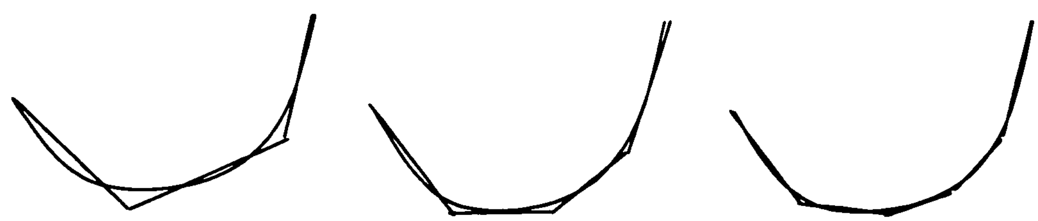


Figure 4.10 Straight-line segments approximation for some raw points.

4.) No sophisticated features on the object surface are required. In feature-based approaches, complex features, such as edges, corners and contours, need to be extracted. This makes these techniques highly sensitive to the representation as these features are difficult to detect and to locate precisely given the current state of computer vision technologies. The proposed approach rather makes use of simple straight lines or flat surfaces which exist on most object surfaces. Even round objects can still be approximated by a set of small straight lines or planar patches as shown in figure 4.11. The number of line segments is adjusted with the required precision to improve the approximation. This makes the proposed approach very robust.



(a) low precision (3 segments) (b) medium precision (4 segments)(c) high precision (5 segments)

Figure 4.11 An example of straight-line segments approximation with various precisions.

5.) Considering the overlap of the line segments as the evaluation factor rather than the number of matched points between the two sets of data [7, 13, 14, 15] speeds up the computation. It also gives more weight to those points that belong to larger straight segments. This strategy seems to be appropriate as large approximated segments based on more data points appear to be a more reliable representation of the object surface for a given required accuracy.

6.) Globally optimized transformations tend to be found, instead of locally optimized transformations. This benefits from the simplicity of the computation of overlap which is used as the evaluation factor. Those segments with similar orientation are grouped together without caring about the geometrical connectivity between them. As a result, the computation of overlap between normal vectors is based on the object's global shape rather than on local features.

7.) Each registration parameter is obtained separately. This significantly reduces computation time.

8.) Finally, this approach can be easily extended to 3D registration, thus providing six DOFs registration [$d\theta$, $d\varphi$, $d\psi$, dx, dy, dz]. For this purpose, straight-line segments are simply replaced by planar patches. In this case, normal vectors are referred to the orientation of the patches in 3D space. Their length reflects the area of the patches, and the modified Gauss circle is extended to a modified Gauss sphere.

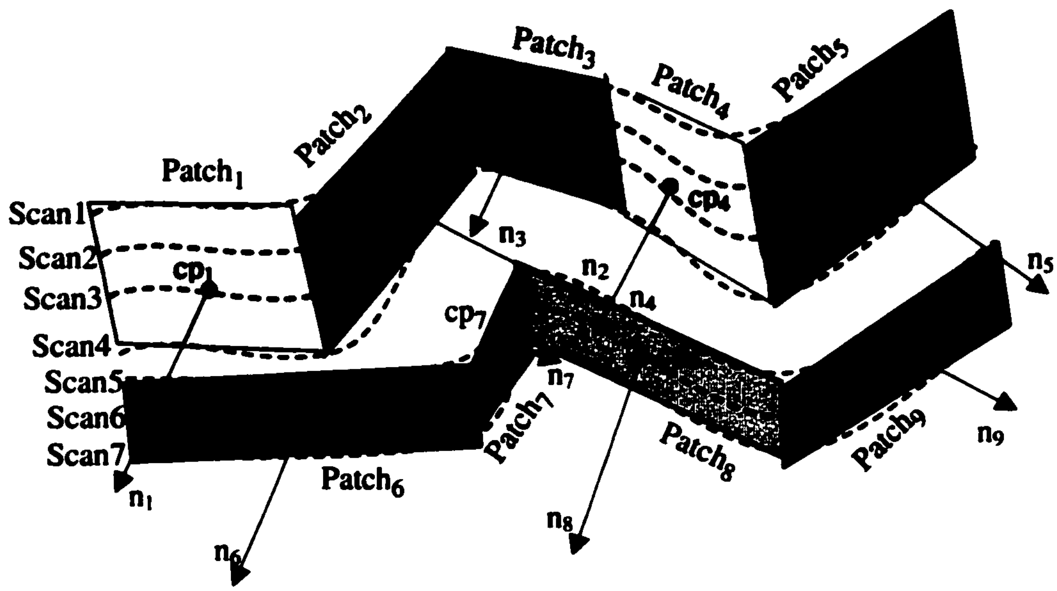
4.3 3D registration

As stated in the previous section, the technique introduced for 2D registration estimation can be extended to 3D registration. The main difference is that planar patches are used instead of straight-line segments. As a consequence, one supplementary step needs to be added at the beginning of the process to merge the 2D range profiles and get planar patches that approximate object surfaces. The following sections mainly describe these steps that differ from 2D registration introduced previously.

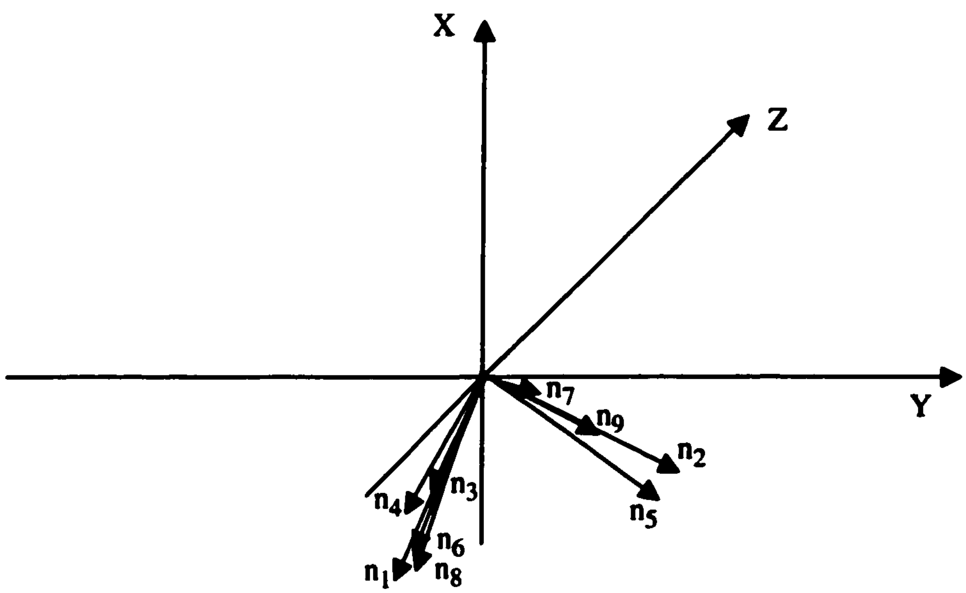
4.3.1 Patch segmentation and encoding

The straight-line segmentation previously obtained for 2D registration provides an efficient way to merge similar neighboring range profiles and to create a simple patch-based surface representation. As we know from the experimental setup described in chapter 3, the raw data provided from the Jupiter range finder consist of a set of 2D scan profiles collected along a straight line followed by the sensor. As the displacement between these two successive profiles is small (5 mm or 3 mm in our experiments), the variation of the shape between two successive profiles should be smooth where the object surface is continuous. As a consequence, these profiles should have a similar normal vector distribution and they can be merged together to create patches. On the other hand, transitions between objects appear as abrupt changes between successive profiles. When such a transition occurs, the respective profiles are assigned to different neighbor patches as shown in figure 4.12 (a). The angle and the length differences between normal vectors are used as similarity evaluation parameters between two candidate range profiles. The threshold values used in our experiments are 0.025 rad for angles and 25% in length.

Considering the example of figure 4.12, scans 1 to 4 have been collected on a continuous surface and can therefore be merged. As these profiles have been previously segmented, a different surface patch is created for each scan segment (patches 1 to 5 inclusively). The same idea applies to scans 5, 6 and 7 as they are similar to eachother. However, there is an important transition between scans 4 and 5 which are successive profiles in the raw data set. Locating this transition allows one to delimit the boundaries between patches associated with the first and the second group of scan lines respectively. The details of implementation on how to fit the patches from grouped scan profiles will be discussed in chapter 5. Once the planar patches are obtained, three patch parameters are computed: the normal vector, n_i, the centroid, cp_i, and the area, s_i, which defines the length of n_i.



(a) Seven scan profiles from one point of view.



(b) Modified Gauss sphere.

Figure 4.12 3D registration principles.

In a way similar to what is done for 2D registration, bringing the extremities of all vectors to the same reference point provides the modified Gauss sphere representing the object surface as shown in figure 4.12 (b). Under rigid transformation constraints, processing the other set of data collected from the second viewpoint produces another modified Gauss sphere, with a similar normal vector distribution but with some rotation and translation differences for the overlapping parts in the scanned scenes.

4.3.2 Registration estimation

Provided with these two modified Gauss sphere representations of surfaces, the registration parameters are computed in the same way as for 2D registration. An exhaustive search for correspondences of normal vectors between the two Gauss spheres is performed and a rotation matrix is computed for each possible set of correspondences. The rotation matrix that maximizes the overlap of the two sets of normal vectors is considered as the best estimate. Theoretically, three correspondences of normal vectors are required to uniquely determine the rotation matrix, from which three rotation parameters (θ, φ, ψ) can be computed [44].

In our experiments, we use the evolving reference frame-based TRPY transformation convention (translations along the X, Y and Z axes (T_x, T_y, T_z) $\rightarrow$ rotation around the Z axis ($\theta|_{Roll}$) $\rightarrow$ rotation around the Y axis ($\phi|_{Pitch}$) $\rightarrow$ rotation around the X axis ($\psi|_{Yaw}$)). The corresponding homogeneous transformation matrix is defined as follows:

$$Q = T \cdot R_{Rotz} \cdot R_{Roty} \cdot R_{Rotx} \tag{4-6}$$

$$Q = \begin{bmatrix} \cos\theta \cdot \cos\varphi & \cos\theta \cdot \sin\varphi \cdot \sin\psi - \sin\theta \cdot \cos\psi \\ \sin\theta \cdot \cos\varphi & \sin\theta \cdot \sin\varphi \cdot \sin\psi + \cos\theta \cdot \cos\psi \\ -\sin\varphi & \cos\varphi \cdot \sin\psi \\ 0 & 0 \end{bmatrix.$$

$$\begin{matrix} \cos\theta \cdot \sin\varphi \cdot \cos\psi + \sin\theta \cdot \sin\psi & X_T \\ \sin\theta \cdot \sin\varphi \cdot \cos\psi - \cos\theta \cdot \sin\psi & Y_T \\ \cos\varphi \cdot \cos\psi & Z_T \\ 0 & 1 \end{matrix}\right] \tag{4-7}$$

$$Q = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{4-8}$$

where, R is a 3 by 3 rotation matrix; T is a 3 by 1 translation matrix, $T = [T_x, T_y, T_z]^T$; ψ, φ and θ are three rotation parameters around the X, Y, and Z axes respectively; T_x, T_y and T_z are three translation parameters along the X, Y, and Z axes respectively.

Provided that n_1 and n_1', n_2 and n_2', n_3 and n_3' are three non-degenerated sets of corresponding normal vectors respectively extracted from the two sets of raw data associated with two viewpoints between which a rotation, R, exists, these vectors must satisfy the following constraint equations:

$$n_1 = R \cdot n_1' \tag{4-9}$$

$$n_2 = R \cdot n_2' \tag{4-10}$$

$$n_3 = R \cdot n_3' \tag{4-11}$$

If we let:

$$n_1 = [x_1, y_1, z_1]^T \qquad \text{and} \qquad n_1' = [a_1, b_1, c_1]^T$$

$$n_2 = [x_2, y_2, z_2]^T \qquad \text{and} \qquad n_2' = [a_2, b_2, c_2]^T$$

$$n_3 = [x_3, y_3, z_3]^T \qquad \text{and} \qquad n_3' = [a_3, b_3, c_3]^T$$

and

$$R = \begin{bmatrix} g & h & i \\ j & k & l \\ m & n & p \end{bmatrix}$$

Then we have:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & b_1 & c_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_2 & b_2 & c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_3 & b_3 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_3 & b_3 & c_3 \end{bmatrix} \cdot \begin{bmatrix} g \\ h \\ i \\ j \\ k \\ l \\ m \\ n \\ p \end{bmatrix} \tag{4-12}$$

which corresponds to:

$$B = A \cdot X \tag{4-13}$$

and can be solved with:

$$X = A^{-1} \cdot B \tag{4-14}$$

Solving for $X = [g, h, i, j, k, l, m, n, p]^T$ provides the rotation matrix parameters. The three rotation parameters are then extracted as follows:

$$\psi = \operatorname{atan}\left(-\frac{l}{p}\right) \qquad \text{rotation around the X axis} \quad (4\text{-}15)$$

$$\varphi = \operatorname{atan}\left(\frac{j \cdot \sin\psi - m \cdot \cos\psi}{g}\right) \qquad \text{rotation around the Y axis} \quad (4\text{-}16)$$

$$\theta = \operatorname{atan}\left(-\frac{h}{g}\right) \qquad \text{rotation around the Z axis} \quad (4\text{-}17)$$

In a simplified context, if we assume that there is no rotation around the Z axis (i.e. $\theta ==$ 0), then the rotation matrix, R, becomes:

$$R = \begin{bmatrix} \cos\varphi & \sin\varphi \cdot \sin\psi & \sin\varphi \cdot \cos\psi \\ 0 & \cos\psi & -\sin\psi \\ -\sin\varphi & \cos\varphi \cdot \sin\psi & \cos\varphi \cdot \cos\psi \end{bmatrix} \qquad (4\text{-}18)$$

In this case, only one correspondence between two normal vectors is required to compute the two remaining rotation parameters (φ and ψ). If n = $[x, y, z]^T$ corresponds to n' = $[a, b, c]^T$, then they must satisfy:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \cdot \sin\psi & \sin\varphi \cdot \cos\psi \\ 0 & \cos\psi & -\sin\psi \\ -\sin\varphi & \cos\varphi \cdot \sin\psi & \cos\varphi \cdot \cos\psi \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad (4\text{-}19)$$

With this simplified constraint equation, θ, φ, and ψ can be computed by the following equations (please see Appendix C for detailed explanations on how to obtain these equations):

$$\theta = 0 \tag{4-20}$$

$$\varphi = \operatorname{asin}\frac{a}{\sqrt{x^2+z^2}} + \operatorname{atan}\left(\frac{x}{z}\right) \tag{4-21}$$

$$\psi = \operatorname{asin}\frac{y}{\sqrt{b^2+c^2}} + \operatorname{atan}\left(\frac{b}{c}\right) \tag{4-22}$$

As with the 2D registration problem, the three translation parameters, $T = [T_x, T_y, T_z]$, can be estimated from the required translations along the reference axes between the centroid points associated with each surface patch. The correspondences of centroid points are achieved by examining the correspondences between normal vectors associated with the same patches. This can be shown mathematically as follows: if a normal vector, n, and a centroid point, cp, refer to one patch while n' and cp' refer to another patch, and n matches n' when the rotation matrix R is applied on the second patch, then we have::

$$\begin{bmatrix} cp \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cp' \\ 1 \end{bmatrix} \tag{4-23}$$

$$cp = R \cdot cp' + T \tag{4-24}$$

$$T = cp - R \cdot cp' \tag{4-25}$$

As many surface patches need to be matched, a good estimate of the translation parameters is the average of the necessary displacement along each axis over all matched centroids, that is:

$$T = \frac{1}{N} \cdot \sum_{i=1}^{N} (cp_i - R \cdot cp_i') \tag{4-26}$$

where N is the number of correspondences between centroid points.

In practice, the percentage of overlapping areas of corresponding patches is taken into account to compute the translation. With this strategy, better results can be achieved as more weight is put on those correspondences that have a larger percent of overlap. The previous equation is then modified to:

$$T = \frac{\sum_{i=1}^{N} (cp_i - R \cdot cp_i') \cdot s_i}{\sum_{i=1}^{N} s_i} \tag{4-27}$$

where s_i is the percentage of overlapping area of the i^{th} correspondence of patches. It is computed as follows. Provided that $patch_1$ and $patch_2$ are two corresponding patches with areas a_1 and a_2 respectively, and a_1 is larger than a_2, then we have:

$$s = \frac{a_2}{a_1} \tag{4-28}$$

In summary, the proposed approach extended to 3D space allows the computation of the full set of six registration parameters (3 rotations and 3 translations) following the procedure described previously. In addition to the characteristics mentioned for the 2D registration problem, the line segments approximation now contributes to make the merge of range profiles efficient. Planar patches of large dimension can be computed efficiently. The proposed planar patches representation is advantageous in comparison to triangular mesh representation as it is easy to implement and makes a direct use of single line of raw data sets. It also uses more raw points for fitting each patch and is less sensitive to measurement errors and varying resolutions.

After this theoretical description of the proposed algorithm, chapter 5 will examine the particularities of the implementation on real data.

Chapter 5 Implementation

5.1 Introduction

Following the theoretical description presented in the previous chapter, some practical aspects of the implementation of the algorithm are now examined. For the proposed registration mechanism, the steps of data processing are summarized in figure 5.1.

5.2 Preprocessing

In practice, two preprocessing steps are executed on the range profile before segmentation. That is outlier removal and profile smoothing. Outlier removal consists of eliminating some isolated points that cannot be realistically considered as being part of a surface as they are separated from their neighbors by a large distance. This eliminates large deviations that often occur in raw measurements. A 1 by 5 size median filter is also applied on the range profile to reduce the effect of Gaussian-like noise. These two steps are important because they influence the accuracy of the line segmentation and, in turn, the final registration estimation. Figure 5.2 shows one scan profile before and after preprocessing. The outliers are removed and the noisy area is smoothed after preprocessing. This results in a better representation of the object surfaces.

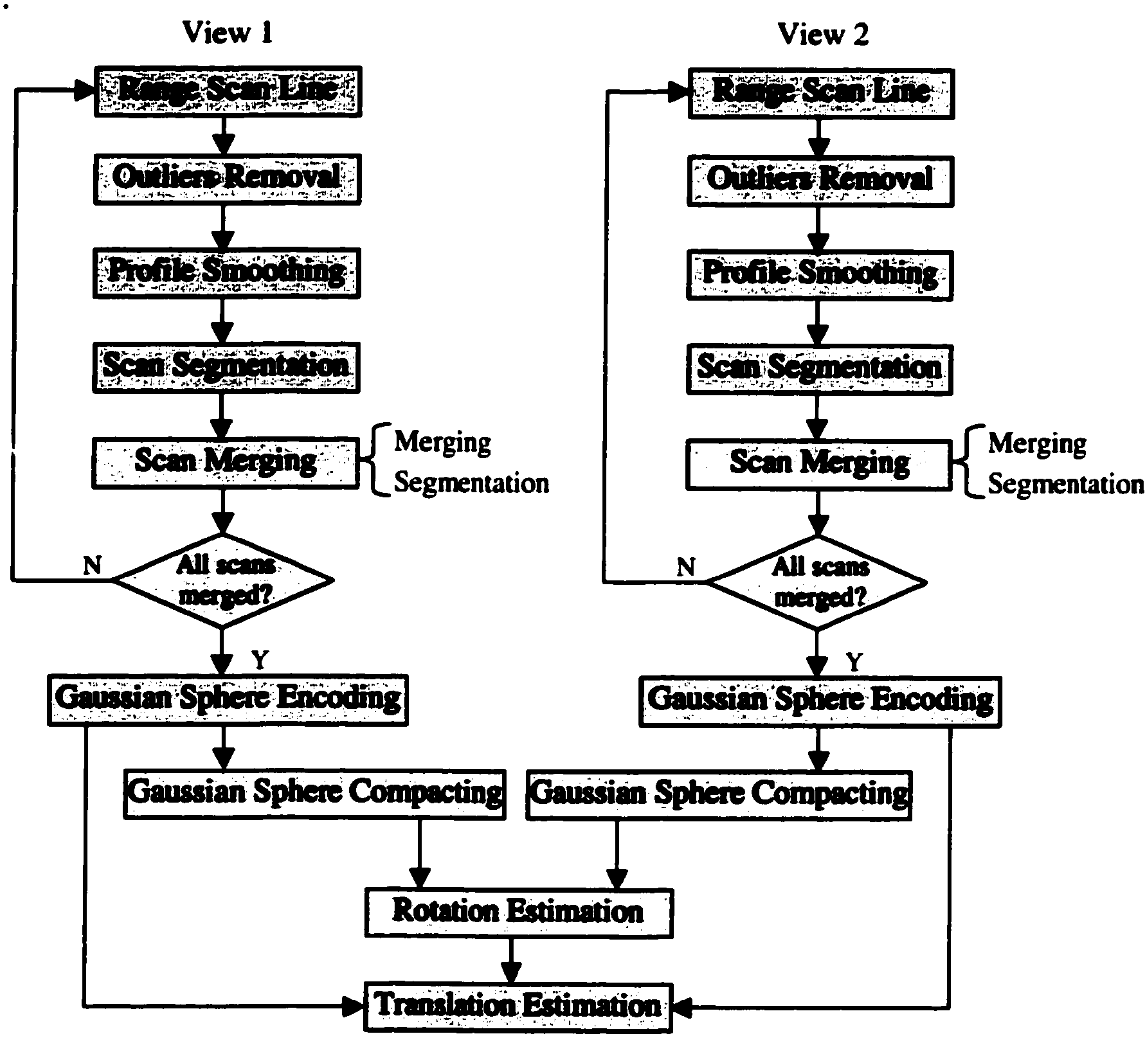
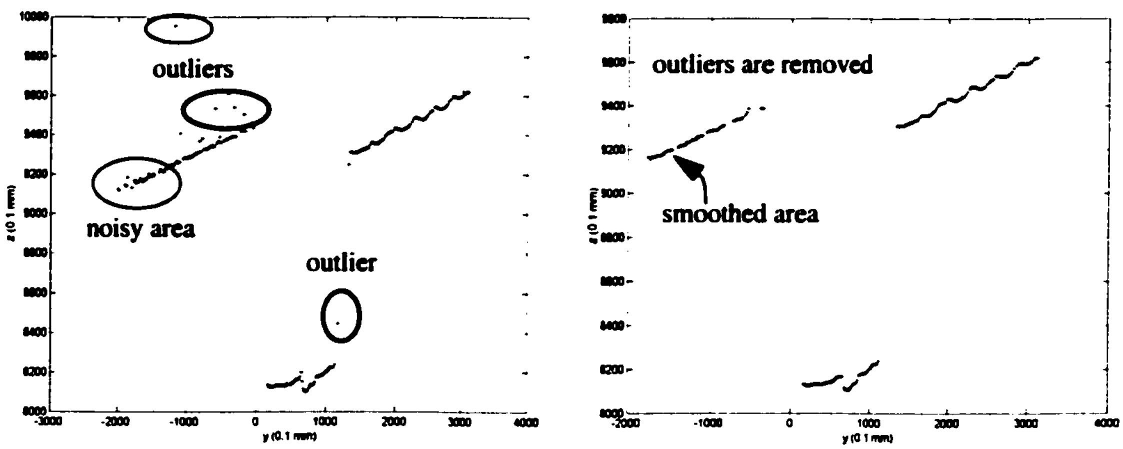


Figure 5.1 Steps of data processing for registration estimation.



(a) Before preprocessing. (b) After preprocessing.

Figure 5.2 Preprocessing of a raw range profile.

5.3 Scan segmentation

In the previous chapter, we discussed straight-line segments approximation for each range profile. Here are the details of the implementation. As shown in figure 5.3, the process starts at one extremity of the range profile. The first three measurements are initially used to obtain a coarse estimate of the orientation of the straight line. The following points encountered along the profile are then successively checked for their proximity in terms of the distance with respect to the initial line estimate. When the deviation between a measurement point and the straight-line estimate is below a given threshold (4 mm in our experiments), then the point is considered to be part of this straight-line segment. A segment is terminated when the following measurement point is located too far away from the straight-line estimate. In this case, a new straight-line segment is initialized and the remaining points in the range profile are successively checked for their proximity with this new segment. The process is repeated until the end of the profile is reached. Once a segment is terminated, it is revisited in order to refine the estimates of its parameters. The parameters of each straight-line segment are then computed on the basis of the entire set of measurement points that are associated with it rather than only from the estimate of the first three points. The process of straight-line fitting on a large number of raw data is discussed in appendix D.

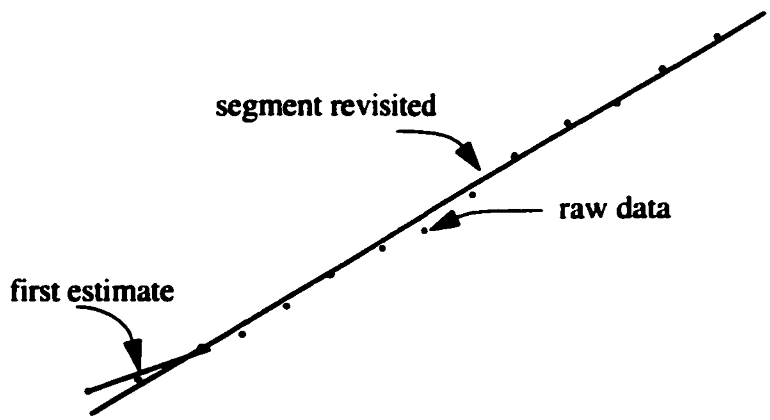


Figure 5.3 Straight-line fitting on a sequence of range measurements.

5.4 Patch segmentation

Provided segmented input profiles have been grouped based on their similarity as discussed in section 4.3.1, planar patches can then be defined for each group of merged straight-line segments. Three points from each segment (one in the middle and two respectively at ¼ and ¾ of the line segment) are considered to estimate the patch orientation. Figure 5.4 illustrates the process of patch definition for the first patch. p_1, p_2 and p_3 are selected from the first segment in scan 1, while p_4, p_5 and p_6 are extracted from the corresponding segment in the second profile. Knowing the distance between two successive profiles, which is determined by the increment of Jupiter sensor movement when gathering the measurements, the X and Y coordinates of p_4 can be estimated. The corresponding Z coordinate of p_4 is then extracted from the second range profile. The same logic applies to estimate the coordinates of p_5 and p_6. Eventually the process is repeated to locate p_7 to p_9 from p_4 to p_6 and so on for as many scans as necessary to fully define a given planar patch.

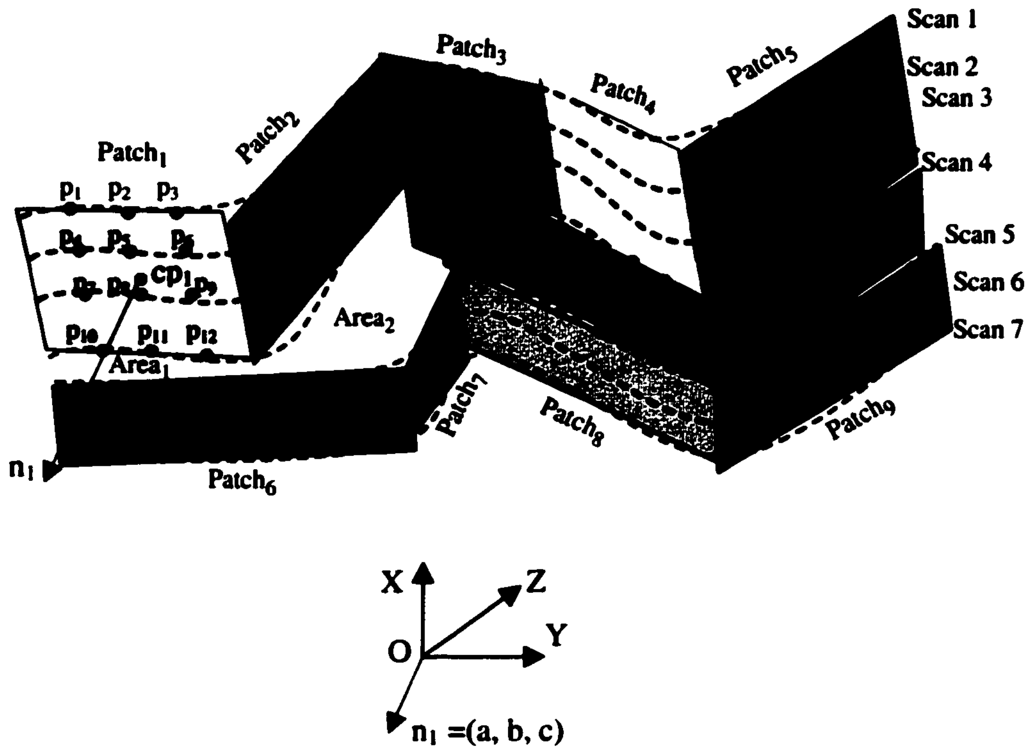


Figure 5.4 Definition of patches as merged segmented scans.

Our experiments revealed that some Z estimates might have a large deviation with respect to the previous range profile. A special treatment has then been developed to correct this situation. The correction is based on the parallel property of the rectangular shape of the patches, i.e., the deviations of Z value of the three points on one line segment compared to their corresponding points in the previous range profile should be similar. If the deviation of the Z estimate of two out of three points in one line segment are larger than a threshold, we replace the three Z estimates with the Z values of their corresponding points in the previous range profile if there are only two profiles that are grouped together, otherwise the Z estimates are established by interpolation from their corresponding points in the associated group of range profiles, based on the assumption that the object surfaces are continuous within the area of one group of range profiles.

When the 3D coordinates of all triplets are known for a given patch, the best fitting planar surface parameters are computed following these steps:

Let the plane equation be:

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \tag{5-1}$$

where a, b, c and d are optimized when E is minimized:

$$E = \sum_{i=1}^{N} (a \cdot x_i + b \cdot y_i + c \cdot z_i + d)^2 \tag{5-2}$$

where $p_i = (x_i, y_i, z_i)$ are the point coordinates obtained above for a given planar patch.

An equivalent way to implement this minimization process is to use the following equation:

$$E = ||\alpha \cdot V||^2 \tag{5-3}$$

where:

$$\alpha = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \dots & \dots & \dots & 1 \\ x_N & y_N & z_N & 1 \end{bmatrix}$$

and

$$V = [a, b, c, d]^T$$

Mathematically, E is minimized when V is a unit eigenvector of $\alpha^T\alpha$ associated with the smallest eigen value [16]. This way, a fitted plane, which minimizes the sum of the squared distance between the data points and the plane, is obtained. If we renormalized V, $[a, b, c]^T$ becomes the unit normal vector to the plane while d corresponds to the shortest distance between the plane and the origin of the coordinate system. The centroid point coordinates, cp, of the patch are defined as the mean of the coordinates of all triplet points belonging to the patch.

For the zones between groups of range profiles, intermediate patches might also be generated to connect the initial ones together. This is reasonable because surfaces still exist in between scanned surfaces even though there is a sharp transition on the objects surfaces. However, we only compute those intermediate patches when the neighboring line segments are parallel between the last range profile in the previous patch and the first range profile in the following patch, as shown for $patch_{10}$, $patch_{11}$ and $patch_{12}$ in figure 5.4. This strategy compensates for some of the lost of information that results from the use of rectangular patches. This adaptation also tends to compensate for imperfect subdivision of range profiles. The representation of other areas, like $area_1$ and $area_2$ in figure 5.4, is to be considered in future developments of this approach where an irregular patch representation methodology or other strategies will be examined to obtain a better surface representation.

5.5 Gauss sphere compactness

After the modified Gauss spheres have been built, we observed that this representation can still be advantageously compacted to a limited number of normal vectors without seriously degrading the surface representation. Specifically, those normal vectors which have a small angular difference in between them can be merged into a single vector and their respective lengths added. Merging these normal vectors is equivalent to the unification of the corresponding planar patches that have a very similar orientation (within some tolerance level) with no consideration of their respective position in space. Figure 5.5 shows the metric of conormality that we introduced to select normal vectors that can be merged. Assuming that $\vec{n_1}$ and $\vec{n_2}$ are two normal vectors corresponding to two patches, and p_1 and p_2 are their projection points on the unit sphere surface, $\vec{n_1}$ and $\vec{n_2}$ can then be merged if the distance between p_1 and p_2 is less than a given threshold (0.14 mm in our experiments). The resulting vector $\hat{n}$ has the total length of $\vec{n_1}$ and $\vec{n_2}$. In practice, the number of normal vectors is significantly reduced after this procedure. This opportunity to further reduce the complexity of the representation is provided by the fact that most objects are approximated by a large number of separated patches, many of them having a similar orientation.

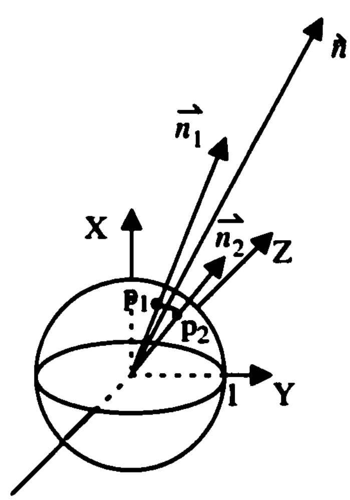


Figure 5.5 Conormality of normal vectors.

5.6 Rotation estimation

In chapter 4, we have derived the closed-form solution for rotation estimation based on three pairs of matched normal vectors. However, the implementation is somewhat more tricky than the theoretical description as the following questions must be addressed: How to choose three non-degenerate normal vectors from one Gauss sphere in an efficient and exhaustive way? How to find the corresponding vectors from another Gauss sphere? Moreover, the orthogonality of the resulting rotation matrix has to be verified before it is applied to compute translation parameters. The diagram of the module that we developed to estimate the rotation is illustrated in figure 5.6.

The angles between each pair of vectors within one compacted Gauss sphere are first computed and they are stored in two arrays: left_ang_diff() and right_ang_diff(). These two arrays are symmetric along the diagonal because the angle between vector i and vector j is the same as the angle between vector j and vector i. For a compacted Gauss sphere containing N vectors, the array becomes:

$$
\begin{array}{cccccccc|c}
 & & & & & & \text{column} & & \\
A_{11} & A_{12} & A_{13} & \cdots & \cdots & \cdots & \cdots & A_{1N} & \\
A_{21} & A_{22} & A_{23} & \cdots & \cdots & \cdots & \cdots & A_{2N} & \\
\cdots & \cdots & \cdots & \cdots & \cdots & A_{kj} & A_{ki} & \cdots & \text{row} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{ji} & \cdots & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\
A_{N1} & A_{N2} & A_{N3} & \cdots & \cdots & \cdots & \cdots & A_{NN} &
\end{array}
$$

where $A_{ij} = A_{ji}$, $i, j \in (1, 2, 3, \ldots, N)$.

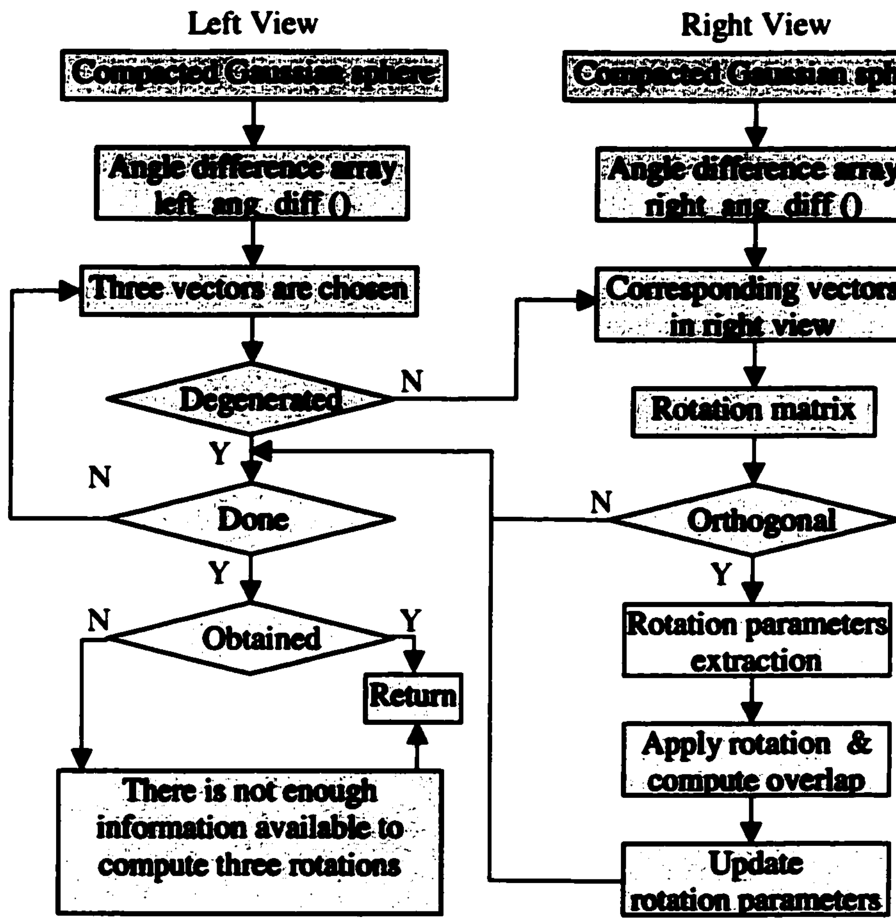


Figure 5.6 Rotation estimation.

When we choose three vectors from the left compacted Gauss sphere, we actually choose three angles from the upper triangle of the left_ang_diff() array which are in one row and one column. In other words, one of the three angles has to be at the intersection of the row and column. This results from the fact that the three angles are generated from three independent vectors, say i, j, k. Provided that i is larger than j which is larger than k, then the locations of the three angles generated from vectors i, j and k in the array are (k, j), (k, i) and (j, i). We do not choose the other three locations that are (j, k), (i, k) and (i, j) because they are in the lower triangle of the array that is symmetric to the upper triangle. The three locations chosen then form one row and one column with one location being at their intersection in the array as shown in the array representation above. The degeneration of the three vectors corresponding to the three chosen angles is then checked. If one of the angle is equal (within some tolerance level) to the sum of the other two, then these three vectors are discarded and the next available group of three vectors is selected and tested.

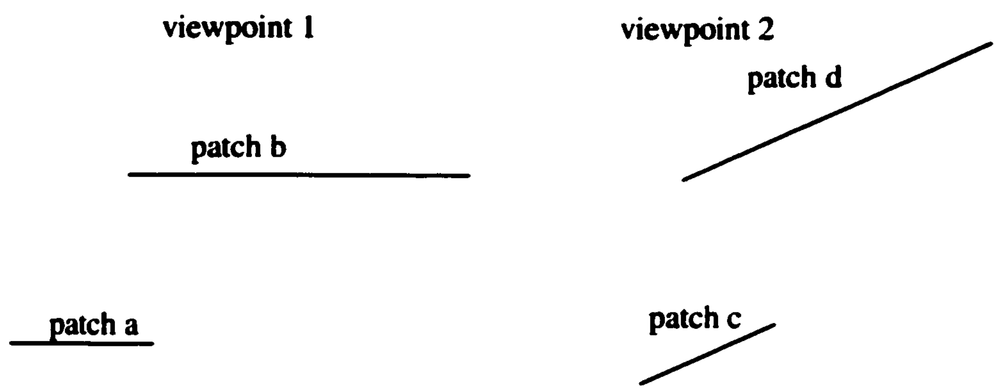
Once three non-degenerate vectors have been found in the left compacted Gauss sphere, three corresponding vectors in the right compacted Gauss sphere are found by searching three angles in the right_ang_diff() array that are equivalent to those selected in left_ang_diff(). They must only consist of one row and one column as well. Then equations from (4-9) to (4-17) inclusively are applied using these vectors to compute the rotation matrix. Finally, the resulting rotation matrix is checked for orthogonality in order to eliminate solutions without physical meaning. After the application of the rotation matrix, the overlap of normal vectors between the right and the left compacted Gauss spheres can be computed. The rotation estimate is then updated with the solution that maximizes the overlap. If all possibilities of three matching vectors are visited and no rotation estimate can be made, we conclude that three rotations (around the X, Y and Z axes) cannot be found based on the information available.

5.7 Translation estimation

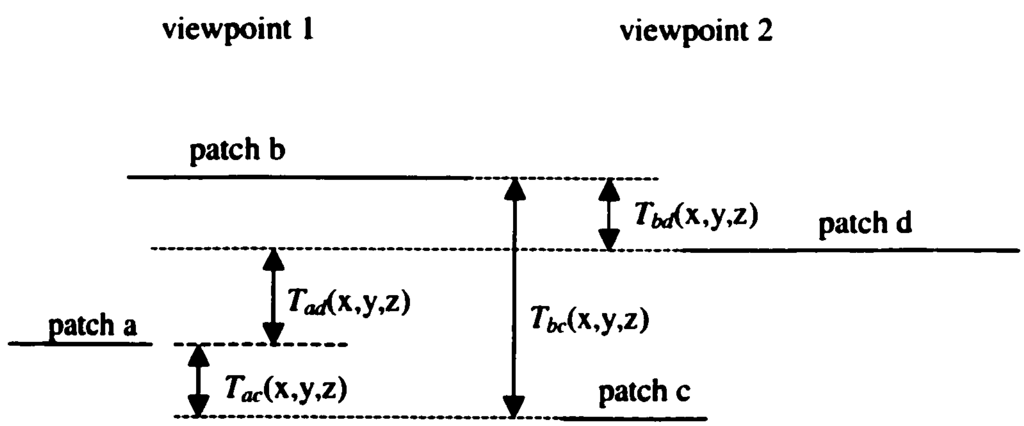
Translation parameters correspond to the shift of matched centroid points in 3D space, as explained in the previous chapter. However, as many patches in the surface description might share the same orientation, this tends to result in some false matches of centroid points. Fortunately, statistics tell us that the correct translation parameter set can be estimated as the one with the maximum number of correspondences. In practice, we therefore only consider those shifts having the maximum of occurrence.

Figure 5.7 and figure 5.8 illustrate this idea in the simple case of two pairs of matching patches. The raw patches from viewpoint 1 and viewpoint 2 are shown together in figure 5.7 (a), and their corresponding patches are redrawn in figure 5.7 (b) after the estimated rotation matrix has been applied to one set of patches. Based on the single criteria of similar orientation, four pairs of matching patches can be obtained, i.e., patch *a* matches patch *c*; patch *a* matches patch *d*; patch *b* matches patch *c*; and patch *b* matches patch *d*. As a result, four possibilities of translation estimates exist: $T_{ac}(x,y,z)$, $T_{ad}(x,y,z)$, $T_{bc}(x,y,z)$ and $T_{bd}(x,y,z)$. However, only $T_{ac}(x,y,z)$ and $T_{bd}(x,y,z)$ lead to a correct pairing of patches. This can be deducted as T_{ac} and T_{bd} are similar, resulting in a double occurrence of the corresponding shift value. The maximum of occurrence can be clearly displayed if we plot the four possible shifts together as shown in figure 5.8. Therefore, the final translation estimate would only take into account $T_{ac}(x,y,z)$ and $T_{bd}(x,y,z)$.

Figure 5.9 illustrates the same idea but with some experimental data. All possible coordinate shifts are now shown as 3D points. We observe that they tend to spread everywhere in the work space. But if the area with the highest density of translation values, which corresponds to the maximum of occurrence, is extracted for each axis, as depicted by the small cube, a good estimate of the translation parameters can be extracted as the weighted average of only those points within the high density area. This estimate then completes the full set of 6 registration parameters that is expected.



(a) Raw patches.



(b) Patches after applying estimated rotation to the second set of patches.

Figure 5.7 Patch matching.

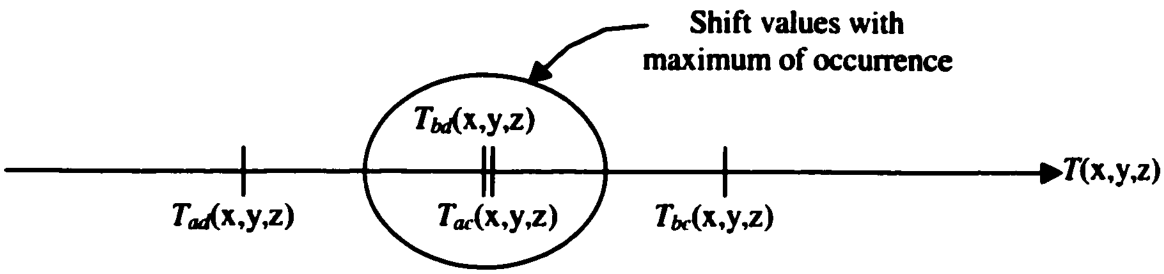


Figure 5.8 Shift occurrence graph for the case in figure 5.7.

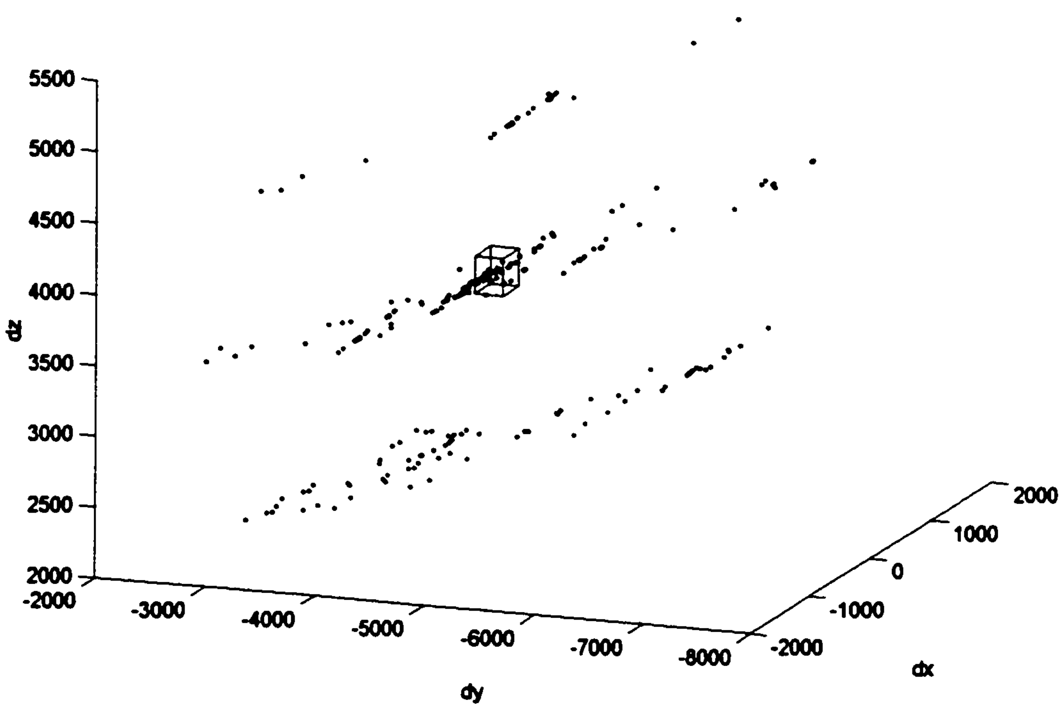


Figure 5.9 Extracting translation parameters from all possible matches.

Chapter 6 Experimental Results

6.1 Introduction

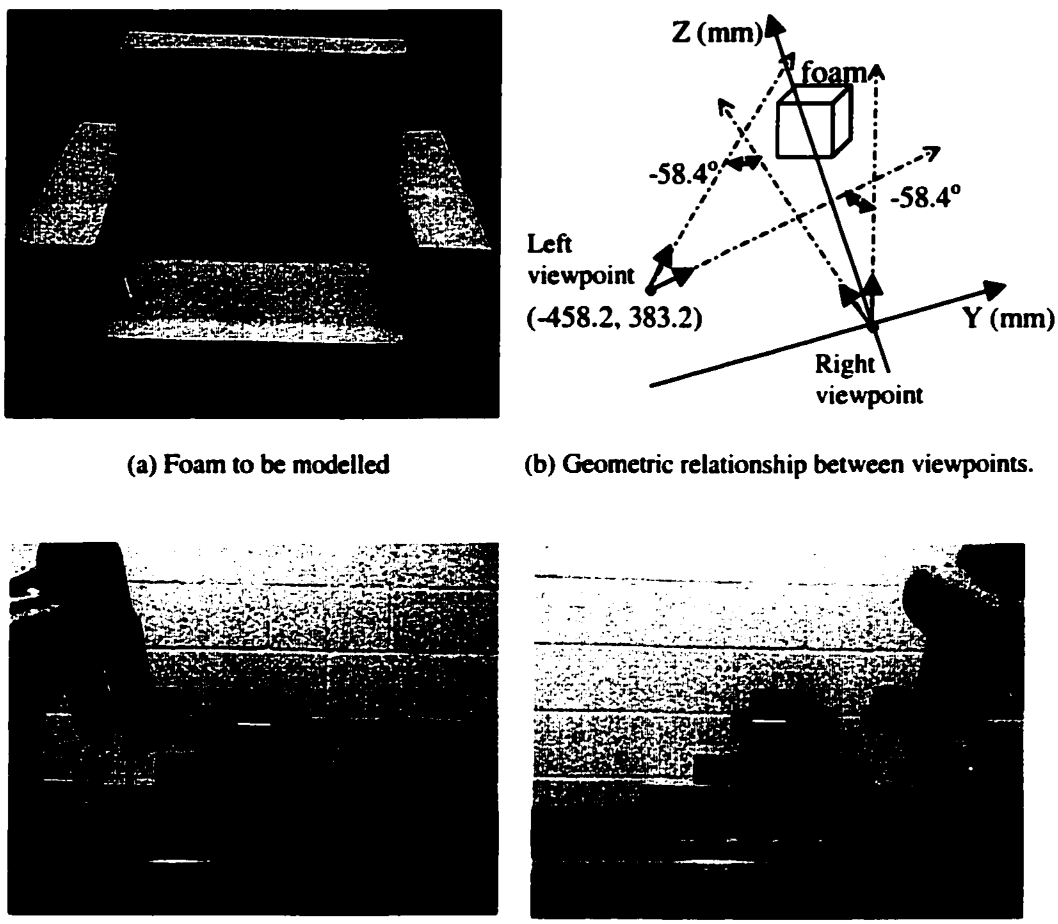
This chapter presents some experimental results obtained with the proposed registration algorithm. The approach is validated both for the 2D and 3D registration cases. The outputs of each step in the 3D registration process are illustrated for completeness. The algorithm is applied on simulated data and on real data collected with the laser range sensor in the experimental setup described in chapter 3. The registration parameters provided from the simulated and the physical setup are referred to as the theoretical values.

The following parameters are used to perform the experiments. The increment of the sensor movement along the X axis between two contiguous scans when collecting the measurements is XSTEP = 5 mm, except for the real computer setup where XSTEP = 3 mm is used; The threshold value to associate one point with a given line segment during line segment approximation is WARP = 4 mm; The threshold value to detect discontinuities on object surface is CLIFF = 10 mm; The maximum allowed rotation for the global distribution of normal vectors belonging to two contiguous scans to be merged together is RotationTh = 0.025 radians; The maximum allowed length difference in percent for normal vectors belonging to two contiguous scans to be merged together is OverlapTh = 20%; The same angular value as RotationTh is taken as the threshold to decide if two normal vectors are close enough to be merged and reduce the size of the surface map, CloseNormal = 0.025 radians. Specifically, RotationTh deals with one group of normal vectors from one range profile at a time, while CloseNormal is concerned about two independent normal vectors. However, both of them consider the overlap of normal vectors. The same parameter value is therefore used as an evaluation factor.

In the following sections, four experiments are presented to validate the proposed algorithm and its implementation. The first one validates the approach in the case of 2D registration. Others are applied to 3D registration.

6.2 2D registration

Figure 6.1 illustrates the experimental testbed for the first experiment. A piece of foam is used as the object to model. It is covered with white paper in order to improve the quality of range measurements. 44 scan lines are captured from each viewing line (left and right) at corresponding elevations and the 2D registration algorithm is applied to each pair of scan lines. The theoretical registration parameters between corresponding scan lines are obtained from the manipulator encoders. The transformation between the viewing lines corresponds to Q = (-58.40 degrees$|_{Rotx}$, -458.2 mm$|_{Ty}$, 383.2 mm$|_{Tz}$), which is defined with respect to the camera reference frame of the top viewpoint in the right viewing line. This geometrical relationship is illustrated in figure 6.1 (b). Experimental results are shown in figure 6.2 and 6.3. In figure 6.2, 42 out of 44 points, each of those being associated with one scan line, provide a very similar rotation estimation. Taking the mean of these points leads to a rotation parameter estimate equal to -58.95°. It is very close to the theoretical value of -58.40°. In the same way, figure 6.3 shows the computed translation parameters (-453.8, 383.6) mm, which are also close to the theoretical values of (-458.2, 383.2) mm. The experimental errors are |58.95 – 58.40| = 0.55 degrees for rotation, and (|453.8–458.2|, |383.6-383.2|) = (4.4, 0.4) mm for translations along the Y and Z axes. The processing time required to get these 44 estimates of transformation is 8 seconds, i.e. 0.18 second per pair of scan lines when the data processing module written in Matlab is running on a Pentium III, 933MHz processor with 256M RAM. This experiment demonstrates the validity of the basic idea of our algorithm.



(a) Foam to be modelled

(b) Geometric relationship between viewpoints.

(c) Left view.

(d) Right view.

Figure 6.1 Experimental setup for the first experiment.

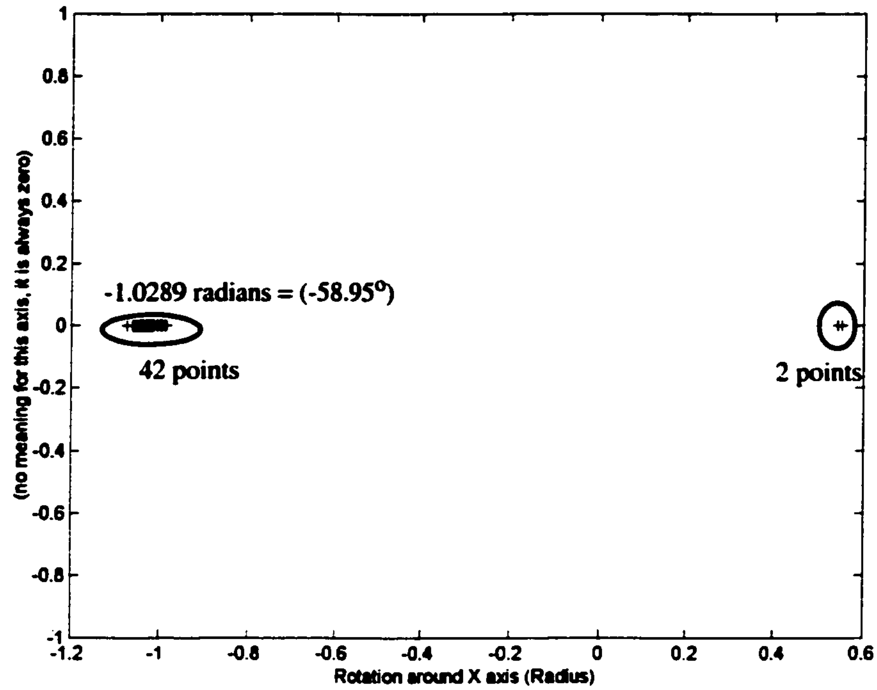


Figure 6.2 Rotation parameter extraction (2D registration case).

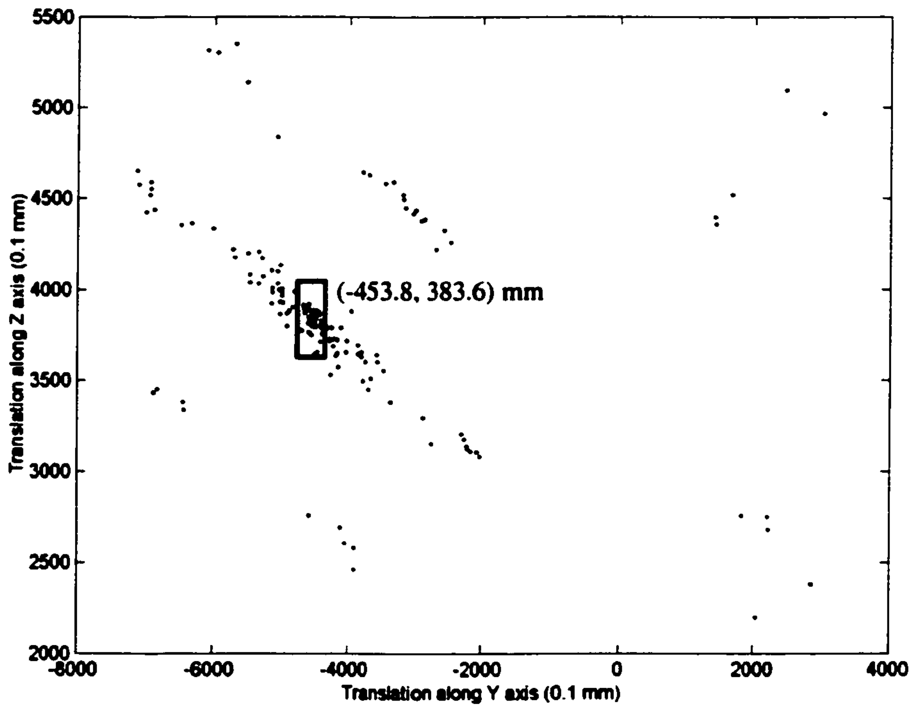


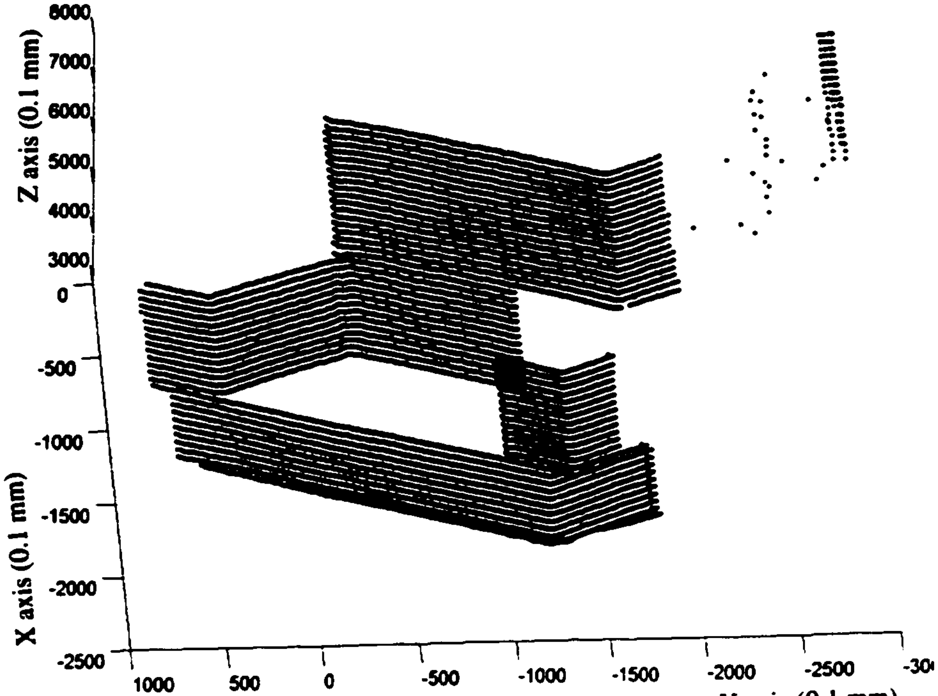
Figure 6.3 Translation parameters extraction (2D registration case).

6.3 3D registration with a simple object

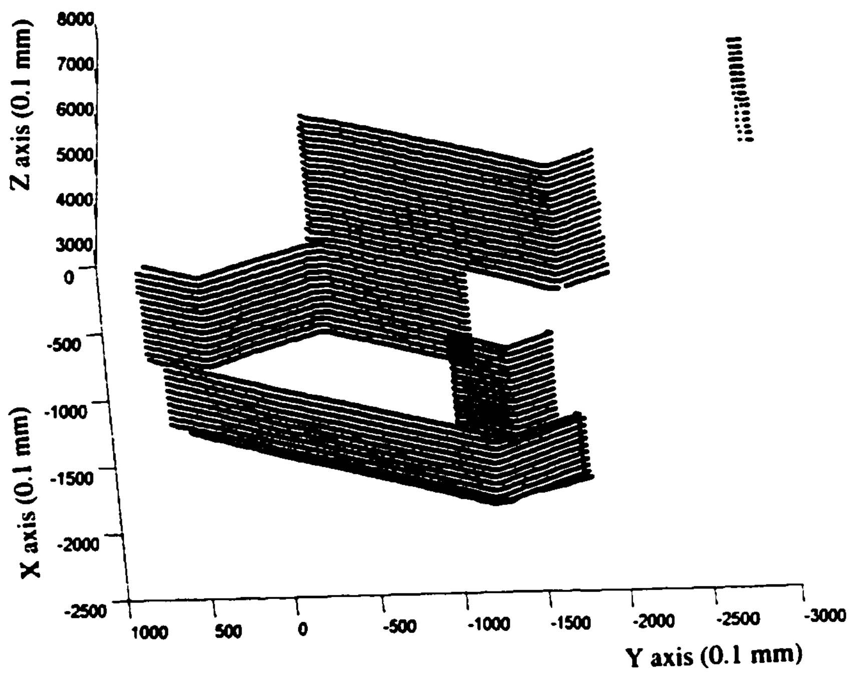
The second experiment employs the same testbed and data as for the first experiment. The difference is that we consider the 44 scan lines from one viewing area as a virtual range image, which includes about 5280 (=44*120) points. Therefore we compute the registration between 3D object surfaces rather than between individual 2D scan lines. Figure 6.4 shows the set of raw range measurements (44 scan lines) from the right side viewing line before and after preprocessing (outliers removing and measurements smoothing). After applying the line segment approximation and the scan merging processes, we plot the right view and its simplified patch representation in figure 6.5. The corresponding plots for the left view are illustrated in figure 6.6. As shown in figures 6.5 and 6.6, creating this compact representation of the object surfaces leads to a significant reduction of the data set from thousands of points to 32 patches (right view), and to 46 patches (left view).

The following step consists in merging patches that have similar orientations (within some tolerance level) for each view. As a result, the representation can still be simplified to 9 normal vectors (right view) and 10 normal vectors (left view). The modified Gauss spheres for the right and the left view measurements are plotted in figure 6.7. This compact mapping significantly reduces the workload for the registration estimation phase.

Matching the vectors from the two modified Gauss spheres according to the proposed algorithm, we obtain the rotation parameters $R = (-57.93^{o}|_{Rotx}, 1.13^{o}|_{Roty}, 0|_{Rotz})$. Applying these rotation values, the coordinate shifts between corresponding centroid points are computed as plotted in figure 6.8. From these shifting values, we estimate the translations to be $T = (1.44|_{Tx}, -448.5|_{Ty}, 375.7|_{Tz})$ mm.

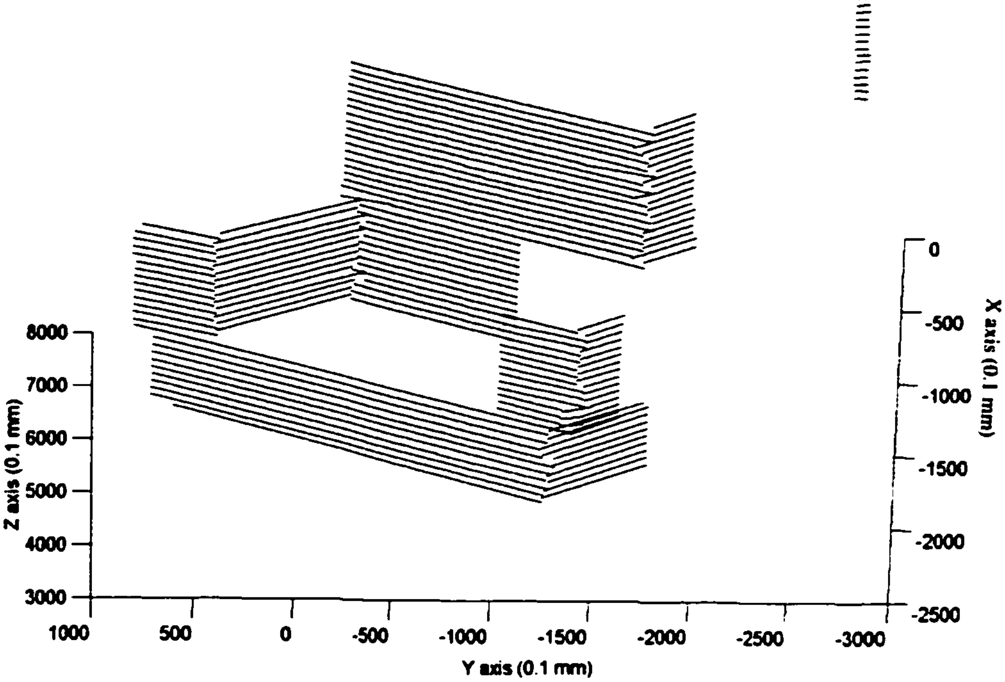


(a) Before filtering.

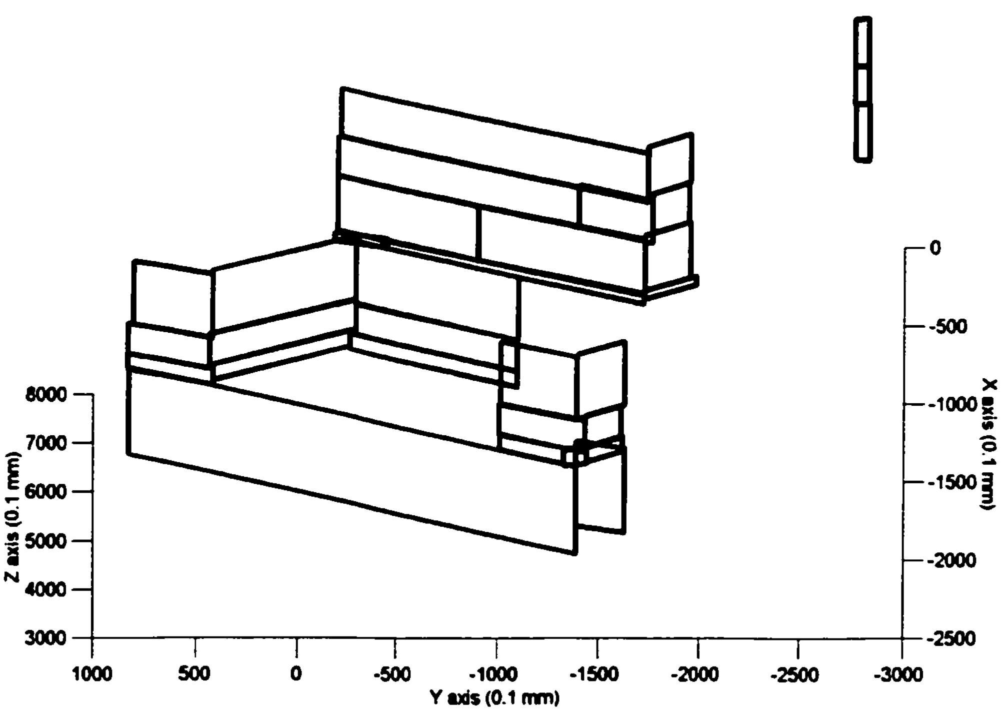


(b) After filtering.

Figure 6.4 Raw range measurements from the right view (foam).

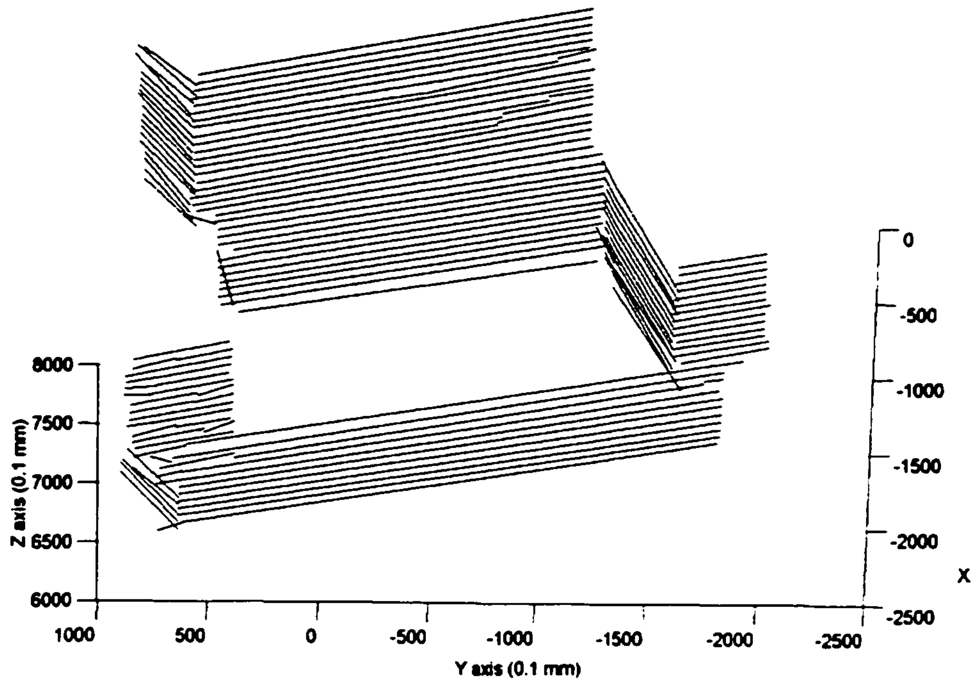


(a) Line segment approximation.

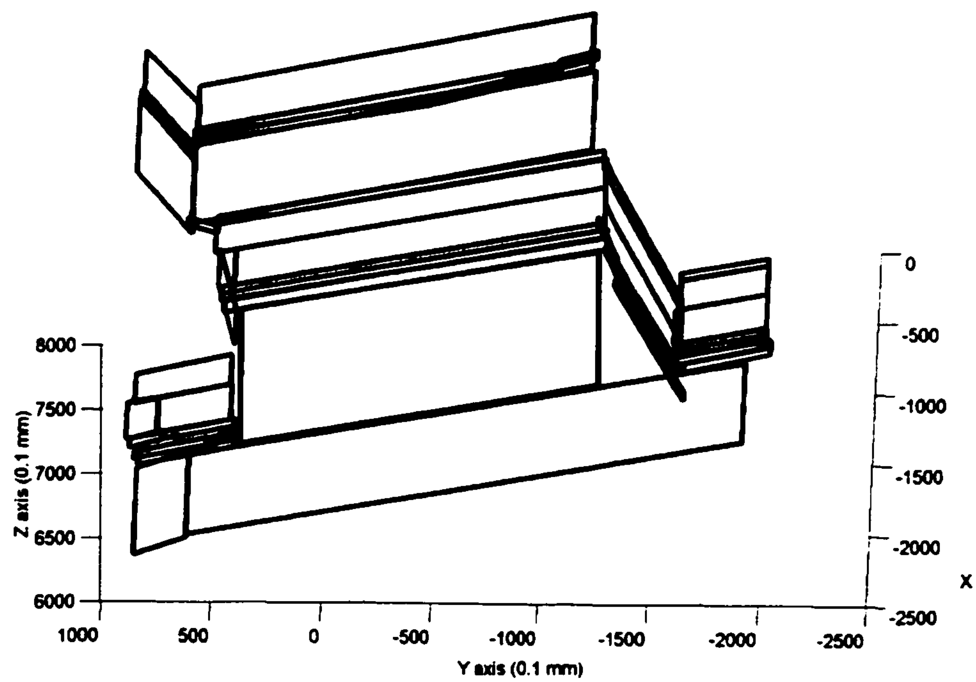


(b) Patch representation.

Figure 6.5 Compact representation of the object surface from the right view (foam).

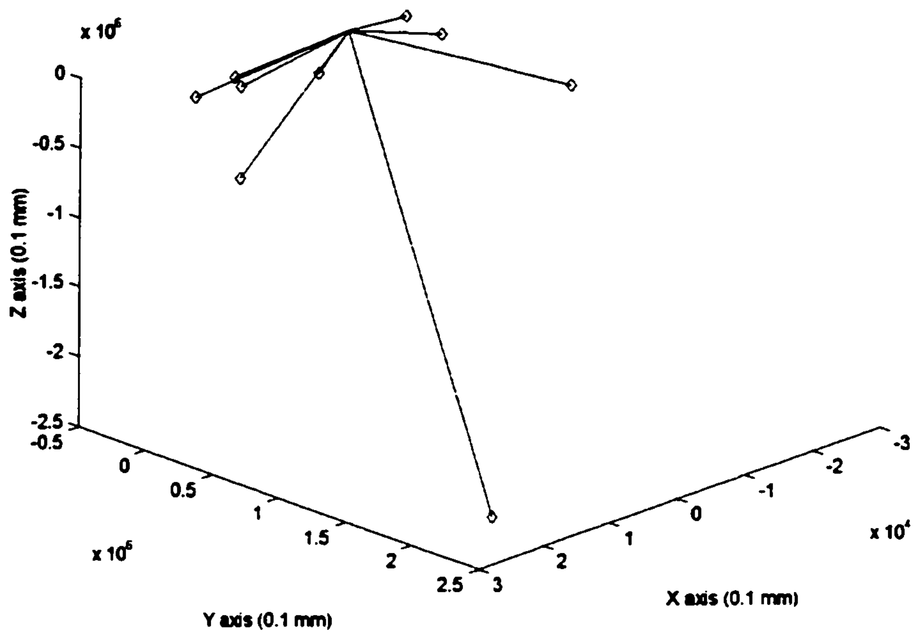


(a) Line segment approximation.

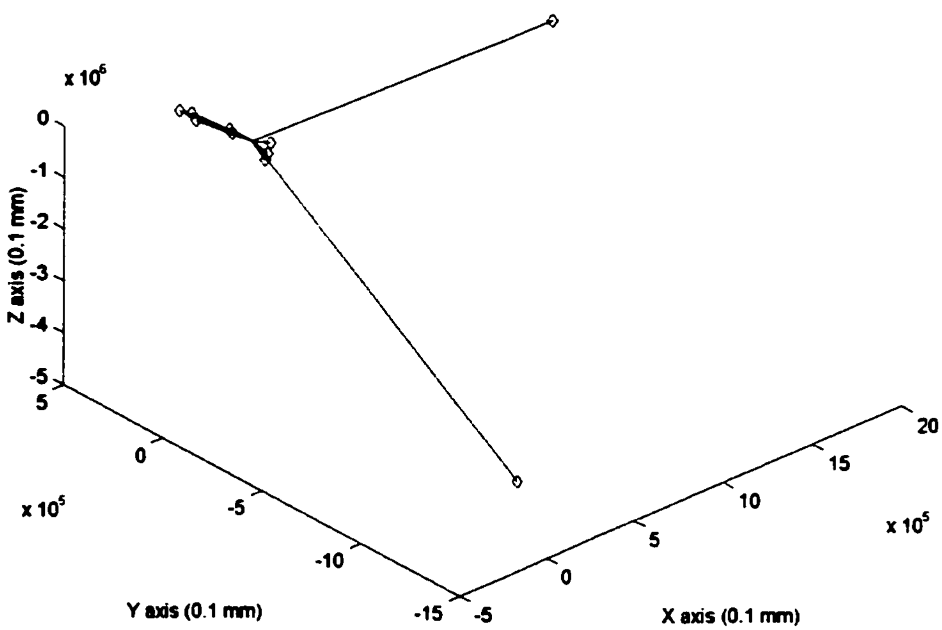


(b) Patch representation.

Figure 6.6 Compact representation of the object surface from the left view (foam).



(a) Right view.



(b) Left view.

Figure 6.7 Modified Gauss sphere representing the compact maps (foam).

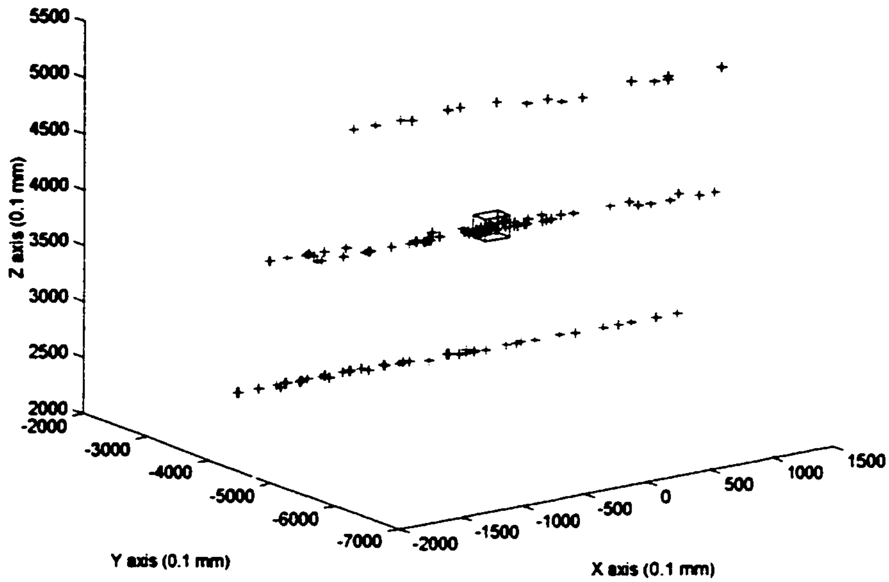
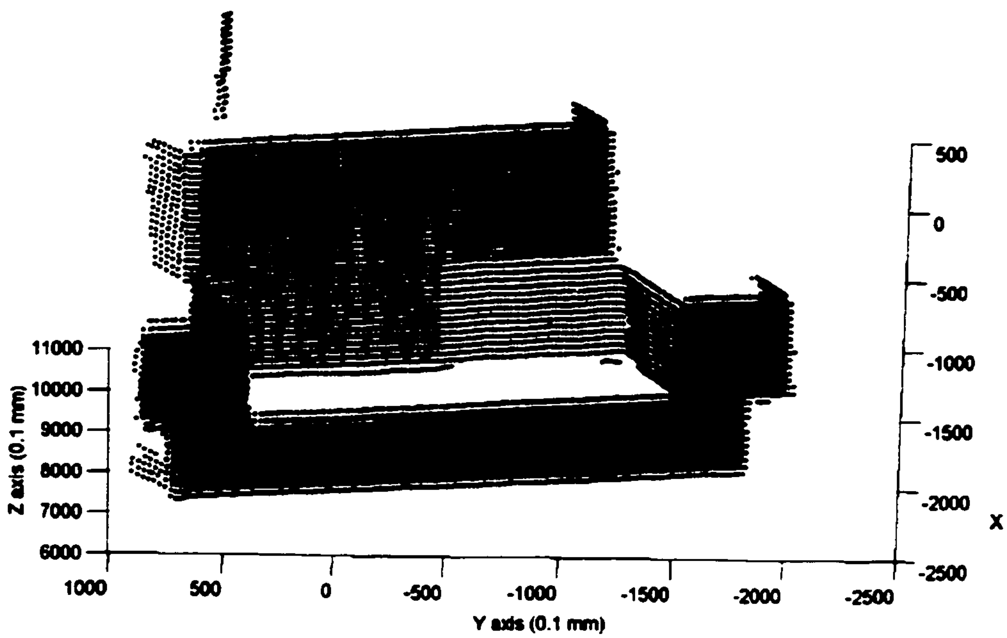


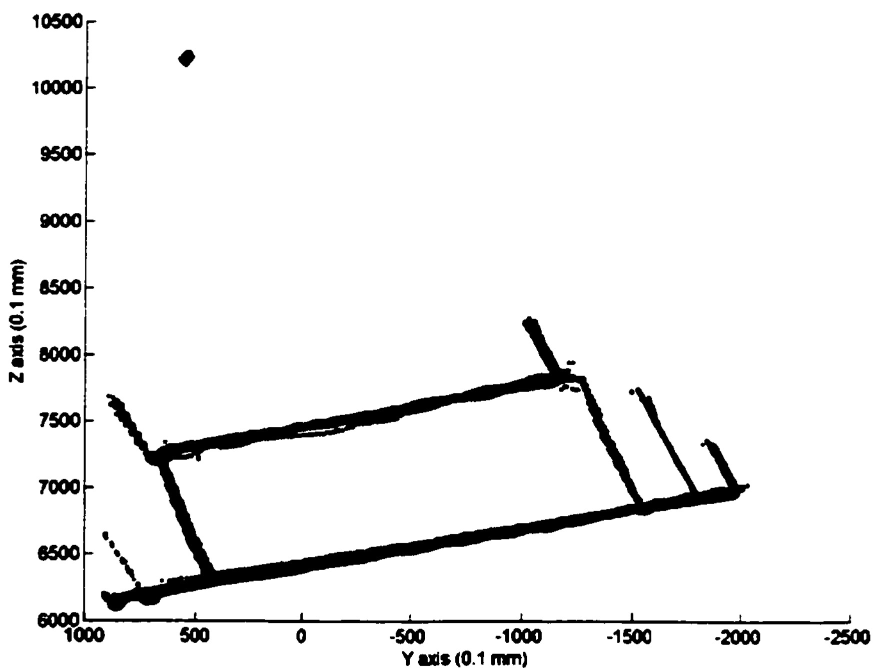
Figure 6.8 Occurrence graph of shifting values between centroids of the right and the left views (foam).

Figure 6.9 shows the superposition of the two sets of raw range measurements after the application of the estimated transformation matrix. This validates the efficiency of our algorithm. Table 6-1 compares the experimental results with the expected theoretical values. It demonstrates that rotation parameter estimates are quite accurate. For translation parameters, the estimates suffer from a slightly higher level of error. This fact is due to the difficulty to determine the region of maximum of occurrence in the occurrence graph that is used to extract translation values. The application of imprecise rotation estimates before translation estimation also makes some contribution of the translation errors.

Computation time for the whole process in this example is about 16 seconds still using Matlab code running on a Pentium III, 933MHz processor. More CPU time is required than in the 2D registration case because the full set of 6 transformation parameters are estimated here rather than only 3 transformation parameters in the previous case.



(a) Front view.



(b) Top view.

Figure 6.9 Superposition of the two sets of data after registration (foam).

Transformation Parameters	Left to Right Viewpoints	
	Theoretical Values	Experimental Results
Rotation (X)	-58.40^o	-57.93^o
Rotation (Y)	0^o	1.13^o
Rotation (Z)	0^o	0^o
Translation (X)	0 mm	1.44 mm
Translation (Y)	-458.2 mm	-448.5 mm
Translation (Z)	383 mm	375.7 mm

Table 6.1 Transformation parameters between two viewpoints on the foam object.

6.4 3D registration with simulated range measurements

Three sets of simulated measurements generated by a range sensor simulator operating on a virtual scene representation were provided to perform the third experiment. 40 noise free scan lines containing 256 measurement points are generated from each viewing region: left view, center view, and right view. The geometrical relationships between these three views are illustrated in figure 6.10. The virtual scene is built according to the real scene as depicted in figure 6.11 except that some simple blocks are used to replace the objects in the scene. The simulated range measurements from the left, the center and the right views are plotted in figures 6.12, 6.13 and 6.14 respectively.

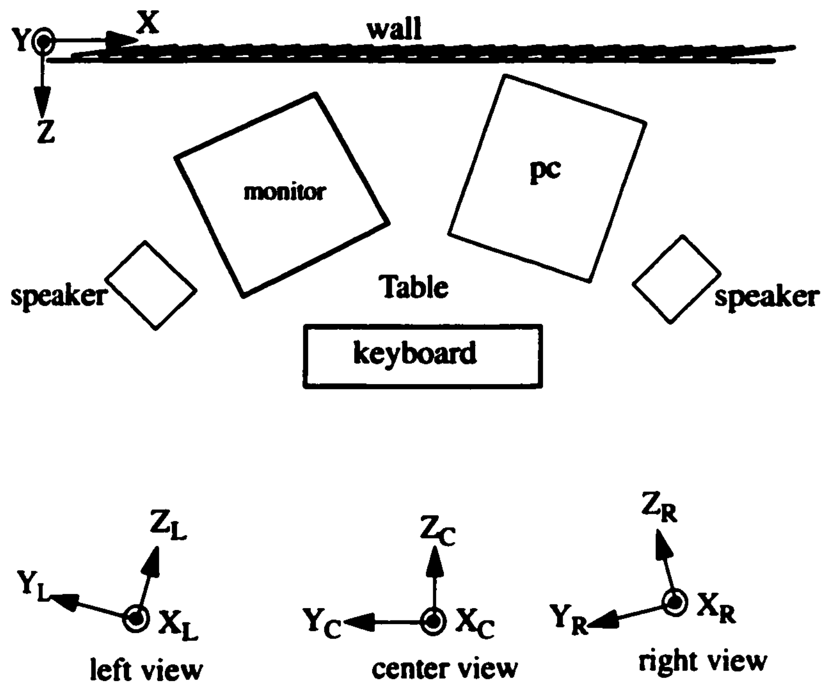


Figure 6.10 Geometrical relationships between viewpoints.

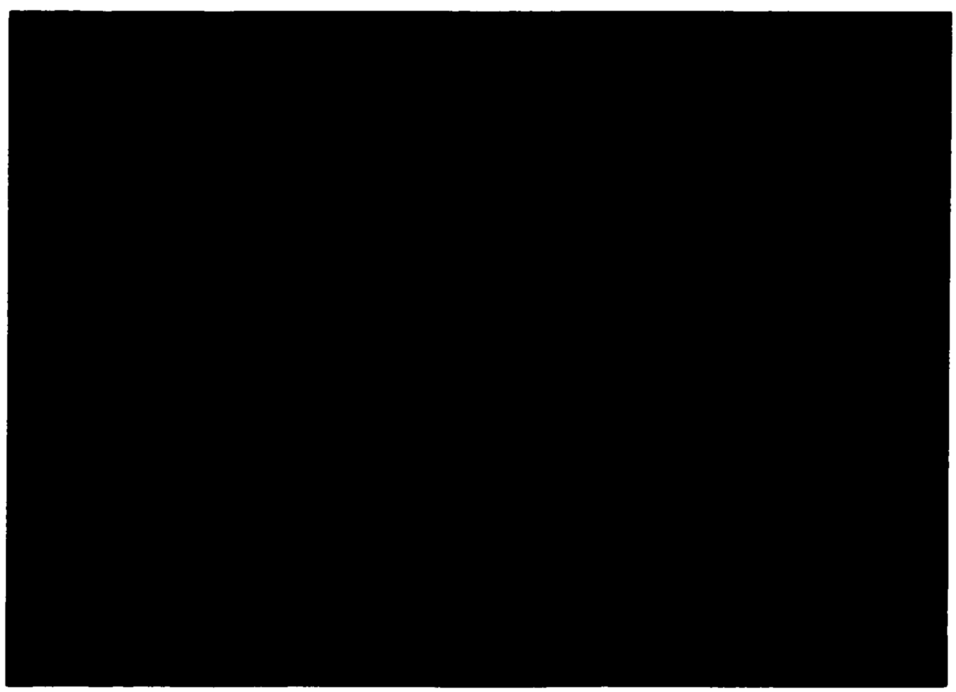


Figure 6.11 Computer scene layout.

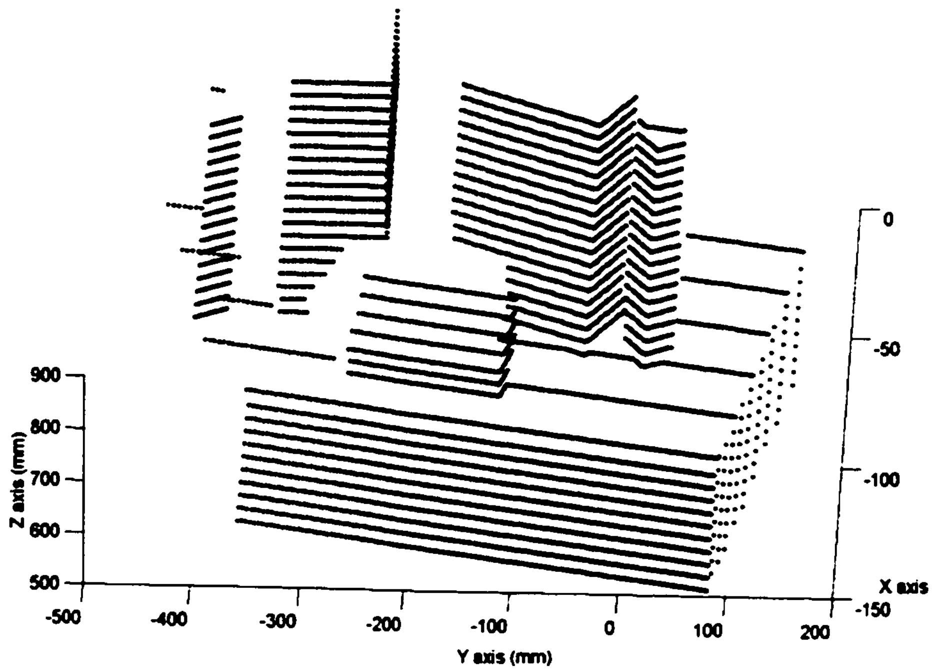


Figure 6.12 Raw measurements from the left view (virtual computer scene).

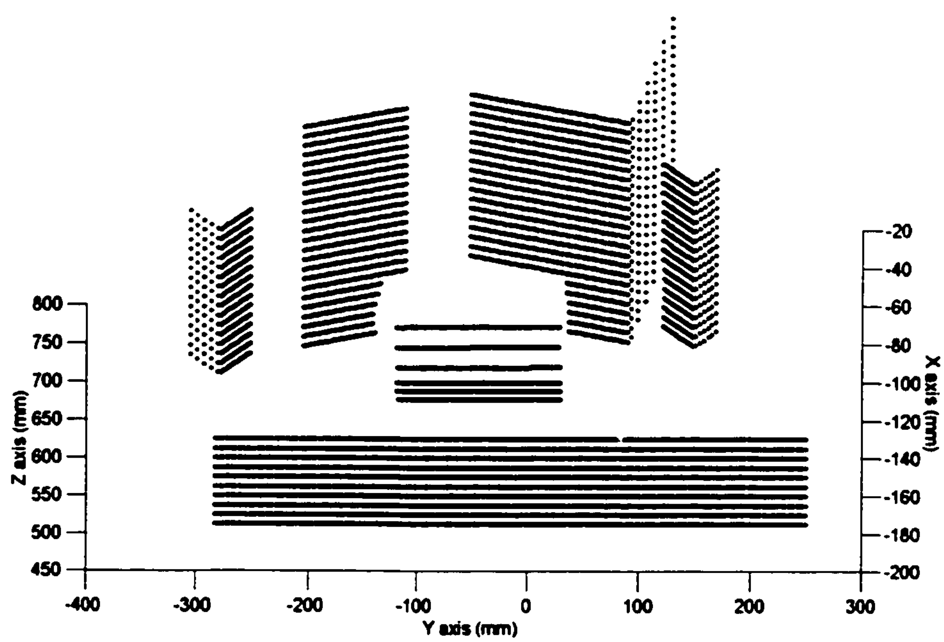


Figure 6.13 Raw measurements from the center view (virtual computer scene).

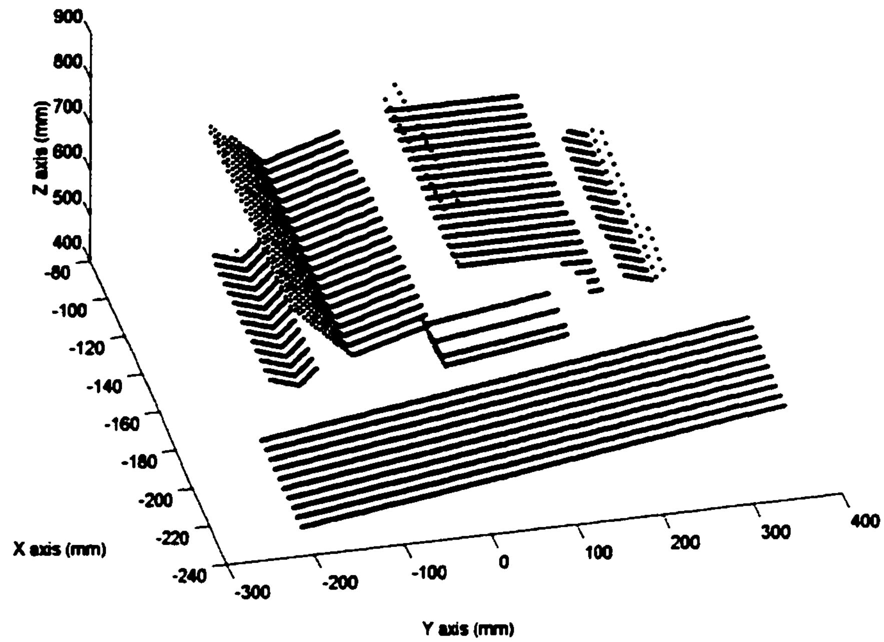


Figure 6.14 Raw measurement from the right view (virtual computer scene).

Figures 6.15, 6.16 and 6.17 show the simplified patch representations for the left, the center and the right views respectively. Their corresponding modified Gauss spheres after merging patches that have similar orientations (within some tolerance level) for each viewpoint are illustrated in figure 6.18, 6.19 and 6.20. This compact representation allows one to map about ten thousand (40*256) points by 13 normal vectors for the left view, 8 normal vectors for the center view and 11 normal vectors for the right view. Starting from this compact mapping, we successively run the program to compute the registration between the center and the left views, between the center and the right views, and between the left and the right views. We observed that a larger error on the translation along Y axis is obtained when we try to estimate the registration between the left and the right views. This is explained by the fact that there is a larger shift along the Y axis between the left and the right viewpoints.

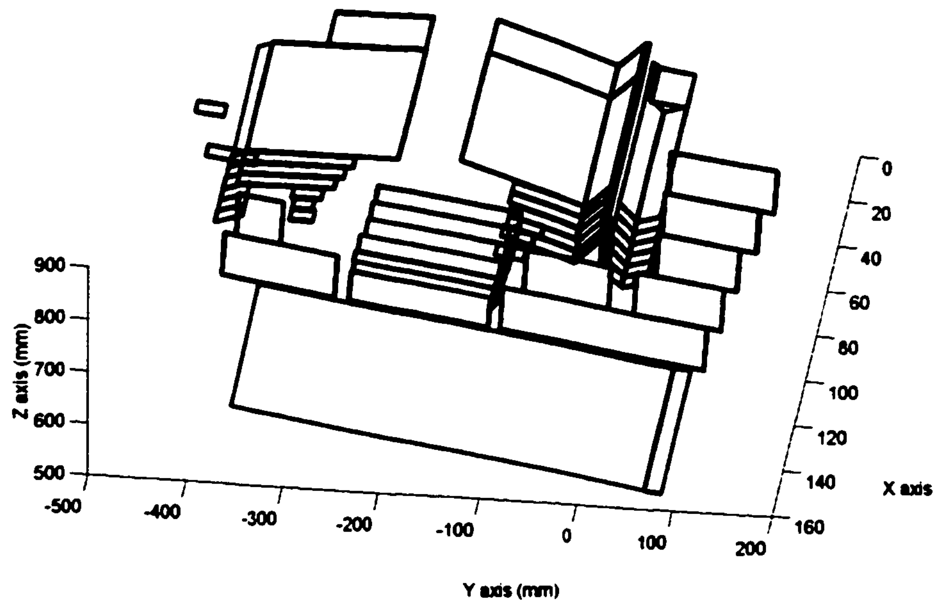


Figure 6.15 Patch representation for the left view (virtual computer scene).

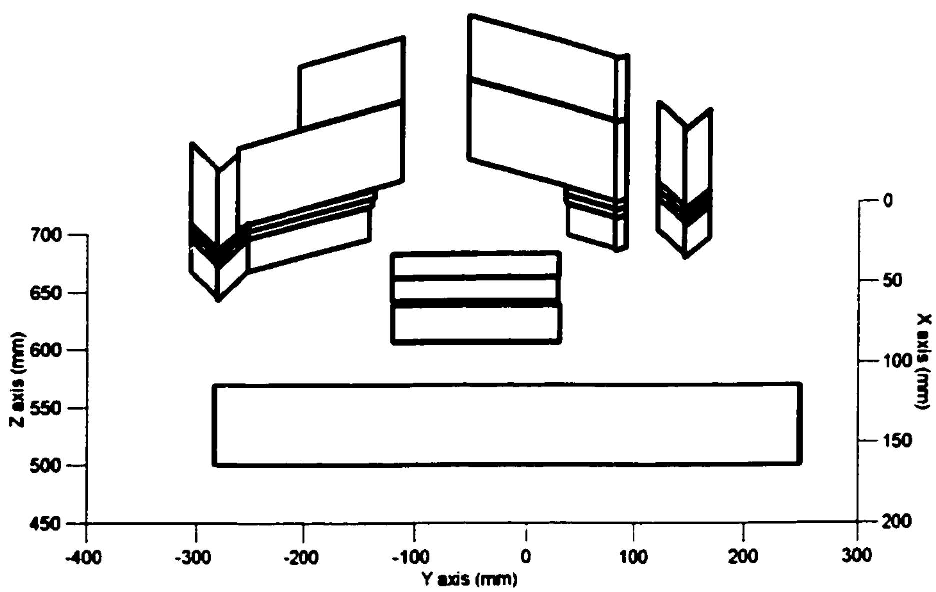


Figure 6.16 Patch representation for the center view (virtual computer scene).

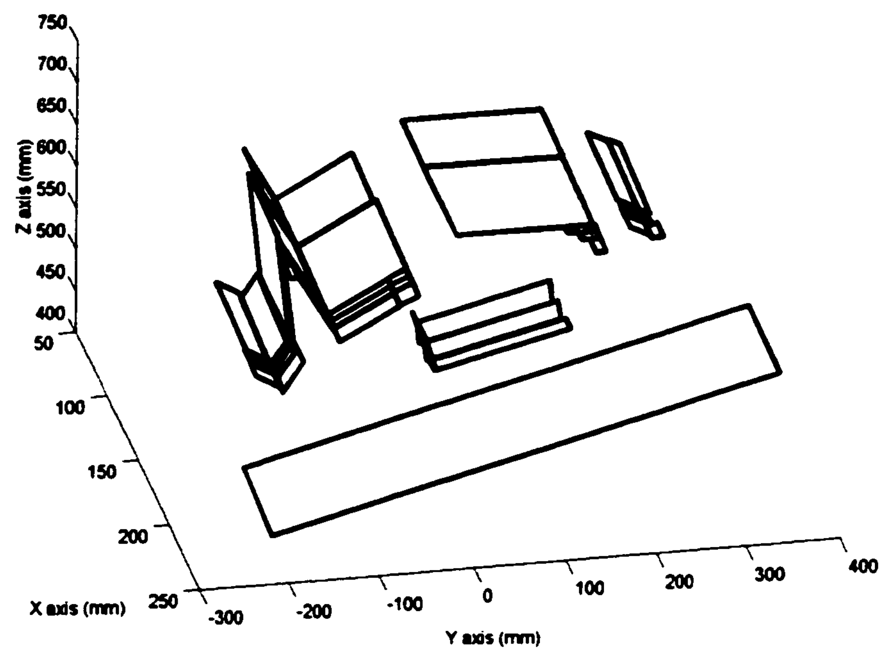


Figure 6.17 Patch representation for the right view (virtual computer scene).

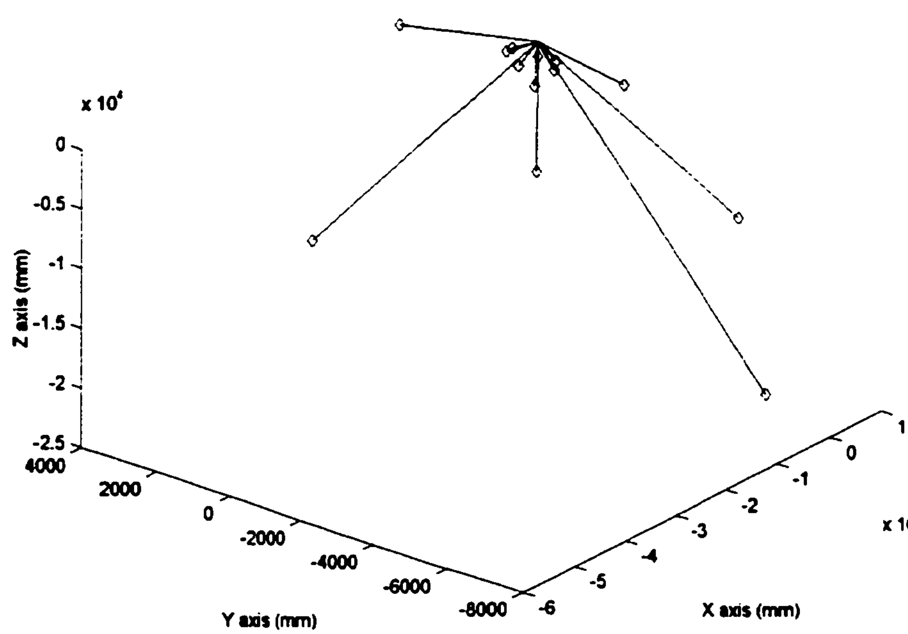


Figure 6.18 Modified Gauss sphere for the left view (virtual computer scene).

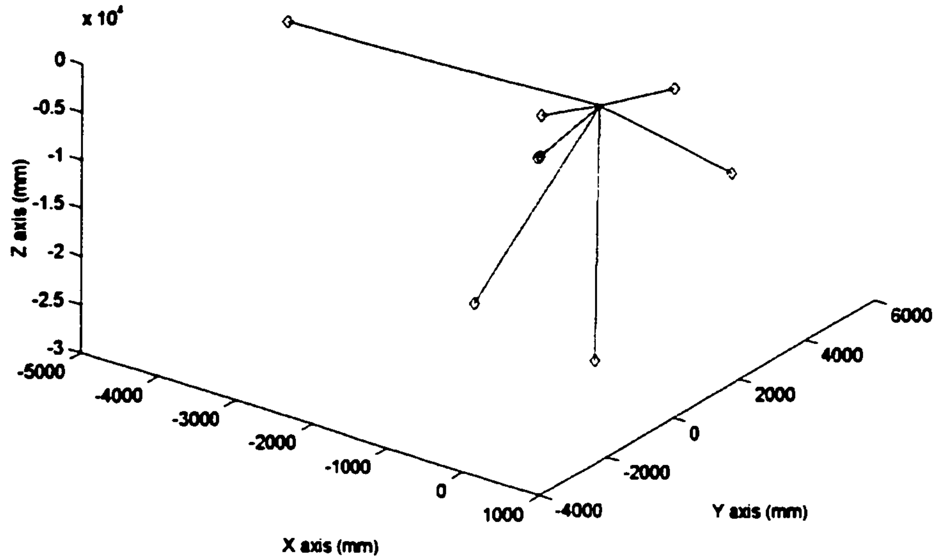


Figure 6.19 Modified Gauss sphere for the center view (virtual computer scene).

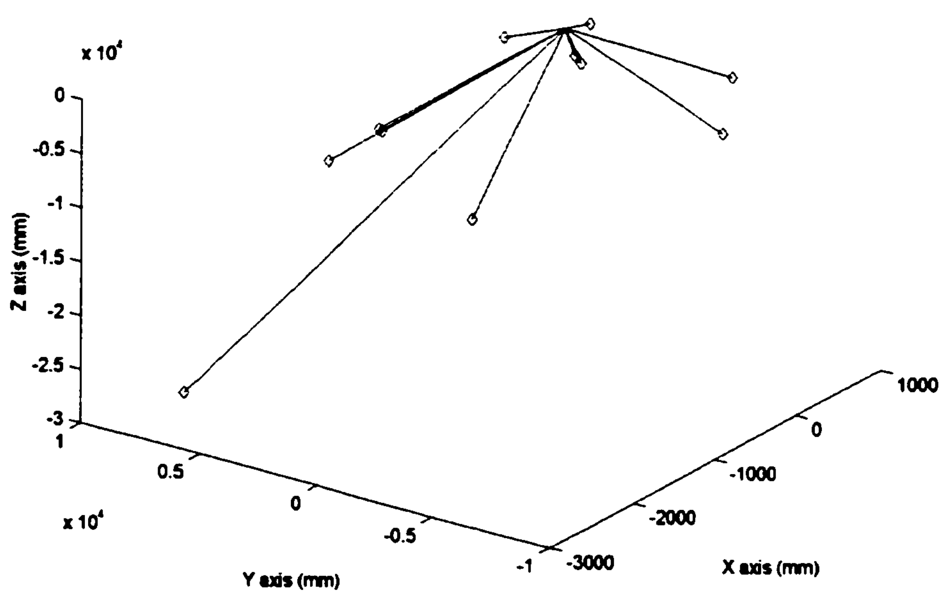


Figure 6.20 Modified Gauss sphere for the right view (virtual computer scene).

Actually, the registration between the left and the right views is better extracted from the two registrations between the left and the center views, and between the right and the center views which are closer one to each other.

Figures 6.21 and 6.22 show the superposition of range measurements originating from the left and the center views, and from the right and the center views after the experimental transformation matrix is applied to one view. When the three views are superposed, the full scene structure is reproduced as depicted in figure 6.23 and the alignment between range images is very good.

Numerical experimental results are presented in tables 6-2 and 6-3 along with the expected theoretical values. Comparing the theoretical and the experimental values, a precision of up to 0.07 degrees in rotation, and 0.1 mm in translation can be achieved. Larger deviations of the translation parameters estimated are limited to about 2% of the theoretical values. Given the difficulties to locate the actual region of maximum of occurrence in the occurrence graph and the fact that a highly simplified surface representation is used to speed up the registration process, these results can be considered very satisfactory.

The required computing time for each of the above two registrations is about 6 seconds respectively. This is less than in the second experiment while a similar number of scan lines are used. This reduction in the computing time is mainly due to the fact that noise free measurements are used in this experiment. This facilitates the merge of scan lines and the determination of the compact surface maps. Also the number of resulting planar patches tend to be smaller, which shortens the search of matching normal vectors.

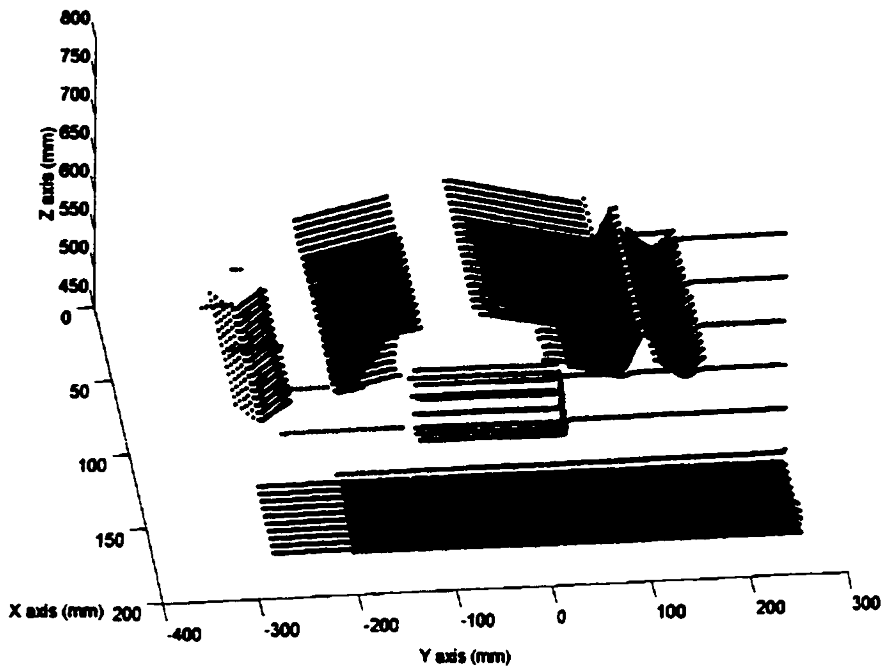


Figure 6.21 Superposed range measurements after registration between the left and the center views (virtual computer scene).

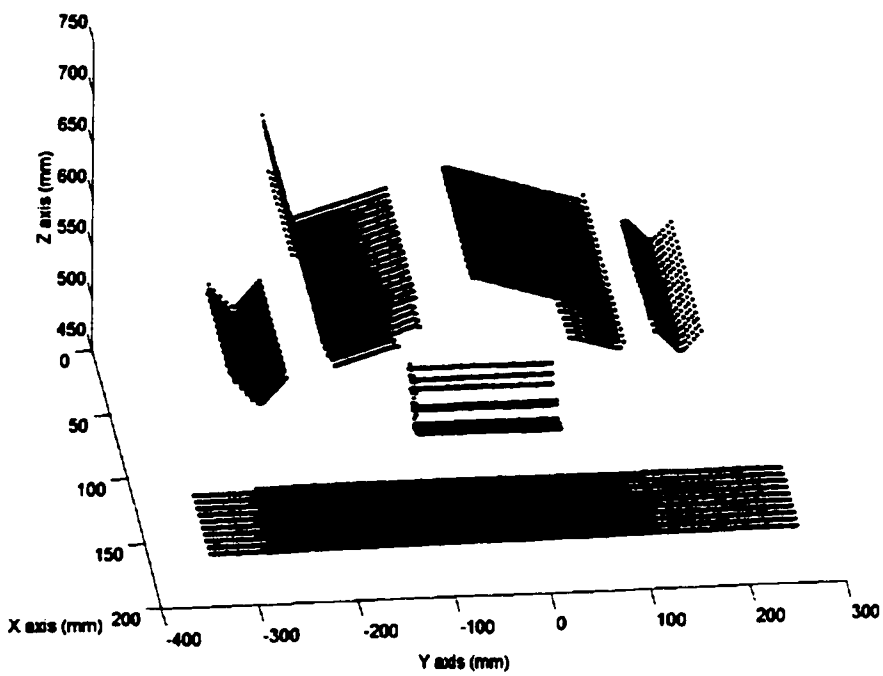
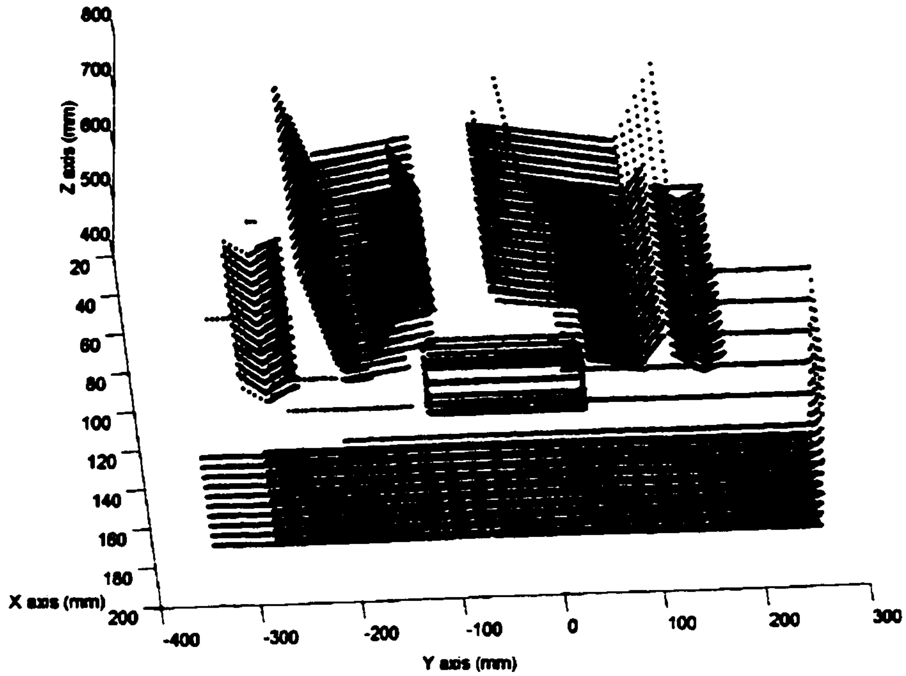
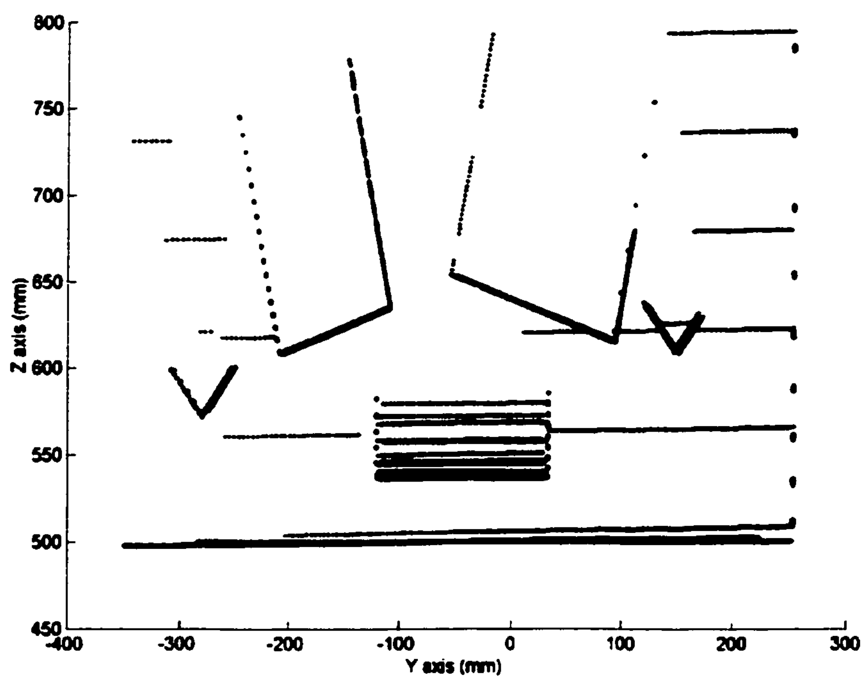


Figure 6.22 Superposed range measurements after registration between the right and the center views (virtual computer scene).



(a) Left, center and right views (front view).



(b) Left, center and right views (top view).

Figure 6.23 Superposed range measurements from the three views after registrations (virtual computer scene).

Transformation Parameters	Center to Left Viewpoint	
	Theoretical Values	Experimental Results
Rotation (X)	0°	-0.11°
Rotation (Y)	5.15°	5.06°
Rotation (Z)	14.89°	15.01°
Translation (X)	0 mm	2.6 mm
Translation (Y)	300 mm	304.5 mm
Translation (Z)	0 mm	0.1 mm

Table 6.2 Transformation parameters between the center and the left views on the virtual computer setup.

Transformation Parameters	Center to Right Viewpoint	
	Theoretical Values	Experimental Results
Rotation (X)	0°	-0.07°
Rotation (Y)	-5.15°	-4.91°
Rotation (Z)	-14.89°	-14.38°
Translation (X)	0 mm	3.57 mm
Translation (Y)	-260 mm	-254.9 mm
Translation (Z)	0 mm	-1.1 mm

Table 6.3 Transformation parameters between the center and the right views on the virtual computer setup.

6.5 3D registration with real range measurements

The fourth experiment is conducted on the real computer scene shown in figure 6.11. This time, the objects are also scanned from the left, the center and the right viewing lines but with a real Jupiter laser range finder as described in chapter 3. Plots of raw range measurements from these three views are shown in figures 6.24, 6.25 and 6.26. The high density of data results from the fact that there are 130 scan lines in each view rather than 40. Figures 6.27, 6.28 and 6.29 illustrate their corresponding simplified patch representations.

As in the previous experiment, the proposed algorithm has been applied on the data to estimate the transformation parameters between the left and the center views, and between the center and the right views. Experimental results are presented in tables 6-4 and 6-5 along with the expected theoretical values. The imprecise location of the region of maximum of occurrence in the occurrence graph and the error contribution from the application of imprecise rotation estimates can also justify the larger deviations in the estimates of translation parameters. But these errors also partially come from the imprecision on the theoretical values that we extracted from the real experimental setup. This will be discussed in the next section. Superposing the raw measurements of figures 6.24, 6.25 and 6.26 after the application of the estimated transformations, we observe that the alignment between matching surfaces is appropriate as shown in figure 6.30.

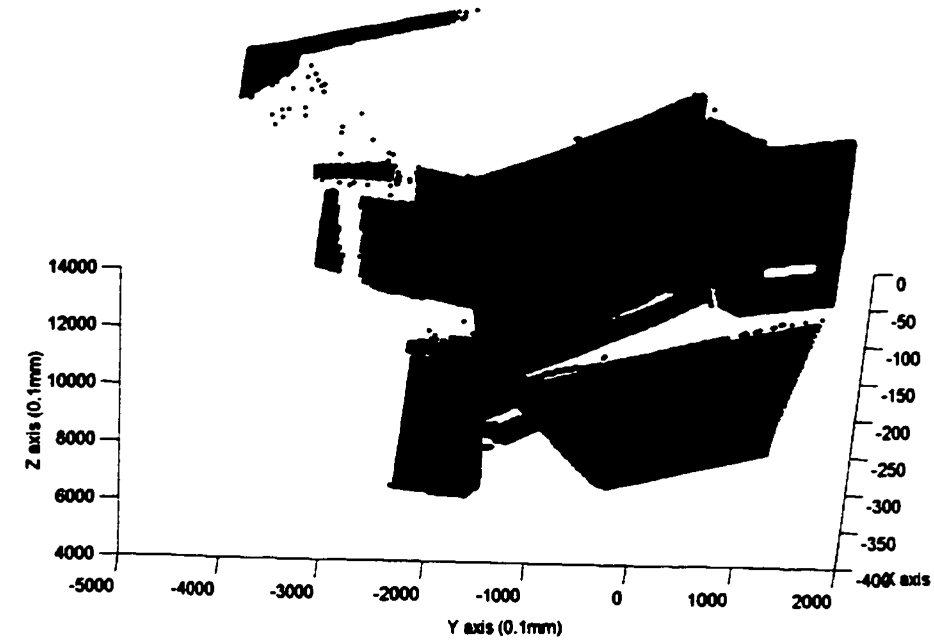


Figure 6.24 Raw measurements from the left view (real computer scene).

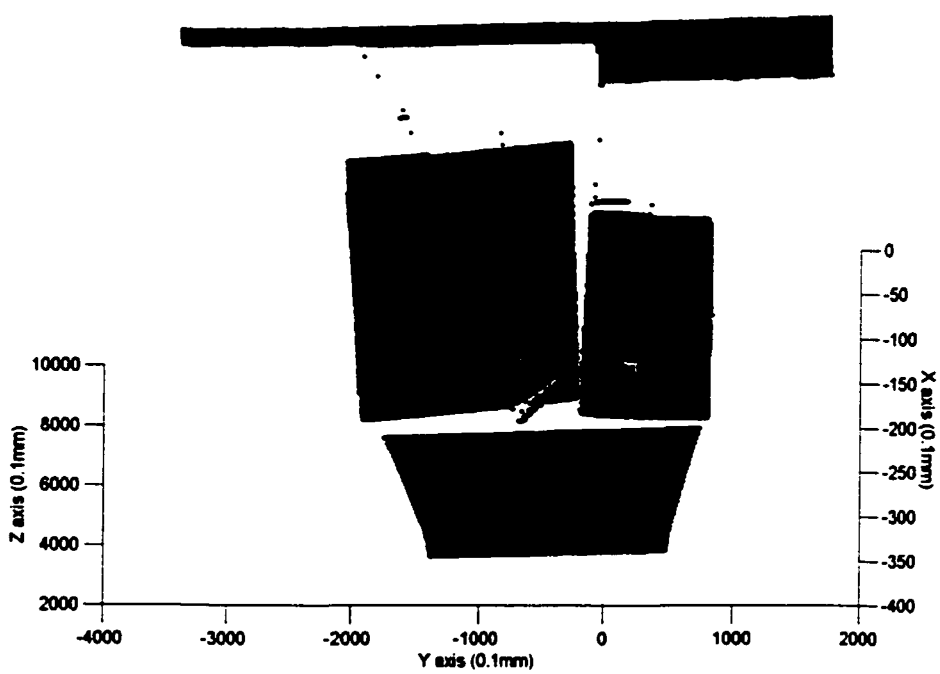


Figure 6.25 Raw measurements from the center view (real computer scene).

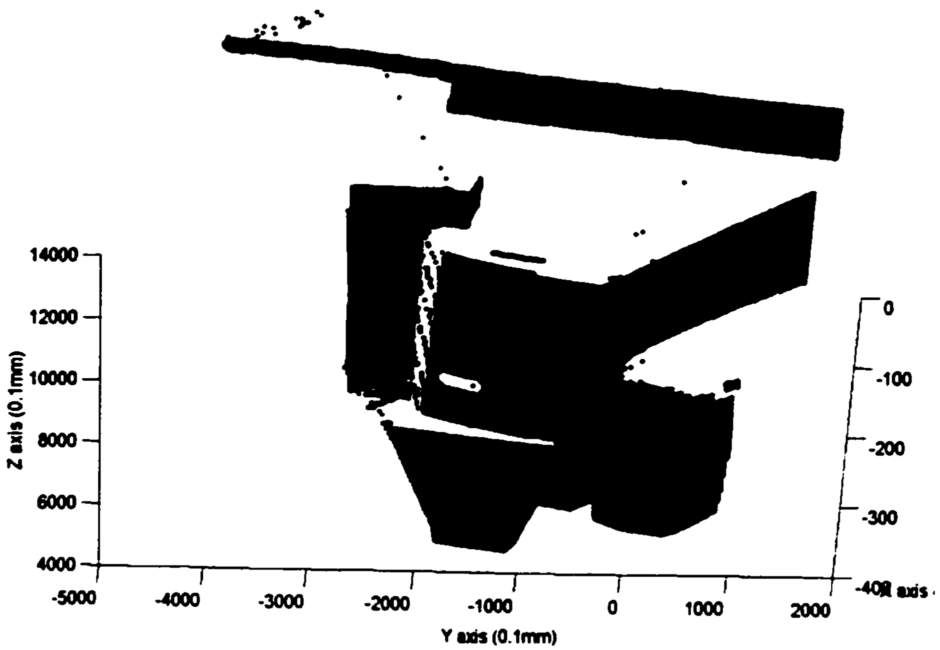


Figure 6.26 Raw measurements from the right view (real computer scene).

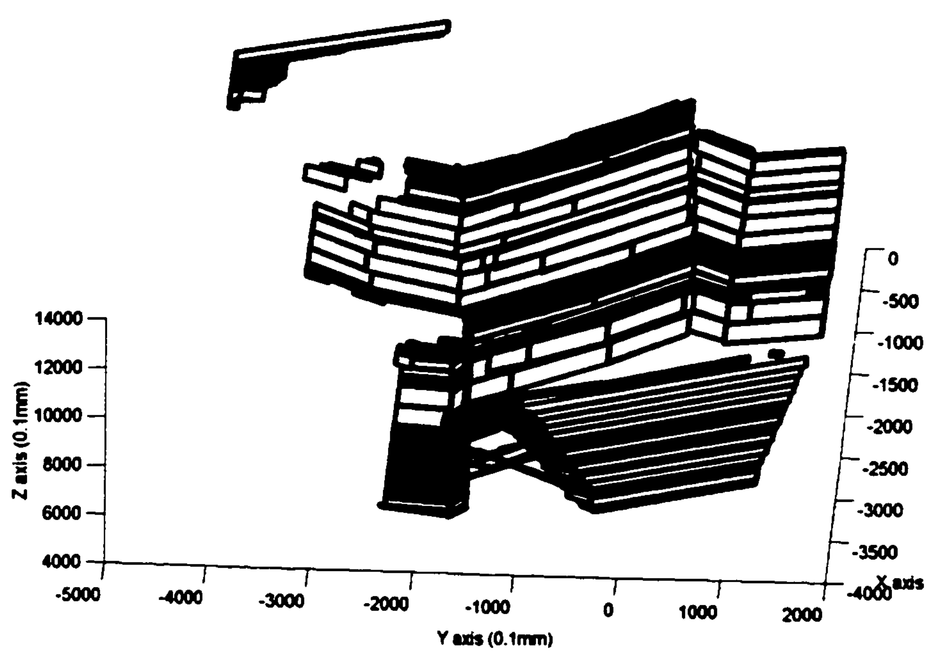


Figure 6.27 Patch representation for the left view (real computer scene).

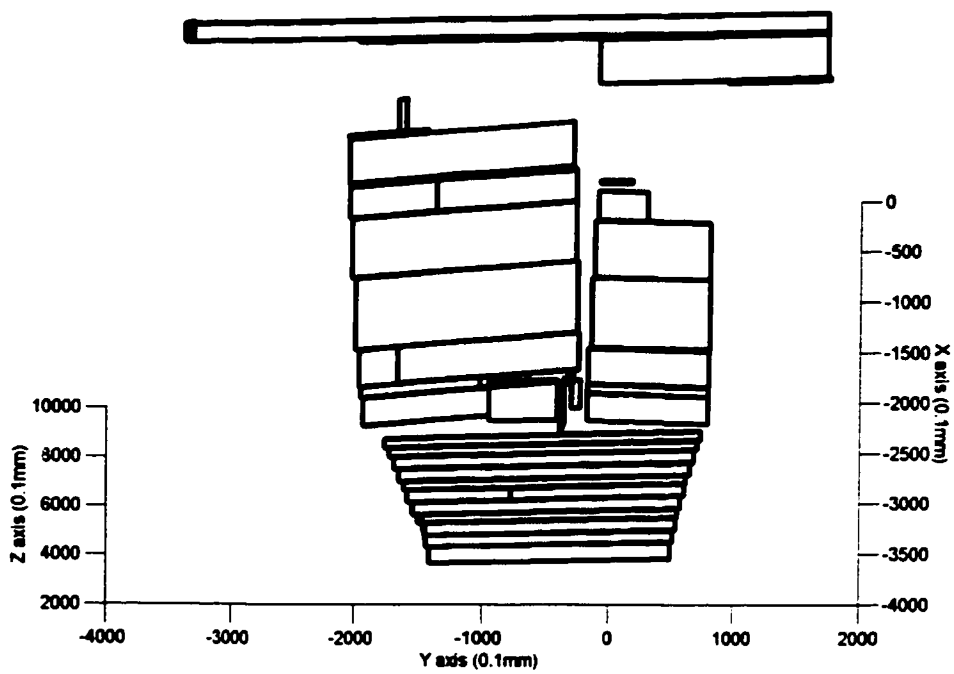


Figure 6.28 Patch representation for the center view (real computer scene).

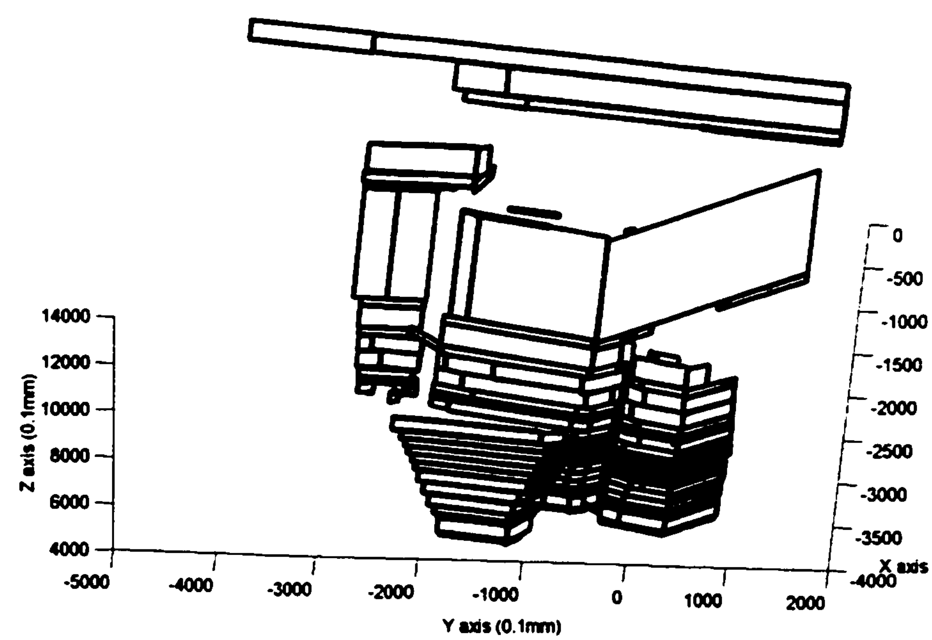


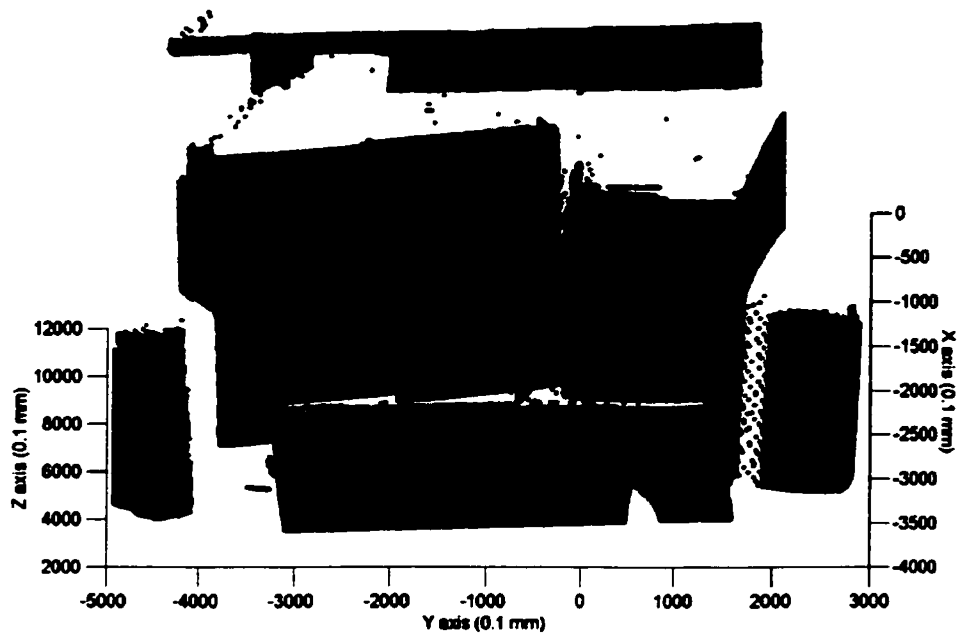
Figure 6.29 Patch representation for the right view (real computer scene).

Transformation Parameters	Left to Center viewpoints	
	Theoretical Values	Experimental Results
Rotation (X)	0^{o}	0.38^{o}
Rotation (Y)	0^{o}	-0.24^{o}
Rotation (Z)	-25.44^{o}	-25.43^{o}
Translation (X)	0 mm	-1.9 mm
Translation (Y)	-410.2 mm	-396.6 mm
Translation (Z)	188.8 mm	186.4 mm

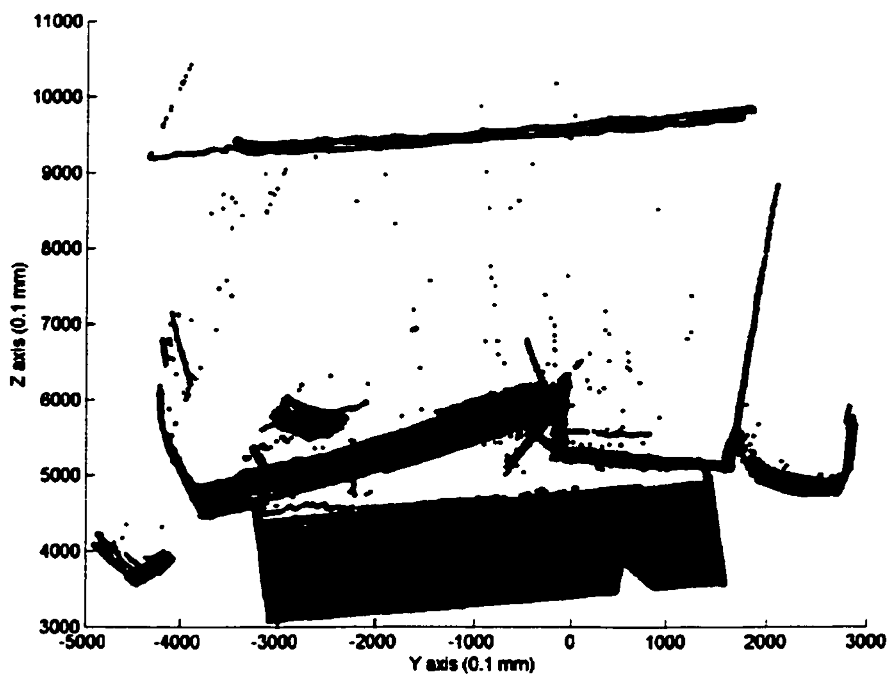
Table 6.4 Transformation parameters between the left and the center views on the real computer setup.

Transformation Parameters	Center to Right viewpoints	
	Theoretical Values	Experimental Results
Rotation (X)	0^{o}	1.23^{o}
Rotation (Y)	0^{o}	-0.24^{o}
Rotation (Z)	-34.43^{o}	-34.73^{o}
Translation (X)	0 mm	6.7 mm
Translation (Y)	-651.4 mm	-653.3 mm
Translation (Z)	-229.6 mm	-226.6 mm

Table 6.5 Transformation parameters between the center and the right views on the real computer setup.



(a) Left, center and right views (front view).



(b) Left, center and right views (top view).

Figure 6.30 Superposed range measurements from the three views after registrations (real computer scene).

The computing time required for each of the above two registrations is about 60 seconds with the same Matlab code running on the same processor as in the previous experiments. It is longer in comparison with that in the third experiment because many more raw measurements are employed with the 130 scan lines in each view. Moreover, this time raw measurements contain noise that contributes to lengthen the determination of the compact surface map. The fact that a large number of patches with different orientations exist on the scene surfaces also makes the implementation of the program slower.

6.6 Discussion

As described in chapter 4, the proposed strategy makes use of a simple representation of surfaces and therefore significantly reduces the computation time of the registration parameters which is mainly related to the number of normal vectors contained in the modified Gauss spheres. As we need to exhaustively search three vectors from each modified Gauss sphere in order to match them and estimate the full set of 3 rotation parameters, the number of normal vectors considered is critical. Taking advantage of the fact that most man-made objects consist of a limited number of faces with different orientations, the minimization of the number of normal vectors that is proposed results in a very compact Gauss sphere representation. As a consequence, the computation time becomes reasonable, that is about 6 seconds for the equivalent of a 40*256 range image used in the above experiments. This computing time is achieved in spite of the fact that the implementation is running on Matlab which is far from being the most efficient platform.

Actually, even for some complex or natural objects, the number of vectors can still be reduced by adjusting some parameters (WARP, RotationTh, OverlapTh or CloseNormal) when performing the line segment approximation and the compaction of normal vectors. In spite of some loss in the resolution of the representation, this strategy proves to be sufficiently accurate for many applications while it significantly speeds up the computation in comparison with more rendering-based representations such as classical triangular meshes for which a large number of facets are usually required to approximate the object surfaces. On the other hand, when the number of normal vectors in one Gauss sphere is less than two, or when three matching vectors from two Gauss spheres are not available, the program is made robust by adjusting itself to estimate two rotations by assuming the third one to be zero under the specification of the user. This reveals to be useful as many applications do not imply three rotations such as mobile robot navigation.

We notice that the error level on translation estimates is somewhat higher than that on the rotation estimates. The fact that the rotation estimates are used to compute the translation ones implies that rotation errors have accumulative impact on translation errors. For the experimental results presented, we located the region of maximum of occurrence of shifting values by selecting those areas with a maximum of occurrence along each axis separately, then we united these areas to form the 3D region of interest. Much time is saved in this way but it has an impact on the accuracy as delimiting this region is not trivial. The union region of those areas containing maximums of occurrence along a specific axis is not necessary the region with maximum of occurrence in 3D space. If more accurate estimates are expected, we can consider the 3D region of maximum of occurrence along the X, Y and Z axes together rather than separately. Therefore, the limitation originating from the incorrect union is eliminated.

In chapter 3, we have discussed the accuracies of the measurements and the theoretical parameters extraction from the experimental setup. The F3 robot arm provides a precision of 0.005^{o} for rotation values. Theoretically, this precision should be extended to $2*0.005 = 0.01^{o}$ after the application of equation (3-1) because Q_1 and Q_2 are two transformations including the rotation. On the other hand, the measurement resolution of the Jupiter laser sensor varies from 0.6 mm to 2.9 mm along the Y axis and from 0.3 mm to 2 mm along the Z axis. Based on these limitations, it is normal that we cannot achieve a higher precision in the results.

One important thing that needs to be mentioned here is that the theoretical registration parameters we extracted from equation (3-1) are not fully reliable. It may have a deviation of up to a few degrees in rotation and few tens of millimeters in translation. This comes from the imprecise calibration of the Tool Center Point of the manipulator mainly resulting from the use of a home-made bracket to attach the camera on the manipulator and from some imprecise calibration of the hardware. As these deviations are not precisely known, we provided 'theoretical values' as an estimated reference. The superposition of the data from different viewpoints while applying the computed registration parameters is used as an alternative way to validate the proposed algorithm.

In the experiments presented above, the rotation errors are between 0.01^{o} and 1.23^{o} for the real data sets and the translation errors are between 1.4 mm and 13.6 mm. Overall, we observe that a maximum relative deviation of about 2% is achieved for both rotation and translation estimations. This appears to be satisfactory for many applications. Moreover, we observed a higher accuracy in the implementation on simulated data due to the absence of noise.

It is also shown in the experiments that a higher precision is achieved for the translation estimates along the Z axis than that along the Y axis. This can be due to the following reasons: The first one is that we applied a larger movement of the sensor along the Y axis than along the Z axis between two viewpoints as shown in the example of the registration between the center and the left viewpoints in the third experiment. In this case, an error of 4.5 mm is obtained for the translation of 300 mm along the Y axis, while an error of 0.1 mm is produced with no translation along the Z axis. This corresponds to a maximum deviation of 1.5% which is satisfactory given the complexity of the scene. Second, the fact that the resolution of range measurements along Z is higher than that along Y for the same position in 3D space as illustrated in chapter 3 can also partially justify why we obtain more precise estimates of translation along the Z axis than along the Y axis. Finally, having a better estimate of translation along Z than along Y when estimating the tool center point (TCP) may be another reason influencing the accuracy of the translation estimates.

Actually, the precision we obtained in the above experiments is good enough for most applications where large environments are to be inspected and medium to coarse quality models are expected. If more precise parameters are required, the ICP (Iterative Closest Point) algorithm discussed in the literature review can be applied to refine the results after the proposed algorithm has been applied. On the other hand, the precision for the translation estimates might also be improved by taking advantage of the distance from the origin to the patch. The same patches used for rotation estimation can be reused to compute three distance vectors, from which three translation parameters are estimated. Compared to the computation of the shifts between centroid points, this strategy can theoretically ensure a more precise result and it is independent from the overlap between two patches as long as they represent the same surface on the object. This strategy is to be tested in future work.

Our experiments revealed that for geometrically symmetric objects having only a few distinctive surfaces, an overlap of up to 50% between the scanned areas might be required to achieve correct registration. However for common man-made objects, the required overlap area is much less. The minimum overlap mostly depends on the complexity of the objects. Also, increasing the overlap between scanned areas increases the reliability but makes the scanning process longer. A trade-off must then be found to minimize overlap while preserving sufficient information for matching.

We also observed that the accuracy of the experimental results depend on that of the patch representation. That is why lots of efforts have been spent on testing various alternatives for the scan merging process that allow to preserve compactness while achieving a sufficient accuracy. Even though some errors persist, the experimental results presented here demonstrate the validity of the proposed approach and its coding. The accuracy of the estimated registration parameters is sufficient for most applications in robot navigation with collision avoidance where computing time is a critical issue that makes more sophisticated algorithms based on high-level features extraction not tractable.

Overall, the scan segmentation and surface representation as a compact modified Gauss sphere reveals to significantly improve the computation efficiency while it is not significantly degrading the accuracy of registration parameter estimates. As we might anticipate, the computation time depends on the size of the workspace and the desired resolution. As a consequence, a compromise must be made between the computation time and the desired accuracy. The strategy also makes patch fitting simpler and faster. Using the overlap of normal vectors as an evaluation factor rather than the number of matched points between the two sets of data speeds up computation. This strategy also uses more points to estimate the surface patch than triangular meshes. This results in surface representations that are less sensitive to noise measurements.

Chapter 7 Conclusion and Future Work

3D modeling is an important issue in computer vision. It has many applications in robot navigation, unknown environment exploration and other intelligent operation situations. In this thesis, we made a detailed study of the Jupiter laser range sensor that can be used to build 3D models of robot workspaces. Based on the fact that such a sensor has a very limited field of view, we developed an automatic approach for the estimation of registration parameters between viewpoints of the sensor. This approach makes use of the raw range measurements only without any external pose tracking device.

All the requirements and goals stated in chapter 4 have been met. As most off-the-shelf range sensors, the Jupiter camera provides single scan lines of raw range measurements rather than full range images that are usually considered in most research projects. As a result, the proposed registration technique makes use of a simple line segment representation to group scan lines together and create planar patch segments to represent object surface. Taking advantage of a modified Gauss sphere representation, the mapping of planar patches is significantly compacted. This solution results in an important reduction of the computational workload while providing efficient means for matching estimation and validation all along the process. The registration parameters are computed without any need for an external device or preprocessing to provide an initial estimate, nor any feature extraction or triangulation. Assuming only object rigidity and some overlap between the scanned areas, the approach allows to estimate the six DOF parameters that link 3D scans gathered from different viewpoints. The proposed approach provided excellent experimental results with a sufficient accuracy for most robot navigation applications with collision avoidance.

The proposed contributions listed in chapter 1 are also achieved. First, an original automatic registration algorithm is proposed and no *a priori* knowledge and no human interaction are required. Second, a modified Gauss sphere is employed to represent the data sets in a compact way. This can reduce the data size from thousands of points to a few planar patches, and can therefore speed up the estimation of registration. Third, the approach works without a cost function. Instead, use is made of the overlap between data sets as an evaluation factor. Not only can it facilitate the estimation, but it also makes the algorithm less sensitive to noise and to varying resolutions found in range measurements. Fourth, a scheme that allows to register two data sets globally is proposed. This is achieved by grouping those patches that have similar orientation (within some tolerance level) without taking into account the geometrical connectivity between them. Finally, 3 rotation parameters are estimated separately from 3 translation parameters. Patch normal vectors are used to first estimate the rotation parameters while the estimation of translation parameters makes use of the centroid points of corresponding patches between two data sets.

Further developments to this registration technique will examine some alternative ways to obtain the patch segments. Instead of following the order of point index when performing line segment approximation, longer segments could be estimated before smaller segments in order to give them a higher priority. This would reduce the impact of noise, provide a better approximation and facilitate the merge of scan lines that would eventually result in a more robust patch estimation. Rectangular shape patches might also be complemented by some irregular shape patches which would provide a more complete approximation of the object surfaces, especially for the transition zones between two consecutive groups of scan profiles.

A better method, which would accelerate the three rotation parameters extraction, could also be explored to search for three pairs of corresponding normal vectors from the two modified Gauss spheres. Factors other than the angular difference between normal vectors can be considered to constrain the search. Improvements on the estimation of translation parameters could also be achieved as discussed in chapter 6. For example, the determination of the high density areas in the 3D shifts occurrence graph could possibly be improved by means of a hierarchical location scheme.

Finally, an implementation in C with optimization of the code will certainly result in an important reduction of the computing time that will make this technique highly tractable for on-line operating systems. An initial trial has shown that the CPU time can be reduced by a factor of 6 to 10, which can be further improved.

Appendix A *.3dx File Format

Table A.1 *.3DX file format needed to retrieve raw range profiles.

Field	Format
Title (unique)	
Identification	"SRI3DX"
Number of 3DX version	unsigned char (8 bits)
Number of CPU version	5*unsigned short (16 bits)
Number of WinUser version	4*unsigned short (8 bits)
Number of profiles	unsigned short (16 bits)
Unit of vision	unsigned short (16 bits)
Content	unsigned char (8 bits)
Profiles (x number of profiles)	
Number of pixels	unsigned short (16 bits)
Direction	unsigned char (8 bits)
Pixels	number of pixels * (short y; short z; unsigned char intensity)

Appendix B RAPL-3 Program

This appendix presents the RAPL-3 program that has been developed to move the Jupiter sensor from one viewing area to another one.

```
***********************************************************************
**********                    Main function                  **********
***********************************************************************

int dire = 1
main
    teachable int fast, slow
    teachable float low, high
    teachable ploc start, view1, view2
    teachable cloc safe
    teachable int dstep, tstep
    teachable float mstep
    int count

    dire = 1
    speed(slow)
    move(start)
    view1 = start
    view1 = getfirstview(view1, high, low)
    print(view1, "\n")
    view2 = getnextview(view1, dstep, tstep)
    print(view2, "\n")
    move(view1)
    print("\nBegin to scan view1\n")
    count =scanview(view1, mstep, high, low)
    print(count, " scans are performed\n")
```

```
print("\nplease wait for next scan while robot is moving\n")
print("\and then press F1 to continue\n")
if(onbutton(B_F1, ON))
    panel_light_set(B_F1, ON)
else
    delay(250)
end if
print("\nBegin to scan view2\n")
count = scanview(view2, mstep, high, low)
print(count, " scans are performed\n")
print("\nEnd of the program\n")
speed(fast)
end main


**********************************************************************
**********                  Procedure (1)                   **********
**********************************************************************

func ploc getfirstview(ploc view1, float high, float low)
    cloc cloc_temp
    ploc ploc_temp
    float [8] loc_cdata

    pos_get(POSITION_ACTUAL, view1)
    ploc_temp = view1
    motor_to_world(ploc_temp, cloc_temp)
    loc_cdata_get(cloc_temp, loc_cdata)
    if loc_cdata[2] > high
        loc_cdata[2] = high
        dire = 1
    elseif loc_cdata[2] < low
        loc_cdata[2]=low
        dire = 0
    end if
    loc_cdata_set(cloc_temp, loc_cdata)
    world_to_motor(cloc_temp, ploc_temp)
    return ploc_temp
    end func
```

```
*************************************************************************
**********                  Procedure (2)                      **********
*************************************************************************

func ploc getnextview(ploc view1, int dstep, int tstep)
    const TRACK = 260000
    int [8] loc_pdata
    cloc cloc_temp
    ploc ploc_temp

    loc_pdata_get(view1, loc_pdata)
    loc_pdata[0] = loc_pdata[0] + dstep
    loc_pdata[6] = loc_pdata[6] + tstep
    if loc_pdata[6] > TRACK
       print("\nOut of track range, use maximum track range instead\n")
       loc_pdata[6] = TRACK
    end if
    loc_pdata_set(ploc_temp, loc_pdata)
    motor_to_world(ploc_temp, cloc_temp)
    world_to_motor(cloc_temp, ploc_temp)
    return ploc_temp
end func


*************************************************************************
**********                  Procedure (3)                      **********
*************************************************************************

func int scanview(ploc startpoint, float mstep, float high, float low)
    ploc ploc_temp
    cloc cloc_temp
    float[8] loc_cdata
    int n

    n = 0
    ploc_temp = startpoint
    motor_to_world(startpoint, cloc_temp)
    move(startpoint)
    do
       loc_cdata_get(cloc_temp, loc_cdata)
```

```
        if dire == 1
            loc_cdata[2] = loc_cdata[2] - mstep
        else
            loc_cdata[2] = loc_cdata[2] + mstep
        end if
        loc_cdata_set(cloc_temp, loc_cdata)
        world_to_motor(cloc_temp, ploc_temp)
        panel_lights_set(0xf, 0x0)
        online(ON)
        print("\nplease press PAUSE/CONTINUE for continue\n\n")
        if(onbutton(B_PAUSE_CONT, ON))
            panel_light_set(B_PAUSE_CONT, ON)
        else
            move(ploc_temp)
            delay(250)
        end if
        n = n + 1
        print(n,"\n")
    until (loc_cdata[2]>high-mstep + 1 || loc_cdata[2]<low+mstep - 1)
    print("\nEnd of one scan\n\n")
    return n
end func
```

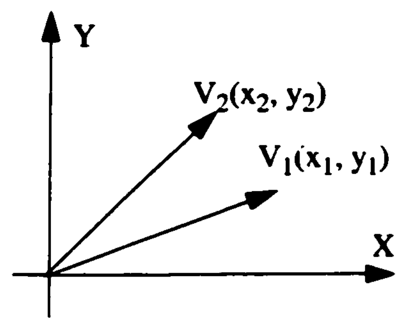
Appendix C Closed-Form Solution For Registration Estimation

C.1 2D registration problem

This appendix discusses the closed-form solution for computing 3 registration parameters given two corresponding normal vectors and the associated centroid points.

Provided V_1 and V_2 are two corresponding normal vectors in two Gauss circles. cp_1 and cp_2 are the centroid points of their respective line segments.

$$V_1 = (x_1, y_1)$$
$$V_2 = (x_2, y_2)$$
$$cp_1 = (x_{p1}, y_{p1})$$
$$cp_2 = (x_{p2}, y_{p2})$$



Then the rotation parameter, $d\varphi$, can be easily computed as follows:

$$d\varphi = \operatorname{atan}\left(\frac{y_2}{x_2}\right) - \operatorname{atan}\left(\frac{y_1}{x_1}\right) \tag{C-1}$$

We also have from the 2D affine transformation matrix:

$$\begin{bmatrix} x_{p1} \\ y_{p1} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & dx \\ -\sin\varphi & \cos\varphi & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{p2} \\ y_{p2} \\ 1 \end{bmatrix}$$

The translation parameters can therefore be derived in equation (C-2) and (C-3).

$$dx = x_{p1} - x_{p2} \cdot \cos\varphi - y_{p2} \cdot \sin\varphi \qquad \text{(C-2)}$$

$$dy = y_{p1} + x_{p2} \cdot \cos\varphi - y_{p2} \cdot \sin\varphi \qquad \text{(C-3)}$$

C.2 Simplified 3D registration problem

Assuming that the rotation around the Z axis, θ, is always zero, and $V_1 = [x, y, z]^T$ and $V_2 = [a, b, c]^T$ are two corresponding normal vectors, the other two rotation parameters (φ, ψ) can then be computed as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi \cdot \sin\psi & \sin\varphi \cdot \cos\psi \\ 0 & \cos\psi & -\sin\psi \\ -\sin\varphi & \cos\varphi \cdot \sin\psi & \cos\varphi \cdot \cos\psi \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$x = a \cdot \cos\varphi + b \cdot \sin\varphi \cdot \sin\psi + c \cdot \sin\varphi \cdot \cos\psi \qquad \text{(C-4)}$$

$$y = b \cdot \cos\psi - c \cdot \sin\psi \qquad \text{(C-5)}$$

$$z = -a \cdot \sin\varphi + b \cdot \cos\varphi \cdot \sin\psi + c \cdot \cos\varphi \cdot \cos\psi \qquad \text{(C-6)}$$

Multiplying (C-5) by cos(ψ) and adding with (C-6) multiplied by sin(ψ), we get:

$$a = x \cdot \cos\varphi - z \cdot \sin\varphi \tag{C-7}$$

From (C-5), we have:

$$\frac{b}{\sqrt{b^2+c^2}} \cdot \cos\psi - \frac{c}{\sqrt{b^2+c^2}} \cdot \sin\psi = \frac{y}{\sqrt{b^2+c^2}} \tag{C-8}$$

$$\sin(\alpha - \psi) = \frac{y}{\sqrt{b^2+c^2}} \tag{C-9}$$

where:

$$\alpha = \operatorname{atan}\left(\frac{b}{c}\right) \tag{C-10}$$

As a result, we obtain the closed-form solution for ψ to be:

$$\psi = \operatorname{asin}\left(\frac{y}{\sqrt{b^2+c^2}}\right) + \operatorname{atan}\left(\frac{b}{c}\right) \tag{C-11}$$

Similarly, the closed-form solution for φ can be extracted from equation (C-7):

$$\varphi = \operatorname{asin}\left(\frac{a}{\sqrt{x^2+z^2}}\right) + \operatorname{atan}\left(\frac{x}{z}\right) \tag{C-12}$$

Appendix D Straight-Line Segment Fitting

This appendix discusses the problem of straight-line segment fitting based on some known points. This is achieved by optimizing two straight line parameters (slope and intercept) such that the sum of the squared distances between the data points and the approximated line segment is minimized. Mathematically, we follow the following steps.

Provided $p_i = (x_i, y_i)$, where i = (0, 1, 2, ..., N-1) are the raw points to be interpolated by a straight-line parameterized as:

$$y(x) = a_0 + a_1 \cdot x \tag{D-1}$$

Interpolate all points in equation (D-1), and rewrite it in matrix format:

$$\begin{bmatrix} y_0 \\ y_1 \\ \dots \\ \dots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & \dots \\ 1 & \dots \\ 1 & x_{N-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \tag{D-2}$$

$$Y = X \cdot A$$

$$X^T \cdot Y = X^T \cdot X \cdot A$$

$$[X^T \cdot X]^{-1} \cdot X^T \cdot Y = [X^T \cdot X]^{-1} \cdot [X^T \cdot X] \cdot A$$

Finally, the two parameters defining the straight line can be computed from:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = [X^T \cdot X]^{-1} \cdot X^T \cdot Y \tag{D-3}$$

References

[1] D. Young, «Computer Vision» course notes, University of Sussex, *http://www.cogs.susx.ac.uk/users/davidy/compvis.*

[2] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", *Computer*, Vol.22, No.6, pp.46-57, Jun. 1989.

[3] P. Payeur, P. Hebert *et al.*, "Probabilistic Octree Modeling of a 3-D Dynamic Environment", in *Proceedings of the IEEE International Conference On Robotics and Automation*, Vol.2, pp.1289-1296, Apr. 1997.

[4] P. Payeur, D. Lauren deau, and C. M. Gosselin, "Range Data Merging for Probabilistic Octree Modeling of 3-D Workspaces", in *Proceedings of the IEEE International Conference On Robotics and Automation*, Vol. 4, pp.3071-3078, May, 1998.

[5] H. Noborio, S. Fukuda and S. Arimoto, "Construction of The Octree Approximating Three-Dimentional Objects by Using Multiple Views", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.10, No.6, pp.769-782, Nov.1988.

[6] Y. Yu, and K. Gupta, "An efficient on-line Algorithm for Direct Octree Construction From Range Images", in *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[7] C. Dorai, G. Wang *et al.*, "Registration and Integration of Multiple Object Views For 3D Model Construction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.1, pp.83-89, Jan. 1998.

[8] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images", *Image and Vision Computing*, Vol.10, No.3, pp.145-155, 1992.

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

[10] G. Turk and M. levoy, "Zippered Polygon Meshes From Range Images", in *Proceedings of SI GGRAPH*, pp.311-318, July 1994.

[11] C. S. Chua and R. Jarvis, "3D Free-Form Surface Registration and Object Recognition", *International Journal of Computer Vision*, Vol.17, pp.77-99, 1996.

[12] K. Higuch, M. Hebert and K. Ikeuchi, "Building 3D Models from Unregistered Range Images", *Graphical Models and Image Processing*, Vol.57, no.4, pp.315-333, 1995.

[13] D. F. Huber and M. Hebert, "A New Approach to 3-D Terrain Mapping", The Robotics Institute, Carnegie Mellon University, *http://www.ri.cmu.edu*.

[14] O. Carmichael, D. Huber and M. Hebert, "Large Data Sets and Confusing Scenes in 3-D Surfaces Matching and Recognition", The Robotics Institute, Carnegie Mellon University, *http://www.ri.cmu.edu*.

[15] G. Roth, "Registering Two Overlapping Range Images" in *Proceedings of the Second International Conference on Recent Advances in 3-D Digital Imaging and Modelling (3DIM'99)*, pp.191-200, NRC 43637, Ottawa, Ontario, Oct. 1999.

[16] C. Dorai, J. Weng *et al.*, "Optimal Registration of Object Views Using Range Data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.10, Oct. 1997.

[17] D.A. Forsyth, J. Ponce, *Computer Vision, A Modern Approach*, Prentice Hall, 2003.

[18] S. F. El-Hakim, J.-A. Beraldin, *et al.*, "Two 3D Sensors for Environment Modelling and Virtual Reality: Calibration and Multiview Registration", in *Proceedings of ISPRS Congress, International Archives of Photogrametry and Remote Sensing, Commission V*, pp.140-146, 1996.

[19] D. Laurendeau, C. Gosselin *et al.*, "A Random Access 3D/2D Vision Sensor", in *Proceedings of Vision Interface*, pp.93-100, Vancouver, BC, June 1998.

[20] L. P. Normand, "Sonar-Based Real World Mapping System", Master's thesis, University of Ottawa, 1990.

[21] J. R. Miller, O. Amidi, and M. Delonis, "Arctic Test Flights of The CMU Autonomous Helicopter", in *Proceedings of the 26th Symposium of the Association for Unmanned Vehicle Systems*, July 1999.

[22] P. Neugebauer, "Geometrical Cloning of 3D Objects Via Simultaneous Registration of Multiple Range Images", in *Proceedings of the 1997 International Conference on Shape Modeling and Applications*, pp.130-139, Mar. 1997.

[23] M. K. Reed, P. K. Allen and I. Stamos, "3-D Modeling from Range Imagery: An Incremental Method With a Planning Component", in *Proceedings of the International Conference on Advances in 3-D Digital Imaging and Modeling*, Ontario, Canada, 1997.

[24] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment For Environment Mapping", *Autonomous Robots* 4(4), pp.333-349, Oct. 1997.

[25] V. Sequeira, K. Ng *et al.*, "Automated 3D Reconstruction of Interiors With Multiple Scan – views", in *Proceedings of Videometrics VI SPIE* vol.3641, pp.106-117, Jan.1999.

[26] M. Whecler, "Automatic Modeling and Localization For Object Recognition", Ph.D thesis, Carnegie Mellon University, Pittsburgh, PA, Oct. 1996.

[27] T. Masuda and N. Yokoya, "A Robust Method For Registration and Segmentation of Multiple Range Images", *Computer Vision and Image Understanding*, Vol.61, No.3, pp.295-302, May 1995.

[28] F. P. Ferrie and M. D. Levine, "Integrating Information From Multiple Views", in *Proceedings of the IEEE Computer Society Workshop on Computer Vision, Miami*, FL, pp.117-122, Dec. 1987.

[29] D. Laurendeau, G. Roth and L. Borgeat, "Optimization Algorithm for Range Image Registration," in *Vision Interface 96*, Toronto, Ontario, May 1996.

[30] C. Chen, Y. Hung and J. Cheng, "RSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.11, Nov. 1999.

[31] A. Johnson, "Spin Images: A representation for 3-D Surface Matching", Ph.D thesis, The Robotic Institute, Carnegie Mellon University, Aug. 1997.

[32] O. D. Faugeras and M. Hebert, "The Representation, Recognition and Locating of 3D Objets", *International Journal of Robotic Research*. Vol.5, no.3, pp.27-52, Fall 1986.

[33] N. Kehtarnavaz and S. Mohan, "A Frame Work for Estimation of Motion Parameters From Range Images", *Computer Vision, Graphics, and Image Processing*, vol.45, pp.88-105, 1989.

[34] A. Johnson and M. Hebert, "Object Recognition by Matching Oriented Points", in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Puerto Rico, Jun. 1997.

[35] A. Johnson and M. Hebert, "View Registration by Matching Oriented Points", in *Proceedings of the International Conference on Recent Advances in 3D Digital Imaging and Modeling*, Ottawa, May 1997.

[36] R. Bergevin, D. Laurendeau and D. Poussart, "Registering Range Views Of Multipart Objects", *Computer Vision and Image Understanding*, Vol.61, No.1, pp.1-16, Jan. 1995.

[37] I. Stamos and P. K. Allen, "3-D Model Construction Using Range and Image Data", in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol.1, pp.531-536, South Carolina, June 2000.

[38] R. Szeliski, "Estimating Motion From Sparse Range Data Without Correspondence", in *Proceedings of the International Conference on Computer Vision*, pp.207–216, Dec. 1988.

[39] B. K. Horn and J. G. Harris, "Rigid Body Motion From Range Image Sequences", *CVGIP: Image Understanding*, vol.53, pp.1-13, 1991.

[40] B. K.–Parsi, J. L. Jones, and A. Rosenfeld, "Registration of Multiple Overlapping Range Images: Scenes Without Distinctive Features," in *Proceedings of the IEEE Computer Vision Pattern Recognition Conference*, San Diego, CA, June 1989.

[41] G. Taubin, "Curve and Surface Estimation in 3-Space with Applications to Position Estimation", Brown University Technical Report LEMS-43, Feb. 1988.

[42] A. K. C. Wong, S. W. Lu, and M. Rioux, "Recognition and Shape Synthesis of 3D Objects Based on Attributed Hypergraphs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.11, no.3, pp.279-290, Mar. 1989.

[43] S. Z. Li, "Inexact Matching of 3D Surfaces", VSSP-TR-3-90, University of Surrey, England, 1990.

[44] E. Trucco, A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall, 1998.